



*Robert Krimmer, Melanie Volkamer, David Duenas-Cid,
Micha Germann, Stéphane Glondu, Thomas Hofer,
Iuliia Krivonosova, Oksana Kulyk, Beata Martin-Rozumilowicz, Peter
Rønne, Mihkel Solvak, Marie-Laure Zollinger (Eds.)*

**Sixth International Joint Conference on Electronic
Voting**

E-Vote-ID 2021

5-8 October 2021

Co-organized by:

University of Tartu
Johan Skytte Institute of Political Studies
Karlsruhe Institute of Technology
Institute of Applied Informatics and Formal Description Methods
Kozminski University
Management in Networked and Digital Societies
E-Voting.CC GmbH
Competence Center for Electronic Voting and Participation
Gesellschaft für Informatik
German Informatics Society, SIG SEC/ECOM
Kastel
Competence Center for Applied Security Technology

PROCEEDINGS



Gesellschaft
für Informatik



Robert Krimmer, Melanie Volkamer, David Duenas-Cid
Micha Germann, Stéphane Glondu, Thomas Hofer, Iuliia Krivonosova,
Oksana Kulyk, Beata Martin-Rozumilowicz, Peter Rønne, Mihkel Solvak,
Marie-Laure Zollinger (Eds.)

6th Joint International Conference on Electronic Voting

E-Vote-ID 2021

5-8 October 2021

**Co-organized by the University of Tartu, Karlsruhe Institute of
Technology, Kozminski University, E-Voting.CC, Gesellschaft für
Informatik and Kastel**



UNIVERSITY OF TARTU
Press

Proceedings E-Vote-ID 2021
University of Tartu Press
ISBN 978-9949-03-735-3

Volume Editors

Prof. Dr. Robert Krimmer
University of Tartu
Johan Skytte Institute of Political Studies
Lossi 36
51003 Tartu, Estonia
robert.krimmer@ut.ee

Prof. Dr. Melanie Volkamer
Karlsruhe Institute of Technology
Institute of Applied Informatics and Formal Description Methods
Kaiserstr. 89
76131 Karlsruhe, Germany
melanie.volkamer@secuso.org

Dr. David Duenas-Cid
Kozminski University
Management in Networked and Digital Societies
Jagiellonska 57
03-301 Warsaw, Poland
University of Tartu
Johan Skytte Institute of Political Studies
Lossi 36
51003 Tartu, Estonia
dduenas@kozminski.edu.pl / david.duenas.cid@ut.ee

Micha Germann
University of Bath
E-mail: mg2107@bath.ac.uk

Stéphane Glondu
Institut National de Recherche en Sciences
et Technologies du Numérique
E-mail: stephane.glondu@inria.fr

Thomas Hofer
Objectif Sécurité
E-mail: thomas.hofer@objectif-securite.ch

Iuliia Krivososova
Tallinn University of Technology
E-mail: iuliia.krivososova@taltech.ee

Oksana Kulyk
IT University of Copenhagen
E-mail: okku@itu.dk

Beata Martin-Rozumlowicz
International Foundation for Electoral
Systems
E-mail: bmartinrozumilowicz@ifes.org

Peter Rønne
University of Luxembourg
E-mail: peter.roenne@gmail.com

Mihkel Solvak
University of Tartu
E-mail: mihkel.solvak@ut.ee

Marie-Laure Zollinger
University of Luxembourg
E-mail: marie-laure.zollinger@uni.lu

This conference is co-organized by:



University of Tartu - Johan Skytte Institute of Political Studies



Karlsruhe Institute of Technology - Institute of Applied Informatics and Formal Description Methods



KOZMINSKI UNIVERSITY

Kozminski University – Management in Networked and Digital Societies



E-Voting.CC GmbH - Competence Center for Electronic Voting and Participation

Gesellschaft
für Informatik



Gesellschaft für Informatik, German Informatics Society, SIG SEC/ECOM



Kastel, Competence Center for Applied Security Technology

Supported by:



Regional Government of Vorarlberg



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Federal Chancellery

General Chairs

Krimmer, Robert (University of Tartu - Johan Skytte Institute of Political Studies, Estonia)

Volkamer, Melanie (Karlsruhe Institute of Technology - Institute of Applied Informatics and Formal Description Methods, Germany)

Duenas-Cid, David (Kozminski University – Management in Networked and Digital Societies, Poland and University of Tartu - Johan Skytte Institute of Political Studies, Estonia)

Track on Security, Usability and Technical Issues

Oksana Kulyk

IT University of Copenhagen, Denmark

Peter Rønne

University of Luxembourg, Luxembourg

Poster and Demo Session

Glondou, Stéphane

Institut National de Recherche en

Sciences et Technologies du

Numérique, France

Track on Administrative, Legal, Political and Social Issues

Mihkel Solvak

University of Tartu, Estonia

Micha Germann

University of Bath, UK

Organizational Committee

Licht, Nathan

Castle, Salina

E-Voting.CC, Austria

Track on Election and Practical Experiences

Beata Martin-Rozumilowicz

International Foundation for Electoral Systems, USA

Thomas Hofer

Objectif Sécurité, Switzerland

Outreach Chairs

Rønne, Peter

University of Luxembourg, Luxembourg

Krivosova, Iuliia

Tallinn University of Technology, Estonia

PhD Colloquium

Iuliia Krivosova

Tallinn University of Technology, Estonia

Marie-Laure Zollinger

University of Luxembourg, Luxembourg

Preface

This volume contains papers presented at E-Vote-ID 2021, the Sixth International Joint Conference on Electronic Voting, held during October 5-8, 2021. Due to the extraordinary situation provoked by Covid-19 Pandemic, the conference is held online for second consecutive edition, instead of in the traditional venue in Bregenz, Austria. E-Vote-ID Conference resulted from the merging of EVOTE and Vote-ID and counting up to 17 years since the first E-Vote conference in Austria. Since that conference in 2004, over 1000 experts have attended the venue, including scholars, practitioners, authorities, electoral managers, vendors, and PhD Students. The conference collected the most relevant debates on the development of Electronic Voting, from aspects relating to security and usability through to practical experiences and applications of voting systems, also including legal, social or political aspects, amongst others; turning out to be an important global referent in relation to this issue.

Also, this year, the conference consisted of:

- Security, Usability and Technical Issues Track
- Administrative, Legal, Political and Social Issues Track
- Election and Practical Experiences Track
- PhD Colloquium, Poster and Demo Session on the day before the conference

E-VOTE-ID 2021 received 49 submissions, being, each of them, reviewed by 3 to 5 program committee members, using a double blind review process. As a result, 27 papers were accepted for its presentation in the conference. The selected papers cover a wide range of topics connected with electronic voting, including experiences and revisions of the real uses of E-voting systems and corresponding processes in elections.

We would also like to thank the German Informatics Society (Gesellschaft für Informatik) with its ECOM working group and KASTEL for their partnership over many years. Further we would like to thank the Swiss Federal Chancellery and the Regional Government of Vorarlberg for their kind support. E-Vote-ID 2021 conference is kindly supported through European Union's Horizon 2020 projects ECEPS (grant agreement 857622) and mGov4EU (grant agreement 959072). Special thanks go to the members of the international program committee for their hard work in reviewing, discussing, and shepherding papers. They ensured the high quality of these proceedings with their knowledge and experience.

October 2021
Virtual Event

Robert Krimmer
Melanie Volkamer
David Duenas-Cid
Oksana Kulyk
Peter Roenne
Mihkel Solvak
Micha Germann
Beata Martin-Rozumilowicz
Thomas Hofer
Iuliia Krivonosova
Marie Laure Zollinger
Stéphane Glondu

Table of Contents

Improving Verifiability

STROBE-Voting: Send Two, Receive One Ballot Encoding	1
<i>Josh Benaloh</i>	
Improved Verifiability for BeleniosVS	15
<i>Thomas Haines and Rajeev Gore</i>	
Provably Improving Election Verifiability in Belenios	28
<i>Sevdenuur Baloglu, Sergiu Bursuc, Sjouke Mauw and Jun Pang</i>	

Internet Voting: Behavioral Aspects

Party cues and trust in remote internet voting: data from Estonia 2005-2019	45
<i>Piret Ehin and Mihkel Solvak</i>	
To i-vote or not to i-vote: Drivers and Barriers to the Implementation of Internet Voting	61
<i>Nathan Licht, David Duenas-Cid, Iuliia Krivonosova and Robert Krimmer</i>	
Turnout in e-voting pilots in the 2021 presidential elections in Ecuador . .	78
<i>Régis Dandoy</i>	

On the offensive: IT Threats

Penetration Testing a US Election Blockchain Prototype	82
<i>Shawn Emery, C. Edward Chow and Richard White</i>	
How to Break Virtual Shareholder Meetings: An Empirical Study	98
<i>Andreas Mayer</i>	
Tales from the Trenches: Case Studies in Election Cybersecurity Preparedness in Texas	111
<i>Elizabeth Kasongo, Matthew Bernhard and Chris Bronk</i>	

Risk Limiting Audits

RiLACS: Risk-Limiting Audits via Confidence Sequences	130
<i>Ian Waudby-Smith, Philip Stark and Aaditya Ramdas</i>	
Assertion-based Approaches to Auditing Complex Elections, with application to party-list proportional elections	146
<i>Michelle Blom, Jurlind Budurushi, Ron Rivest, Philip Stark, Philip J. Stuckey, Vanessa Teague and Damjan Vukcevic</i>	

Risk-limiting Audits: A Practical Systematization of Knowledge 162
Matthew Bernhard

Managing Election Integrity

Retaining Election Officials is Imperative to Secure Future Elections 179
David Levine

Vote Secrecy and Voter Feedback in Remote Voting – Can We Have Both? 190
Jan Willemson, Pritt Vinkel and Arne Koitmaa

Australian Senate Scrutiny Reform Proposal 204
Ian Brightwell

Security of Electronic and Paper-Based Voting

Cyber Awareness Training for Election Staff using Constructive Alignment 222
Thomas Chanussot and Carsten Schuermann

And Paper-Based is Better? Towards Comparability of Classic and
Cryptographic Voting Schemes 232
*Marc Nemes, Rebecca Schwerdt, Dirk Achenbach, Bernhard Löwe and
Jörn Müller-Quade*

Improving the Accuracy of Ballot Scanners Using Supervised Learning . . . 235
*Sameer Barretto, William Chown, David Meyer, Aditya Soni, Atreya
Tata and J. Alex Halderman*

Enhancing Privacy in Voting

Who was that masked voter? The tally won't tell! 252
*Peter Y. A. Ryan, Peter B. Roenne, Philip Stark, Dimiter Ostrev,
Najmeh Soroush and Fatima-Ezzahra El Orche*

Extending the Tally-Hiding Ordinos System: Implementations for
Borda, Hare-Niemeyer, Condorcet, and Instant-Runoff Voting 269
*Fabian Hertel, Nicolas Huber, Jonas Kittelberger, Ralf Küsters, Julian
Liedtke and Daniel Rausch*

Hyperion: An Enhanced Version of the Selene End-to-End Verifiable
Voting Scheme 285
Peter Y. A. Ryan, Peter B. Roenne and Simon Rastikian

The Road to Deploying e-Voting

The challenges of enabling public scrutiny 289
Xavier Monnat and Simon Oswald

Use of Electronic Voting in the Albanian Parliamentary Elections in 2021	304
<i>Jurlind Budurushi</i>	
Use of innovative technologies in the electoral process in Armenia	320
<i>Ardita Driza Maurer, Justin Nettman and Rafik Grigoryan</i>	
Usability and Voter Perception	
Voter Perceptions of Trust in Risk-Limiting Audits	335
<i>Asmita Dalela, Oksana Kulyk and Carsten Schürmann</i>	
Usable Verifiable Secrecy-Preserving E-Voting	337
<i>Oksana Kulyk, Reto König, Philipp Locher, Jonas Ludwig and Melanie Volkamer</i>	
”Just for the sake of transparency”: Exploring Voter Mental Models of Verifiability	354
<i>Marie-Laure Zollinger, Ehsan Estaji, Peter Y. A. Ryan and Karola Marky</i>	
PhD Colloquium	
Evaluating Voter’s Assessment of Verifiability Using Text Classification with Machine Learning	371
<i>Ehsan Estaji</i>	
Fault Tolerant Result Collation Scheme for Nigerian Electoral System	373
<i>Adeoye Olayinka Olaoluwa and Rafiu Ganiyu</i>	
A New Technique for Deniable Vote Updating	375
<i>Najmeh Soroush</i>	
Why does e-voting have to be perfect?	377
<i>Tamara Finogina</i>	
Electronic voting in Russia: the scrutiny of ‘e-voting’ in an authoritarian context	379
<i>Bogdan Romanov</i>	
Regulating Internet voting by analogy: does it work? Challenges and concerns for secret suffrage	381
<i>Adrià Rodríguez-Pérez</i>	
To i-vote or not to i-vote: Drivers and Barriers to the Implementation of Internet Voting	383
<i>Nathan Licht</i>	

Technological Change of ONPE, an Electoral Management Body in Peru – Voters and civil servants’ perceptions	385
<i>Pablo Hartill</i>	
Multi-dimensionality of trust towards e-voting	387
<i>Peeter Leets</i>	
Poster and Demo Session	
Towards a quantum resistant i-voting system	390
<i>Jordi Cucurull, Tamara Finogina, Aleix Amill, Noemí Folch and Nuria Costa</i>	
Microcontroller-Based Voting Client for the Estonian Internet Voting System	392
<i>Valeh Farzaliyev, Kristjan Krips and Jan Willemson</i>	
Polys Online Voting System: Lessons Learned from Utilizing Blockchain Technology	393
<i>Aleksandr Korunov, Aleksandr Sazonov and Petr Murzin</i>	
Internet Voting as Part of Mobile Cross-Border Government Services for Europe (mGov4EU)	395
<i>Robert Krimmer, Lisa Burgstaller, Tina Hühnlein, Thomas J. Lampoltshammer, Noemí Folch, Herbert Leitold, Arne Tauber, Detlef Hühnlein and Carsten Schmidt</i>	
An Overview of the Voatz Election Platform	398
<i>Nimit Sawhney, Simer Sawhney, Eric Landquist and Philip Andreae</i>	

Program Committee

Marta Aranyossy	Corvinus University of Budapest
Roberto Araujo	Universidade Federal do Pará (UFPA)
Jordi Barrat i Esteve	eVoting Legal Lab
Bernhard Beckert	Karlsruhe Institute of Technology
Josh Benaloh	Microsoft
Matthew Bernhard	University of Michigan
David Bismark	Votato
Enka Blanchard	Université de Lorraine
Stephen Boyce	IFES
Jurlind Budurushi	Cloudical Deutschland GmbH
Christian Bull	The Norwegian Ministry of Local Government and Regional Development
Susanne Caarls	Election Consultant
Gianpiero Catozzi	UNDP
Thomas Chanussot	IFES
Jeremy Clark	Concordia University
Veronique Cortier	CNRS, Loria
Régis Dandoy	Universidad San Francisco de Quito
Staffan Darnolf	IFES
Constantin Catalin Dragan	University of Surrey
Ardita Driza Maurer	self-employed; Zentrum für Demokratie Aarau/Zurich University
David Duenas-Cid	Kozminski University
Helen Eenmaa	University of Tartu
Philipp Egger	Staatskanzlei Kanton St.Gallen
Aleksander Essex	University of Western Ontario
Joshua Franklin	National Institute of Standards and Technology
Micha Germann	University of Bath
J Paul Gibson	Mines Telecom
Rosario Giustolisi	IT University of Copenhagen
Kristian Gjøsteen	Norwegian University of Science and Technology
Stéphane Glondu	INRIA
Nicole Goodman	University of Toronto
Rajeev Gore	The Australian National University
Ruediger Grimm	University of Koblenz
Rolf Haenni	Bern University of Applied Sciences
Thomas Haines	Queensland University of Technology
Thomas Hofer	Objectif Sécurité
Norbert Kersting	Univ Muenster
Steve Kremer	INRIA
Robert Krimmer	University of Tartu
Iuliia Krivonosova	Tallinn University of Technology
Ralf Kuesters	University of Stuttgart

Oksana Kulyk	IT University of Copenhagen
Olivier Leclère	State of Geneva
Leontine Loeber	University of East Anglia
Ryan Macias	RSM Election Solutions LLC
Jevgeni Malosev	E-vote-id
Beata Martin-Rozumilowicz	IFES
Ronan McDermott	mcdis
Vladimir Misev	OSCE/ODIHR
Johannes Mueller	University of Luxembourg
Magdalena Musial-Karg	Adam Mickiewicz University
Andras Nemeslaki	BME
Stephan Neumann	Landesbank Saar
Hannu Nurmi	University of Turku
Jon Pammett	Carleton University
Liisa Past	Information System Authority, Republic of Estonia
Olivier Pereira	UCLouvain
Goran Petrov	OSCE
Stéphanie Plante	University of Ottawa
Josep MÀ ^a Reniu	University of Barcelona
Peter Roenne	University of Luxembourg
P. Y. A. Ryan	University of Luxembourg
Peter Sasvari	National University of Public Service
Steve Schneider	University of Surrey
Berry Schoenmakers	Eindhoven University of Technology
Carsten Schuermann	IT University of Copenhagen
Uwe Serdült	Ritsumeikan University
Delfina Soares	University of Minho
Mihkel Solvak	University of Tartu
Oliver Spycher	Swiss Federal Chancellery
Vanessa Teague	Thinking Cybersecurity
Tomasz Truderung	University of Trier
Priit Vinkel	State Electoral Office of Estonia
Melanie Volkamer	Karlsruhe Institute of Technology / SECUSO / KASTEL Security Research Labs
Kåre Vollan	Quality AS
Roland Wen	The University of New South Wales
Gregor Wenda	BMI
Jan Willemson	Cybernetica
Peter Wolf	International IDEA
Michael Yard	IFES
Filip Zagorski	Wroclaw University of Technology
Marie Laure Zollinger	University of Luxembourg

Author Index

A	
Achenbach, Dirk	228
Amill, Aleix	383
Andreae, Philip	391
B	
Baloglu, Sevdenur	28
Barretto, Sameer	230
Benaloh, Josh	1
Bernhard, Matthew	111, 159
Blom, Michelle	143
Brightwell, Ian	201
Bronk, Chris	111
Budurushi, Jurlind	143, 297
Burgstaller, Lisa	388
Bursuc, Sergiu	28
C	
Can, Cui	380
Chanussot, Thomas	217
Chow, C. Edward	80
Chown, William	230
Costa, Nuria	383
Cucurull, Jordi	383
D	
Dalela, Asmita	327
Dandoy, Régis	77
Driza Maurer, Ardita	313
Duenas-Cid, David	60
E	
Ehin, Piret	44
El Orche, Fatima-Ezzahra	246
Emery, Shawn	80
Estaji, Ehsan	346, 362
F	
Farzaliyev, Valeh	385
Finogina, Tamara	368, 383
Folch, Noemí	383, 388
G	
Ganiyu, Rafiu	364
Gore, Rajeev	15
Grigoryan, Rafik	313
H	
Haines, Thomas	15

Halderman, J. Alex	230
Hartill, Pablo	376
Hertel, Fabian	263
Huber, Nicolas	263
Hühnlein, Detlef	388
Hühnlein, Tina	388
K	
Kasongo, Elizabeth	111
Kittelberger, Jonas	263
Koitmae, Arne	186
Korunov, Aleksandr	386
Krimmer, Robert	60, 388
Krips, Kristjan	385
Krivososova, Iuliia	60
Kulyk, Oksana	327, 329
König, Reto	329
Küsters, Ralf	263
L	
Lampoltshammer, Thomas J.	388
Landquist, Eric	391
Leets, Peeter	378
Leitold, Herbert	388
Levine, David	175
Licht, Nathan	60, 374
Liedtke, Julian	263
Locher, Philipp	329
Ludwig, Jonas	329
Löwe, Bernhard	228
M	
Marky, Karola	346
Mauw, Sjouke	28
Mayer, Andreas	96
Meyer, David	230
Monnat, Xavier	282
Murzin, Petr	386
Müller-Quade, Jörn	228
N	
Nemes, Marc	228
Nettman, Justin	313
O	
Olaoluwa, Adeoye Olayinka	364
Ostrev, Dimiter	246
Oswald, Simon	282
P	
Pang, Jun	28

R	
Ramdas, Aaditya	127
Rastikian, Simon	279
Rausch, Daniel	263
Rivest, Ron	143
Rodríguez-Pérez, Adrià	372
Roenne, Peter B.	246, 279
Romanov, Bogdan	370
Ryan, Peter Y. A.	246, 279, 346
S	
Sawhney, Nimit	391
Sawhney, Simer	391
Sazonov, Aleksandr	386
Schmidt, Carsten	388
Schuermann, Carsten	217
Schwerdt, Rebecca	228
Schürmann, Carsten	327
Solvak, Mihkel	44
Soni, Aditya	230
Soroush, Najmeh	246, 366
Stark, Philip	127, 143, 246
Stuckey, Philip J.	143
T	
Tata, Atreya	230
Tauber, Arne	388
Teague, Vanessa	143
V	
Vinkel, Pritt	186
Volkamer, Melanie	329
Vukcevic, Damjan	143
W	
Waudby-Smith, Ian	127
White, Richard	80
Willemson, Jan	186, 385
Z	
Zollinger, Marie-Laure	346

Improving Verifiability

STROBE-Voting: Send Two, Receive One Ballot Encoding

Josh Benaloh

Microsoft Research

Abstract. Numerous designs for end-to-end verifiable voting systems have been proposed in recent years to accommodate in-person voting scenarios and Internet voting scenarios, but very few have been offered to support vote-by-mail and none that are practical with current equipment and processes. This work describes a simple approach to end-to-end verifiable vote-by-mail that can be implemented with little change to existing processes. A specific architecture is described in this work, but the basic technique can also be used to enable many existing end-to-end verifiable systems to support vote-by-mail.

Since most election jurisdictions utilize some combination of in-person voting and vote-by-mail, there is great value in an approach which allows both modes to be unified into an end-to-end verifiable system that produces a single verifiable tally.

1 Introduction

Vote-by-mail (VbM) is becoming an increasingly popular mode of voting, and the importance of VbM is drastically magnified in a time of pandemic when in-person voting creates potential health risks. However, it is difficult to ensure the integrity of VbM systems, and while expert assurances can be given, there is no direct evidence provided to voters that their votes have been correctly recorded and counted.

End-to-end (E2E) verifiability is an existing technology that allows voters to confirm for themselves that their votes have, indeed, been correctly recorded and counted. Specifically, in an E2E-verifiable election, two properties are achieved.

1. Voters can confirm that their own votes have been correctly recorded.
2. Any observer can confirm that all recorded votes have been correctly tallied.

Many end-to-end (E2E) verifiable election systems include an interactive step in which voters have an opportunity to make a choice to confirm the accurate recording of their votes. While such an interaction can be instantiated in-person or even via Internet voting, such opportunities are generally not available for voters who cannot interact directly with a voting system.

The lack of support for vote-by-mail not only limits the assurances that can be offered to mail-in voters, it also poses significant challenges for many practical environments which offer an assortment of modes. Many jurisdictions do not

wish to report tallies separately for in-person voters and mail voters. However, E2E-verifiability requires the publication of a public tally to be verified. Thus, if only in-person votes are verifiable, the in-person and mail-in vote tallies must be reported separately. This can make many E2E-verifiable systems incompatible with the basic requirements of many jurisdictions.

The techniques of STROBE-Voting bridge that gap by allowing the same verifiable system to be used for both in-person and mail-in voters. STROBE-Voting is described herein as a companion to a class of E2E-verifiable voting systems following the approach of Benaloh in [4]. Such systems include Helios [1], STAR-Vote [3], and ElectionGuard [11]. However the techniques of STROBE-Voting can also be applied to pre-printed ballot designs like Scantegrity [7] and Prêt à Voter [9].

With STROBE-Voting, mail-in voters can simply hand-mark paper ballots as they do with traditional mail-in ballots. The blank ballots can be delivered by post or downloaded over the Internet. Voters who choose to do so can retain information that enables verification and then check the retained data against election artifacts posted by election administrators — usually after the close of voting.

Comparison to Remotegrity

Like STROBE-Voting, Remotegrity [15] is a design which can be applied to multiple E2E-verifiable systems to enable mail-in voting. However, Remotegrity requires code voting and scratch-off labels and only applies to pre-printed ballot systems. STROBE-Voting is much simpler — requiring no special effort on the part of voters — and is applicable to a much wider array of E2E-verifiable systems.

Limitations

STROBE-Voting is not a voting system unto itself but rather a heuristic technique that can be applied within a wide variety of systems. As such, it is not appropriate to include proofs of effectiveness outside of the context of individual systems. However, to better explain how STROBE-Voting can be used, a more detailed example is included to demonstrate how STROBE-Voting might be applied.

As with Remotegrity, a simplifying assumption is made herein that centralized ballot printers will not compromise privacy by retaining secret info about ballots they print. This is an important limitation which will be discussed in greater detail subsequently in this work. One use case eliminates this assumption by utilizing electronic ballot delivery in which secret data that could compromise voter privacy is generated on voter devices and never exposed.

2 Sample Methodology

A STROBE-Voting ballot can look almost identical to a traditional vote-by-mail paper ballot. The only additions are a very short code beside every selectable option (e.g., two letters or three digits) and a larger (32-byte) hash code at the bottom of the ballot. Voters need do nothing more than fill in ovals (or make other marks) corresponding to their selections. Any additional voter actions are completely optional.

A randomized encryption is produced for each selectable option on each ballot (the encryption changes for each new ballot). These encryptions are retained but **not** printed on ballots. Instead, the encryptions are locked by being hashed together, and this hash of all encryptions on a ballot is printed at the bottom of the ballot. Other than the ballot hash, the only vestiges of the encryptions that appear on ballots are short codes that are deterministically derived from each encryption and printed beside the corresponding options. The short *selection codes* are entirely locked by the larger *ballot code* and serve simply as identifiers for each selection. The only restriction on selection codes is that they must be unique within each contest on a ballot. If the internal randomness used to produce a ballot and ultimately the long ballot code results in duplicate short selection codes, then all or part of that randomness is replaced until all short selection codes within a ballot are unique (within each contest). **It is important to recognize that even though the selection codes are very short, there is no way to search for or deduce the selections to which they correspond without breaking the corresponding encryptions. The ease with which one can find other encryptions which produce identical short selection codes is not a security threat since the encryptions are fully locked by each ballot’s long ballot code.**

The randomness used for each encryption can be generated independently, so if a short selection code collides with another for the same contest, the randomness used in that encryption can be changed. This can be repeated as necessary to ensure that all short selection codes within any contest are unique. In theory, a single-byte short selection code can accommodate as many as 256 selections per contest. However, the time to generate suitable encryptions would slow significantly as the number of options approaches 256. Of course, the code can be lengthened as desired to accommodate larger numbers of selections, for computational efficiency, for usability in a large ballot if it is desired to have all short codes be distinct across the entire ballot, or if (as may be desirable for some scenarios) all of the randomness on a ballot is to be generated deterministically from a single seed

As with Helios, STAR-Vote, ElectionGuard, and related systems, every selectable option is encoded with exponential ElGamal [12] which facilitates easy distributed key generation and homomorphic tallying. To preview the more detailed description given below, a vector of encryptions is prepared for each selectable option in which the encryption corresponding to the selected option is an encryption of one and the remaining values are encryptions of zero. The encryption of each selection is hashed with SHA-256 and then truncated to a

single byte to form the short selection code for that selection.¹ The byte can be represented in human readable form in a variety of ways including a two-letter code, a three-digit code, a letter followed by a digit, or the common form of two hexadecimal characters.

As described above, if two of the selections within any one contest yield the same short selection code, some of the randomness used for encryption is changed to ensure that all selection codes within any single contest are distinct. Finally, all of the full encryptions used to produce all of the short selection codes are hashed together to form the 32-byte ballot hash code. As will be detailed below, within each contest, the encryptions are sorted before being hashed to avoid revealing the association between the encryptions and the individual selections to which they correspond.

Every set of STROBE-Voting encryptions is paired with an independently-generated twin set of encryptions. Each voter receives both twins and makes selections using either one of the two encryptions sets.

In its simplest presentation, each voter receives (either by post or by download) two blank STROBE-Voting ballots with the instructions that only one should be returned and the other may be used as a spare in case of errors. However, it will be described later how STROBE-Voting can be accomplished by providing only a single ballot to each voter.

After receipt of a STROBE-Voting ballot, an election jurisdiction publishes the following.

- All of the encryptions, short selection codes, and the long ballot code on each of the received ballot and its twin² (note that this step could be performed on all ballots prior to receipt)
- Non-interactive zero-knowledge (NЗИK) proofs that every encryption is either an encryption of zero or an encryption of one, that exactly one element in each vector is an encryption of one, and that every position has exactly one encryption of one across all of the vectors for a contest
- All of the short selection codes corresponding to selections the voter made on the received ballot
- Decryptions and randomness used to generate all encryptions on the ballot that is the twin to the ballot received

This information allows voters and observers to check that all ballots are well-formed. Because a voter could have cast either one of the two twin ballots, evidence of the correctness of the cast ballot (the encryptions correctly match the

¹ A hash isn't strictly necessary here. Any deterministic function of the encryption vector would suffice. One could even just use the last byte of the last encryption of the vector. The selection code merely serves as a convenient means of identifying which of the encryptions (published elsewhere and locked by the 32-byte ballot code) has been selected.

² Within each contest, the encryptions should be sorted alphanumerically according to their short selection codes to avoid revealing the association between the encryptions and the selections.

printed options) is indirectly provided by the demonstration of the correctness of its uncast twin ballot.

In effect, the published NIZKs prove to all observers that each ballot is well-formed and does not carry more (or fewer) votes than allowed. The twin ballot provides evidence to the voter that the selection codes recorded for that voter correctly match the selections made by the voter.

The well-formedness of any ballots in the published record can be confirmed by independent election verifiers. So a diligent voter need only check that the correct short codes are listed for the returned ballot (identified by its ballot hash code) and that the hash code of the twin is listed as an unsealed ballot and has matching short codes. A convenient way of displaying uncast twin ballots to voters is to publish an image of each blank twin ballot. This enables a diligent voter to easily compare this published image with the physical ballot (which remains in the voter’s possession) and confirm that the short codes match. This allows for a clean division of responsibilities wherein those voters who choose to do so can verify the correct recording of their selections by simply comparing ballots and codes — without having to perform any mathematical operations. Verification of ballot correctness can be combined with verification of election results — which can be done by any interested parties (including voters) by writing and/or executing independent election verification applications.

3 Undervotes

To accommodate the possibility that some voters may choose to not vote in some contests, each contest also has a blank or did-not-vote option and associated short selection code. This did-not-vote option is treated like any other selection, and its short selection code is derived from a vector of encryptions of zero for all ordinary options and an encryption of one associated with the did-not-vote option. If the voter does not select any of the contest options, the short selection code associated with the did-not-vote option is published for that contest as part of the information associated with the returned ballot. This prevents the published information from revealing which contests were voted on any given ballot.

In some elections, there are contests in which a voter is permitted to select more than one option. In such cases, the short selection code for each selected option will be published. Multiple did-not-vote selection codes will be provided for that contest — enough to accommodate the number of selections a voter may make for that contest. For example, in a “select up to three” contest, three did-not-vote selection codes are prepared and shown on the ballot. If a voter makes three selections in that contest, the three corresponding selection codes are published. If the voter makes only two selections, the two corresponding selection codes are published together with one of the three did-not-vote selection codes. One selection requires the publication of the one selected short code together with two of the did-not-vote selection codes, and no selections requires the publication of all three did-not-vote selection codes.

4 A Detailed Example

To better describe the approach, a small example is included below. It is assumed that a single public encryption key has been produced by combining (using a threshold key distribution) ElGamal public keys generated independently by a pre-determined set of election trustees.³ Encryption using public key is denoted by \mathcal{E} .

Suppose that an election is to be conducted with a single “vote for one” contest featuring candidates Alice, Bob, and Carol. To accommodate the possibility that a voter might not choose any candidates, a fourth pseudo-candidate *did-not-vote* is added and treated just like any actual candidate.

The structure of the encryptions of the votes in is shown in the following table.

Candidate voted for	Vote (vector of encryptions)	Sample selection code
Alice	$\langle \mathcal{E}(1), \mathcal{E}(0), \mathcal{E}(0), \mathcal{E}(0) \rangle$	Q4
Bob	$\langle \mathcal{E}(0), \mathcal{E}(1), \mathcal{E}(0), \mathcal{E}(0) \rangle$	D6
Carol	$\langle \mathcal{E}(0), \mathcal{E}(0), \mathcal{E}(1), \mathcal{E}(0) \rangle$	L7
did-not-vote	$\langle \mathcal{E}(0), \mathcal{E}(0), \mathcal{E}(0), \mathcal{E}(1) \rangle$	R9

The selection codes are generated as one-byte truncations of SHA-256 hashes of each associated vector of encryptions. The short selection code for each vote is printed beside the oval used by the voter to select a candidate. Other than the inclusion of a single long ballot code on each printed ballot (described further below), the addition of a short selection code beside each oval is the only observable change to a physical ballot.

The associated physical ballot may look something like the following.

Alice Q4
 Bob D6
 Carol L7
 none R9

Ballot code: XC3K0-A21BM-8WP8Q-MWQ6E-UYW9Y-ZPBL5-93LRE-M3J62-MJ1W7-87DYF

The encryptions associated with each ballot are made public as part of the election. However, they are re-ordered (sorted according to their short selection codes — which are guaranteed to be distinct) to avoid revealing which encryption corresponds to which candidate.

The public information associated with a ballot is shown below.

³ There is nothing novel about the generation and sharing of exponential ElGamal keys to produce a single ElGamal key for which threshold decryption is possible. This is done in STAR-Vote, ElectionGuard, and (without the threshold decryption capability) Helios. Since the details are not relevant, they are not included herein.

Vote (vector of encryptions)	Sample selection code
$\langle \mathcal{E}(0), \mathcal{E}(1), \mathcal{E}(0), \mathcal{E}(0) \rangle$	D6
$\langle \mathcal{E}(0), \mathcal{E}(0), \mathcal{E}(1), \mathcal{E}(0) \rangle$	L7
$\langle \mathcal{E}(1), \mathcal{E}(0), \mathcal{E}(0), \mathcal{E}(0) \rangle$	Q4
$\langle \mathcal{E}(0), \mathcal{E}(0), \mathcal{E}(0), \mathcal{E}(1) \rangle$	R9

Ballot code: XC3K0-A21BM-8WP8Q-MWQ6E-UYW9Y-ZPBL5-93LRE-M3J62-MJ1W7-87DYF

The published information associated with each ballot also includes the following non-interactive zero-knowledge proofs (using the Fiat-Shamir heuristic [13]).

- Every encryption is an encryption of either zero or one (Cramer-Damgård-Schoenmakers disjunctive technique [10] applied to Chaum-Pedersen proofs [8] of encryption of zero and encryption of one).
- Every row is shown to have exactly one encryption of one (Chaum-Pedersen proof that the product of the encryptions in each row is an encryption of one).
- Every column is shown to have exactly one encryption of one (Chaum-Pedersen proof that the product of the encryptions in each column is an encryption of one).

All of the encryptions for this contest are then hashed in the sorted order shown here using SHA-256 to produce the long ballot code that is printed on the ballot. (In a ballot with multiple contests, the encryptions of each contest are hashed together in this way — preserving contest order — to form a single ballot code).

The long ballot code serves to lock all of the encryptions on the ballot. Without this long code, a malicious election administrator could substitute different encryptions that produce identical short selection codes in some or all contests (recall that the actual encryptions are *not* displayed on the ballot). These distinct encryptions could correspond to selections that do not match those printed on the ballot (for instance, switching two candidates) and allow votes to be thereby changed. Hashing all of the encryptions into the long ballot code makes it infeasible for an attacker to change any of the encryptions without causing the hash to no longer match the long ballot code (which *is* displayed on the ballot).

At the conclusion of an election, all of the selected encrypted votes are homomorphically combined to form encrypted tallies which are then verifiably decrypted to produce verified tallies — as in schemes with similar homomorphic tallying.

While the NIZK proofs above demonstrate that each ballot — and therefore each encrypted vote cast on each ballot — is well-formed, it does nothing to ensure that the short selection codes are matched against the correct candidates on each ballot. This is the function of the twin ballots which will be fully decrypted after the conclusion of voting.

5 Single-Ballot Variations

Whether it be due to cost or discomfort, some jurisdictions may prefer to not provide two blank ballots to each voter. The approach described above can be executed on a single-ballot design as follows.

Postal Delivery of Blank Ballots

Instead of a single short selection code beside each selectable option, two independent selection codes are printed (e.g., Q4-M1). A single ballot code computed as the hash of the two (implicit) ballot codes is printed at the bottom of this ballot. A voter may then indicate which of the two short code sets (left or right) are to be published as cast and which are to be fully decrypted. This choice may be made explicitly (by having the voter mark a choice directly on the ballot) or implicitly by another means (such as using the way that a ballot is folded⁴ or the orientation in which it is placed in its return envelope⁵ as a selector as in [5]).

The challenge here is usability for voters. While an explicit choice may be more natural for a voter who wishes to verify (e.g., “I chose to have the left codes revealed and the right codes used to record my vote.”), it is difficult to present this choice to voters in a way that does not risk confusing voters who have no interest in verification. An implicit choice eliminates the risk of confusing uninterested voters, but it can complicate the process for both election administrators and voters who wish to verify their ballots. For instance, if ballot orientation is used, verifying voters would need to carefully note the orientations in which they insert their ballots into their envelopes, and election administrators would need to carefully note the orientations of received ballots and to take different actions based upon these orientations. While this can meet the definition of E2E-verifiability, it stretches credulity to suggest that this could be performed at scale by enough voters and jurisdictions to have a meaningful impact.

For these reasons, it may be preferable to use the simpler approach of providing each voter with two entirely distinct ballots and instructing that only one be marked and returned.

Electronic Delivery of Blank Ballots

If blank ballots are delivered to voters electronically, voters can “request” multiple ballots and return only one. In this case, unreturned ballots can be opened as though they were delivered by post. But it is no longer necessary to provide two ballots to each voter. Instead, the mere threat that a requested ballot may not be cast renders improper ballots detectable.

⁴ Tri-fold ballots have two natural distinct foldings — putting either the top third or the bottom third between the remaining two thirds.

⁵ A typical rectangular ballot can naturally be inserted into an envelope in one of four orientations.

This process can be further improved if blank ballots are generated by voters themselves on their own devices. Uncast ballots can be challenged for integrity either by having the printing device open the encryptions directly or by voters returning blank ballots along with cast ballots for central decryption. When done this way, no device outside of the control of a voter will know the associations between ballot selections and short codes on any cast ballot. This gives voters far greater privacy guarantees than with centrally-printed ballots.

6 Usability

From the perspective of voters, there are many benefits to the basic approach of mailing two blank ballots.

- The explanation of one ballot being a spare is quite simple and natural. It seems less likely to cause confusion than many alternatives.
- Voters needn't be burdened with the question of which set of encryptions to use.
- Simple disambiguation rules can be used by election officials to disambiguate cases when voters return both ballots — avoiding disenfranchisement of these voters.
- Interested voters who want to check the accuracy of their unreturned ballots can retain the entire ballots to facilitate these checks.
- A copy of the long ballot code can be printed onto a tear-off strip immediately below (or above) the copy that remains on the ballot.⁶ This facilitates an easy check by voters prior to removing the strip.

Single ballot variants may be easier and more economical for election administrators, and they do not hinder voters who are not interested in verifying their ballots. But voters who want to verify the integrity of their ballots will need to somehow make (and record for their own use) decisions about which set of ballot codes to use and record some or all of the unused ballot codes and associated selections. This seems like quite a lot to ask — even from a diligent voter.

There would be great temptation — especially in the single ballot case — for a voter to retain a photograph of a ballot before returning it. But this should be discouraged because it could facilitate coercion (see the discussion of coercion and vote-selling in section 8).

7 Hybrid Voting Systems

The encryption used in the above description of STROBE-Voting is compatible with that used in Helios, STAR-Vote, and ElectionGuard.

This is especially important because many jurisdictions that collect some votes in-person and others by mail do not wish to report tallies separately across modes. In such jurisdictions, the lack of an E2E-verifiable VbM method can

⁶ The idea of a tear-off strip was suggested by an anonymous reviewer.

inhibit the use of E2E-verifiability for in-person voting — even if only a small portion of votes are received by mail. STROBE-Voting can bridge this gap and thereby help promote the adoption of E2E-verifiable solutions.

Hybrid modes do impair statistical analysis since the models of ballot challenges are different. In Helios, STAR-Vote, and ElectionGuard, there is no limit to the number of ballots that can be challenged or spoiled by a single voter — although challenging of ballots is usually rare. With most instantiations of STROBE-Voting, there is effectively one challenged ballot for every ballot cast. A much larger proportion of VbM ballots than in-person ballots will be challenged, but a highly-skeptical individual voter has no means to raise the direct confidence of an individual cast ballot beyond 50% unless the voter is offered some means to request additional blank ballots.

While this does not seem to be a problem in practice, it does make it more difficult to calculate the assurance level of an election. However, since there would likely be a far greater portion of challenged ballots in the VbM votes, the addition of STROBE-Voting to an in-person E2E-verifiable solution will almost certainly not reduce the level of assurance in the integrity of the tally.

8 Attacks

Since this work is intended primarily to describe a concept that may serve as a component of other systems, rigorous security proofs are not presented. However, some potential attacks are addressed here.

Collision of short selection codes

It might seem as though the short selection codes create a potential weakness. However, longer codes would not provide any additional security. The security is derived from the encryptions of the votes and the hash of these encryptions that forms the full-length ballot code on each ballot. The short selection codes serve only as simple deterministic identifiers to distinguish between the various encryptions. While it would not be difficult to create alternate ballots with matching short selection codes that apply to different selections, there would be no benefit in doing so. The association between the short selection codes and the selections they represent is determined entirely by the long ballot code for which it is infeasible to find collisions.

Small perturbations

Since each voter receives two ballots (or two sets of codes on a single ballot), a malicious election administrator could send a single voter one good ballot and one corrupted ballot (or one corrupted code set) and hope that the voter casts the corrupted ballot (or code set). In so doing, the malicious administrator would be exposed to a 50% risk of its fraud being revealed, but there would be a 50% chance of successfully corrupting a single ballot. The chance of corrupting n

ballots without being revealed would be 1 in 2^n , so this is not likely to be a very fruitful attack.

A system can also be designed to allow particularly suspicious voters to receive additional ballots — thereby enabling the voter’s probability of detecting fraud to be raised as far as desired (within practical limits).

Clash attacks

Clash attacks involve an authority providing identical ballots to multiple voters who might be likely to cast identical votes. The authority might then record only one of the identical ballots. The intent would be to allow each voter to be able to verify the correct recording of a vote without revealing to the voter that this is a duplicate.

However, as with other E2E-verifiable designs following the “cast or spoil” approach of [4], clash attacks are not a concern. The ability to spoil a vote or ballot can reveal an attempt to create a clash in the same way that it can reveal an attempt to create an incorrect encryption — and with similar probabilities and risks to a malicious system. In the case of incorrect encryptions, the system will be revealed as malicious if a voter chooses to spoil an incorrectly encrypted vote. In the case of attempted clashes, the system will be revealed as malicious if one voter chooses to cast a particular ballot while another chooses to cast the twin. In this latter instance, the first voter would expect to find the associated ballot code on the list of cast ballots while the other would expect to find the same ballot code on the list of uncast/spoiled ballots. Putting the same ballot code on both lists would immediately implicate the voting system as malfeasant.

Note that the device that created an errant ballot may not be required to be involved in its spoiling — or even be aware of its spoiling. The election trustees who share the private key to decrypt election tallies can decrypt any spoiled ballots on their own. So failure to open a spoiled ballot would be a direct and public impeachment of the integrity of the election.

Attempting to vote two ballots

Different jurisdictions manage their VbM ballots in vastly different ways. In some instances, blank ballots are considered a controlled resource and putting two ballots into the hands of voters might create a risk of double voting.

However, in many jurisdictions, the controls are placed at the time ballots are received. For example, in the author’s home state of Washington in the U.S., voters can easily download and print as many blank ballots as they wish. Upon receipt of each completed ballot, election officials check a signature and eligibility (including that no prior vote from that voter has been received in that election) before accepting and counting the ballot.

Clearly a two-ballot approach is a better fit for a jurisdiction that does not place controls on blank ballots. A jurisdiction in which a vote on any legitimate ballot will be accepted and processed will presumably fair better with a single-ballot variation.

Susceptibility to coercion and vote-selling

Like any unsupervised mode voting, vote-by-mail can subject voters to coercion and allow vote-selling more easily than supervised modes of voting.⁷ No attempt is made here to justify or advocate for VbM. However, it is reasonable to find ways to make VbM methods as strong as possible. Adding E2E-verifiability to VbM doesn't eliminate the coercion and vote-selling threats, but it nevertheless improves integrity by enabling verifiability of correct recording and counting of votes. This allows votes cast by mail using STROBE-Voting to be combined with votes cast by other modes to produce a single unified verifiable tally.

For better or worse, VbM is widely used and its prevalence is growing. VbM is stronger with STROBE-Voting than without. That said, STROBE-Voting can provide another vector for coercion and vote-selling beyond ordinary VbM. The combination of a genuine photograph of a ballot that has been cast together with the published short codes for that ballot offers a strong indication of the contents of a recorded vote. This can be mitigated by a variety of means — including the use of image editing and other tools — some of which may be provided by election administrators themselves. It should also be noted that this form of coercion is only possible before a vote is cast — a voter who, after following the instructions and casting a proper ballot, is asked to reveal the ballot contents will have no means to do so. This is why more direct means of verification are not offered to voters. However, it is important to maintain context and understand that with or without STROBE-Voting, remote voters can either choose to have or be coerced into having an observer during the voting process or to simply surrender a ballot entirely.

Privacy implications of centralized ballot printing

As with numerous other approaches to make paper delivery of blank ballots E2E-verifiable (e.g., [2], [5], and [15]), centralized printing of blank ballots may enable a central entity to retain information that can later be used to compromise voter privacy. This concern can be mitigated with indirect approaches like code voting ([6]) at a significant cost to usability.

It is important to note that this threat can be eliminated entirely if ballots are delivered electronically and generated privately on voter devices as described at the end of section 5.

9 Conclusions

STROBE-Voting is a technique that is simple both to understand and to implement. It enables E2E-verifiable election systems designed for in-person voting to be extended to serve mail-in and other remote voters.

⁷ Juels, Catalano, and Jakobsson ([14]) offers a possible approach to deter vote-selling and coercion in remote voting, but the usability creates practical challenges.

While the acronym *STROBE* is only vaguely suggestive of the technique used in which some, but not all, of the ballots are “illuminated” by decryption, the methods have the potential to have an important impact on the use of E2E-verifiability and to promote new research to make E2E-verifiability even simpler to deploy and use.

Although the value of E2E-verifiability today is clear in an environment where numerous voters are questioning whether their votes have been correctly counted, there is ample room for new research and developments in areas such as dispute resolution, coercion-resistance for remote voting, and privacy enhancements for remote printing. STROBE-Voting adds another tool that can be used to enhance the applicability and value of E2E-verifiability.

Acknowledgements

The author would like to thank Jan Willemson and several anonymous reviewers for providing many helpful comments and suggestions that greatly improved this work.

References

1. **Adida, B.** *Helios: Web-based Open-Audit Voting*. Proceedings of USENIX Security Conference. August, 2008. San Jose, CA.
2. **Adida, B** and **Rivest, R. L.** *Scratch & Vote: self-contained paper-based cryptographic voting*. Proceedings of WPES 2006 – ACM Workshop on Privacy in the Electronic Society. October, 2006. Alexandria, VA.
3. **Bell, S., Benaloh, J., Byrne, M. D., DeBeauvoir, D, Eakin, B., Fisher, G., Kortum, P., McBurnett, N., Montoya, J., Parker, P., Pereira, O., Stark, P. B., Wallach, D. S., and Winn, M.** *STAR-Vote: A Secure, Transparent, Auditable, and Reliable Voting System*. Proceedings of Electronic Voting Technology Workshop / Workshop on Trustworthy Elections. August, 2013. Washington, DC.
4. **Benaloh, J.** *Simple Verifiable Elections*. Proceedings of Electronic Voting Technology Workshop. August, 2006. Vancouver, BC.
5. **Benaloh, J., Ryan, P. Y. A., and Teague, V.** *Verifiable Postal Voting*. Proceedings of 21st Cambridge International Workshop on Security Protocols. March, 2013. Cambridge, UK.
6. **Chaum, D.** *SureVote: Technical Overview*. Proceedings of Workshop on Trustworthy Elections. August, 2001. Tomales Bay, CA.
7. **Chaum, D., Carback, R., Clark, J., Essex, A, Popoveniuc, S., Rivest, R. L., Ryan, P. Y. A., Shen, E., and Sherman, A. T.** *Scantegrity II: End-to-End Verifiability for Optical Scan Election Systems using Invisible Ink Confirmation Codes*. Proceedings of Electronic Voting Technology Workshop. July, 2008. San Jose, CA.
8. **Chaum, D. and Pedersen, T. P.** *Wallet Databases with Observers*. Proceedings of Crypto '92. August, 1992. Santa Barbara, CA.
9. **Chaum, D., Ryan, P. Y. A., and Schneider, S.** *A Practical Voter-Verifiable Election Scheme*. Proceedings of European Symposium on Research in Computer Security. September, 2005. Milan, Italy.

10. **Cramer, R., Damgård, I., and Shoenmakers, B.** *Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols*. Proceedings of Crypto '94. August, 1994. Santa Barbara, CA.
11. Microsoft **ElectionGuard** Software Development Kit is available at <https://github.com/microsoft/electionguard>. (See also <https://blogs.microsoft.com/on-the-issues/2019/09/24/electionguard-available-today-to-enable-secure-verifiable-voting/>.)
12. **ElGamal, T.** *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*. IEEE Transactions on Information Theory, vol. 31, 1985.
13. **Fiat, A. and Shamir, A.** *How to Prove Yourself: Practical Solutions to Identification and Signature Problems*. Proceedings of Crypto '87. August, 1987. Santa Barbara, CA.
14. **Juels, A., Catalano, D., and Jakobsson, M.,** *Coercion-Resistant Electronic Elections*. Proceedings of WPES 2005 – ACM Workshop on Privacy in the Electronic Society. November, 2005. Alexandria, VA.
15. **Zagórski, F, Carback, R., Chaum, D., Clark, J., Essex, A., and Vora, P.** *Remotegrity: Design and Use of an End-to-End Verifiable Remote Voting System*. Proceedings of the International Conference on Applied Cryptography and Network Security. June, 2013. Banff, AB.

Improved Verifiability for BeleniosVS

Thomas Haines¹ and Rajeev Goré^{2*}

¹ Australian National University, Canberra, Australia

² Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Abstract. The BeleniosVS electronic voting scheme offers an attractive mix of verifiability and privacy properties. Moreover, using the ProVerif protocol-verification tool, BeleniosVS has automatic machine-aided analysis of (end-to-end) verifiability in 96 different threat models with the machine-aided analysis finding proofs in 22 cases and finding attacks in the remaining 74 cases. The high number of threat models covered by ProVerif delivers a much richer security analysis than the norm.

We revisit the BeleniosVS scheme and propose several refinements to the ProVerif security model and scheme which increase the number of threat models in which the scheme has verifiability from 22 to 28. Our new ProVerif security model also implies end-to-end verifiability but the requirements are easier to satisfy. Interestingly, in all six improvements, both the changes to the security model and one or more changes to the scheme are necessary to prove verifiability.

Keywords: Verifiability · Machine-checked proofs · ProVerif · BeleniosVS

1 Introduction

Secure electronic voting is a difficult security problem with numerous competing constraints. The key approach to securing electronic elections is so-called “end-to-end verifiable electronic voting” which is usually broken down into three sub-properties: namely, cast-as-intended, collected-as-cast and counted-as-collected. Cast-as-intended captures the idea that the encrypted ballot the voter casts contains the vote she intended. Collected-as-cast captures the idea that the cast ballot was collected by the election authority without tampering. Counted-as-collected captures the idea that the collected ballots are properly counted. Our work focuses on the end-to-end verifiability of the BeleniosVS voting scheme [6], which is the culmination of a line of schemes starting with Helios [1].

BeleniosVS, like all voting schemes, should provide verifiability and privacy under reasonable assumptions. Privacy captures the intuition that nothing more should be leaked about the voter’s ballot than can be discerned from the outcome of the election; for the rest of this paper we will eschew further discussion

* Rajeev Goré acknowledges the support of the National Centre for Research and Development NCBR, Poland, under the PolLux/FNR-CORE project STV (POLLUX-VII/1/2019).

of privacy. BeleniosVS has end-to-end verifiability under certain trust assumptions; that is, under certain trust assumptions we can be assured that for every honest voter, the encrypted ballot in the tally which belongs to her (because the accompanying signature verifies with her verification key) does indeed decrypt to the ballot she believes she cast. As we have previously said, BeleniosVS has been analysed in 96 different threat models, 22 of which have verifiability which makes providing a simple description of the assumptions problematic.

Informally, BeleniosVS achieves end-to-end verifiability under certain assumptions as follows:

1. it achieves cast-as-intended because a voter can check that the cast ballot contains the vote she intended under the assumption that either the auditing device or registrar—which we will formally introduce in section 2.1—is honest;
2. it achieves collected-as-cast since the voter can check that the ballot posted by the voting server was the one she cast, under the assumption that the registrar is honest; and finally
3. it achieves tallied-as-collected because a voter can check the zero-knowledge proofs produced by the talliers which ensure the published ballot is correctly decrypted and tallied.
4. moreover, the authentication measures (signatures) prevent the ballot box from being stuffed unless the registrar is corrupt.

Any claimed verifiability and privacy properties must assume some underlying security model. There are two prominent such models called the symbolic (Dolev-Yao [11]) model and the computational model (more specifically, most voting schemes are analysed in the Random Oracle Model (ROM) [3]). The symbolic model assumes that the underlying cryptographic primitives are black-boxes (perfectly secure), and the attacker can only use these primitives as such. On the other hand, the computational model assumes that all data types are bit-strings and all computation is done by probabilistic polynomial-time Turing machines. The computational model captures more attacks but is harder to reason about; both models miss certain classes of attacks (such as timing attacks).

The authors of Belenios provided mathematical proofs of the correctness of the claimed verifiability and privacy of Belenios using ProVerif [5], an automatic theorem prover for verifying properties of security protocols which uses the symbolic model. In contrast, tools such as Coq [4] and EasyCrypt [2] work in the computational model. We also use ProVerif for our work.

ProVerif is sound wrt the symbolic model: that is, if it finds an attack or concludes there isn't one then we can be sure that these conclusions are true in the symbolic model. However, one challenge in using ProVerif for complicated schemes is its incompleteness: that is, there are queries on which ProVerif may not terminate, and hence be unable to decide whether or not there is an attack, in which case, we learn nothing by using ProVerif. Due to this, BeleniosVS [6] was not able to be directly proved end-to-end verifiable in ProVerif:

rather the authors introduced sufficient conditions for verifiability; these conditions are checked by ProVerif. Both the sufficient conditions and the definition to which they refer have the advantage over prior definitions of enforcing vote eligibility; that is the definitions prevent the stuffing of the ballot box with unauthorised votes. However, this comes at a high cost because the definition no longer distinguishes between systems where attacks are detected and system where attacks are not detected but rather deems both to be insecure (Sec. 3). We present new related conditions which do distinguish between detected and undetected attacks but at the cost of not capturing ballot stuffing; any scheme proved secure under our new conditions should separately be proved to be free of ballot stuffing. It remains an open problem to devise a definition which combines the strengths of both the original definition in [6] and ours.

In general, theorems about verifiability (in ProVerif) are either about the unreachability of certain “undesirable” states or about correspondence assertions; we will be interested in correspondence assertions. These are of the form “if event e has occurred then event e' has occurred previously”. For example, a state where the voter n believes she voted for a particular candidate c should imply that the ballot published next to her name is for candidate c . To encode this into ProVerif, let $\text{VERIF}(n, c)$ denote that voter n believes she voted for candidate c and let $\text{TALLY}(n, c)$ denote that the ballot published for voter n decrypts to candidate c . Then we model our previous claim by saying that $\text{VERIF}(n, c) \Rightarrow \text{TALLY}(n, c)$; that is if a voter believes a certain vote is cast then it was. ProVerif attempts to prove that this correspondence holds for all executions of the protocol or alternatively to find an execution which does not satisfy the correspondence. (For those interested in the technicalities we recommend reading the ProVerif manual [5]).

ProVerif identifies many attacks on BeleniosVS in the ProVerif security model, but some of these attacks are not “real” because they cannot manifest in the implementation of BeleniosVS; the false “attacks” identified by ProVerif would have been detected by the voter during the voting phase. For example, the adversary votes on behalf of the voter (without detection) but does not fake the voting server’s acknowledgment when the voter tries to vote (the lack of which the voter will detect). The six threat models in which we improve on the previous analysis are all interesting since they require both our changes to the security model and to the scheme. Specifically in all six, ProVerif correctly identified that the scheme was insecure (both in the ProVerif model and with respect to end-to-end verifiability); however, the attacks ProVerif found would have been detected by the voter and hence are not proper attacks on verifiability. If we introduce our improved security model ProVerif still finds attacks (albeit more complicated than the previous attacks). The more complicated attacks were the inspiration for our improvements to the scheme which ultimately increase the security.

1.1 Contribution

- We identified that the definition of end-to-end verifiability in [6] does not meaningfully capture verification failures; this results in a definition which is much closer to integrity than verifiability. The use of the definition to analysis verifiability returns false attacks which obscures the real attacks and hinders the analysis of improvements to the scheme.
- We refined BeleniosVS to include a defence-in-depth approach, using various overlapping independent means to mitigate attacks. Even if the voter chooses not to verify the voting sheet, the voting device and server (if honest) will still perform certain checks and notify the voter whether or not they (the voting device and server) believe the ballot was cast successfully. Consequently, in our refined scheme, verifiability may hold even if the adversary can cast ballots on behalf of the voter, since the attack will still be detected (depending on the exact threat model).
- The ProVerif encoding of cast-as-intended in [6] required the property to hold even if the verification checks failed. We changed the property so that no guarantees are provided if verification fails. This weakens the security requirement while preserving the sufficiency for verifiability (though not for the definition of verifiability in [6])
- We refined the channel analysis to be more consistent. Our refined analysis highlights the importance of keeping basic protections like TLS and digital certificates.

Our more nuanced sufficient conditions ensure that the attacks found by ProVerif work against the real scheme and are not artefacts of our model and verifiability definition. These attacks highlighted edge case vulnerabilities which we prevent by introducing simple mechanisms. Overall, our refinements to the security model and scheme increase the number of threat models in which the scheme has verifiability from 22 to 28.

2 Background

In Helios [1], the election authority sends an email to each eligible voter containing instructions on how to construct a valid ballot. The voter constructs her ballot on her personal computing device, which encrypts the ballot using the public key of a special voting server run by the election authority, it also constructs (zero-knowledge) proofs that the encrypted ballot is well-formed wrt the instructions. She sends her ciphertext to the voting server over the internet, thus the voting server collects the (encrypted) ballots and proofs from all voters. These encrypted ballots and proofs are published by the voting server, and the published ballots are decrypted and tallied by a group of (physically distributed) authorities called the talliers. While the voting server has some means (normally email address) to authenticate votes from eligible voters, there is no external mechanism (such as signatures) to authenticate that the ballots cast actually came from eligible voters.

The Belenios family of voting schemes originated with Belenios [10] which is a more secure version of Helios [1] with better authentication of ballots. Specifically in Belenios, each voter has a signing key—and verification key—which the voters use to sign their encrypted ballot and proofs. Anyone can now check that the submitted ciphertext came from an eligible voter by verifying the signature with the voter’s verification key. However, a coercer could simply demand that the voter divulge the randomness used in the encryption of their vote, enabling the coercer to learn how that voter voted once the ballots are published.

The next version, BeleniosRF [7], replaced the signatures and proofs with rerandomisable alternatives, which allow the voting server to randomise the ciphertexts, proofs, and signatures before publishing. Now, the coercer cannot be certain that the randomness is what the voter claims it to be, so the voter cannot be coerced to prove how she voted using this tactic. BeleniosVS [6] extends BeleniosRF by including a cast as intended mechanism in which the voter can check that the cast ciphertext decrypts to the vote she intended to cast. However, due to the rerandomisation, she cannot check the ballot posted by the voting server was the one she cast. Being unable to perform a direct collected-as-cast check in either scheme complicates the end-to-end verifiability claims of these schemes.

2.1 BeleniosVS

For completeness we will introduce the BeleniosVS protocol here; the original description can be found in [6]. We will avoid giving in full all the technical details that are not relevant to our contributions.

The protocol involves seven distinct primary entities.

The election administrator publishes the parameters of the election including the list of eligible voters and candidates.

The registrar generates the voter credentials and voting sheets and issues them to the voters. A voter’s voting sheet contains encryptions—of the candidates—which are signed using her credentials.

The voter uses her voting device to cast a ballot from the voting sheet she received from the registrar with the password she received from the voting server. Optionally she can use her auditing device to verify the correctness of the voting sheet.

The voting device receives the password and ballot from the voter which it submits to the voting server.

The auditing devices check the correctness of the voting sheet.

The voting server performs some basic checks and if they pass, posts the ballot to the bulletin board.

The bulletin board is public and can be audited by anyone.

The tallying authorities takes the ballots from the bulletin board from which they compute and publish the result.

The system proceeds in the following three phases.

Setup phase: The election administrator publishes the election parameters.

Then the tallying authorities generate the encryption public key and secret keys. The registrar then prepares the voting sheets by generating signing credentials (keys) for each voter; it, then for each voter, encrypts the candidates using the encryption public key and signs each ciphertext with the voter’s signing credential. The registrar also includes, for later auditing, on the voting sheet the signing credential and the randomness used to encrypt the candidates. It then sends the voting sheets to the voter.

Voting phase: The voter receives the voting sheet through some channel other than her voting device. She can optionally get her auditing device to check the correctness of the voting sheet. The voter also receives a password from the voting server. The voter, using her voting device authenticates to the voting server using her password. She then scans the signed ciphertext corresponding to her preferred candidate into her voting device. The voting device submits this ciphertext to the voting sever. The voting server checks the validity of the signature and that no ballot had already been received from that voter. If both checks pass the voting server post the ciphertext to the bulletin board and sends an acknowledgment to the voter by way of the voting server.

Tallying phase: The tallying authority decrypts the contents of the bulletin board and publishes the result with appropriate proofs.

3 Verifiability

The authors of BeleniosVS extending on [8,9] give a pen-and-paper definition of end-to-end verifiability in the symbolic model (Sec. 3.6 of [6]). The informal variant of definition from [6] says that the election result should contain only

- the votes from the voters who successfully performed the verification specified by the protocol;
- some of the votes from the voters who did not make any verification;
- and at most k dishonest votes where k is the number of dishonest voters (under the control of the attacker).

A carefully reader will have noted that the list above makes no mention of what happens to votes from voters who attempted to verify but are unsuccessful, that is for whom the verification checks failed. One would assume that such votes cannot be included in the tally (except as part of the dishonest votes). However, the formal definition given in [6] considers detecting cheating as equivalent to not checking at all and so such voters are put in the second group (which was informally described as not making any verification).

The formal (pen-and-paper) definition is defined using events, which denote the progress of the protocol for each voter.

Voter(id, cred, l): voter id is registered with credential $cred$. The label l records if the voter is honest or dishonest.

Voted(id, v): voter id has cast a vote for v

Goting-to-tally(id, cred, b): ballot b has been recorded on the bulletin board by the voting server. According to the voting server, b is associated with voter id with credential $cred$.

Verified(id, v): voter id has voted for v and has successfully performed all required checks. She should be guaranteed her ballot will be properly counted for v .

The description of Verified is misleading because it is used in [6] even for voters who do not audit. We think the use is correct but the description should be updated, since even if the voter does not make any additional checks it still models that their device told them the ballot was correctly recorded. An observant reader will have noted that events provide no way to denote a voter who performs the security checks and detects a problem; for this reason we added a new event. However, due to a technicality in the current model a voter who receives an error during a verification check halts; because of this, at present, all voters who finish have successfully verified. (Future work may wish to resolve this issue but we chose not to because it would require us to change already defined events.)

Finished(id, cred): voter id with credential $cred$ has finished voting.

3.1 Changes to the ProVerif model

In this section we will discuss our changes to the model which are motivated by the investigation of the ProVerif model in [6] and the attacks that ProVerif finds. In [6, page 379] Cortier et al. write “*in all cases...[which are marked as insecure]..., we found a real attack.*” The statement is completely accurate but at least for us was confusing. The statement means that there are attacks against BeleniosVS in the ProVerif model but it does not mean that the attacks would go undetected in reality. In other words, we initially incorrectly interpreted the statement to mean that the authors **validated** the model by checking the attacks found by ProVerif were attacks on the real system. In fact, their ProVerif model finds various spurious attacks and we make several refinements to the model to remove them. We stress that though the attacks are spurious the system (without changes) is still insecure in the relevant threat models. We have checked that all threat models in which ProVerif finds an attack in our enhanced model have attacks against the real system. It is important to remove these spurious attacks so that ProVerif can find attacks that would work on the deployed scheme. Once ProVerif found the useful attacks we were able to update the scheme with countermeasures and have ProVerif validate that the countermeasures worked.

Our more complicated security conditions and scheme require us to use more recent features of ProVerif which have only been added after the ProVerif version used in [6]. Even so when considering cases where the register is corrupt, and hence, the public credential may not be unique, we had to assume an extra axiom to make it terminate. The axiom states that only one ballot per name is added

to the ballot box (if the server is honest) which is true since the server enforces the condition (Refinement 4). (It might be possible for ProVerif to check this property and so remove the axiom but we were unable to do so).

One of the difficulties we faced in this work was deciding how to interpret various ambiguities in the original paper [6]. We have tried to make use of the ProVerif files to clarify the authors’ intent but with somewhat limited success; the models of the different properties in ProVerif capture slightly different variants of BeleniosVS. For example, the comments in the verifiability model suggest the channel between the voting device and voting server is authenticated but not encrypted. Whereas the receipt-freeness definition assumes the channel is secret. We have attempted to unify these to the largest extent possible.

What does it mean to verify and whose voter is it anyway? In e-voting we tend to think of verification by the voter as Boolean, the voter either does or does not (choose to) verify (her ballot). To some extent, in the original ProVerif analysis of BeleniosVS analysis, and to a greater extent in our refinement, such verification is a sliding scale which captures different actions in different threat models. Even a voter who takes no explicit verification steps is still modelled as believing her intended vote was cast as intended (unless her computer tells her something went wrong). So, even for a voter who does not take any extra verification steps, her device (if honest) will report an error unless all of the checks the device performs pass.

It is standard in the analysis of e-voting voting schemes to split voters into honest and dishonest. However, in our refined threat model of BeleniosVS, we are confronted with voters whose credentials are completely leaked but who make no attempt to cooperate with the adversary; should they be modelled as honest or dishonest? The issue is more nuanced than it might first appear and we adopt the approach of considering voters honest if they attempt to vote (and complain if they fail) but dishonest if they do not attempt to vote themselves or do not complain when voting fails. In doing so we have eliminated the earlier category of “voters who did not make any verification”, all voters either attempt to vote and report any error raised by their device, or are considered dishonest.

3.2 The sufficient conditions were not necessary.

As we have noted, the description of Verifiability in Sec. 3.6 of BeleniosVS paper [6] does not explicitly say that the properties do not have to hold if verifiability checks fail. Indeed, the ProVerif definition of cast-as-intended requires the properties to hold even if verification fails.

Refinement 1 *We changed the ProVerif cast-as-intended definition so that the condition need only hold if the voter doesn’t complain.*

In the ProVerif security model for BeleniosVS each property has two variants, one which identifies the voter by name and another which identifies the voter based on credential. This is necessary because names are only a reliable

identification if the voting server is honest, and analogously credentials are only reliable if the registrar is honest. We updated both the cast-as-intended definitions but for brevity only describe the change on the name based definition since they are nearly identical.

Cast as intended (ID-based)-Old: For every ballot in the tally, belonging to name n for choice v either the voter n is dishonest or the voter is honest and intended to cast a ballot for v . (We denote by \diamond variables which may take any value.)

$$\text{GOING_TO_TALLY}(n, \diamond, \diamond, v) \rightarrow (\text{VOTER}(n, \diamond, C)) \vee (\text{VOTER}(n, \diamond, H) \wedge \text{VOTE}(n, v)).$$

Our updated variant is nearly identically but adds the requirement that the voter has not complained.

Cast as intended (ID-based)-New: For every ballot in the tally, belonging to name n for choice v either the voter n is dishonest or the voter is honest, believes their ballot to have been correctly cast and intended to cast a ballot for v . (We denote by \diamond variables which may take any value.)³

$$\text{GOING_TO_TALLY}(n, \diamond, \diamond, v) \wedge \text{FINISHED}(n, \diamond) \rightarrow (\text{VOTER}(n, \diamond, C)) \vee (\text{VOTER}(n, \diamond, H) \wedge \text{VOTE}(n, v)).$$

Refinement 2 *We split the bidirectional channel between the voting device to the voting server into separate channels in each direction. This allows us to refine the adversary’s control over the channels. In particular, we refine the case where the voter’s password is leaked so that the adversary can impersonate the voter but not the voting server.*

While the first two refinements prevent certain attacks, they do not actually improve the security of the scheme in any threat model since the adversary can perform slightly more complicated attacks to break verifiability. Nevertheless, removing these simple attacks is necessary so that ProVerif identifies the more complicated attacks.

4 Changes to the scheme

In addition to our changes to the model we also made two changes to the scheme.

Refinement 3 *The voting device sends a random nonce to the server along with the vote. The voting server sends this nonce back with the acknowledgement. An honest device only tells the voter that her ballot was cast if the nonce received with the acknowledgement matches the nonce sent.*

³ Intuition would be better served by using VERIFIED in place of FINISHED in the correspondence assertion below, however for technical reasons this is not possible. Since in our situation FINISHED implies VERIFIED for honest voters, there is no technical issue.

We have deliberately kept this refinement simple to increase deniability; interestingly neither the paper introducing BeleniosRF [7] or BeleniosVS [6] describe in detail how the voter should mislead the coercer.⁴ The formal security definitions for both do not capture what the adversary learns by demanding the acknowledgment from the voters. Intuitively, both schemes have good coercion resistance but various nuances are not captured by the formal definition. Intuition also suggest that our refinement does not worsen the situation since the nonce is chosen uniformly, randomly and independently (of the ballot).

The nonce returned to the device is the one submitted with the accepted ballot, since assuming the voting server and device are honest the adversary cannot fake the acknowledgment. However, one might wonder if the adversary can learn the nonce and then submit a different ballot with this nonce and hence trick the voter. We have proved in ProVerif that the adversary can (when both the voting server and voting device are honest) only learn the nonce after the ballot is received by the voting server (at which point no further ballot will be accepted). The ultimate validation of *the refinement* is that it *ensures verifiability* in scenarios which are otherwise vulnerable to attack.

Refinement 4 *The voting server keeps a list of the names of those who have voted. It will not accept more than one vote from each name.*

The original scheme ensured that only one ballot was accepted per credential but allowed multiple votes to come from the same name (though this should never occur if the registrar and voting server are honest). We extend the scheme to also prevent multiple ballots being accepted for the same name. This refinement (and the accompanying axiom we mentioned earlier) are necessary for ProVerif to terminate in two cases where the registrar is dishonest. So far as we can tell this is an artefact of the proof; that is we know of no attack, in an otherwise secure threat model, which is prevented by this refinement. Nevertheless, the refinement seems eminently sensible and we recommend its use in practice.

5 Conclusion Key Takeaways

Based on our analysis and investigation we draw the following two conclusions; the first is important for automatic machine-checked analysis and the second for all e-voting schemes.

Model validation Our work highlights the importance of validating the attacks found by automatic theorem provers; it is always important to check that they work on the deployed scheme.

Defence in depth Our work highlights the importance of basic consistency checks in the honest protocol.

⁴ There is some informal discussion but it is unclear what threat model is trying to be captured.

Distinction between verifiability and eligibility verification The definition of “verifiability” in [6] being closer to integrity had the virtue of capturing ballot stuffing attacks which our new definition does not; our new definition is satisfied even if it is possible to cast ballots on behalf of honest voters, who don’t vote, without knowing their credential. Our new definition would need to be used alongside a definition of eligibility verification which prevents this attack. Future definitions may wish to consider introducing a special category for honest voters, with leaked credentials, who do not vote.

We show in table 1 a summary of the ProVerif analysis. We have highlighted our improvements in blue. We list the assumptions on participants and information with blank denoting honest, \blacklozenge denoting dishonest (or leaked), and - denoting none. R denotes the registrar, VS the voting server, VD the voting device, AD the auditing Device, CS the code sheet, P the password. We also include an example attack where relevant.

Dishonest parties and leaked data						Properties	Attack
R	VS	VD	AD	CS	P	Verifiability	
						✓	NA
\blacklozenge					\blacklozenge	✓	NA
			\blacklozenge	\blacklozenge		✓	NA
			-	\blacklozenge		✓	NA
\blacklozenge	\blacklozenge			\blacklozenge		✓	NA
\blacklozenge	\blacklozenge		-	\blacklozenge		✓	NA
				-	\blacklozenge	✓	NA
\blacklozenge	\blacklozenge					×	Casts a different vote
\blacklozenge		\blacklozenge				×	Casts a different vote
\blacklozenge				\blacklozenge		×	Bad voting sheet
\blacklozenge				-		×	Bad voting sheet
\blacklozenge					\blacklozenge	✓	NA
\blacklozenge					\blacklozenge	✓	NA
\blacklozenge		\blacklozenge				×	Casts a different vote
\blacklozenge				\blacklozenge		×	Casts a different vote
		\blacklozenge	\blacklozenge			×	Casts a different vote
		\blacklozenge		\blacklozenge		×	Casts a different vote
		\blacklozenge			\blacklozenge	✓	NA
			\blacklozenge		\blacklozenge	✓	NA
			\blacklozenge	\blacklozenge	\blacklozenge	✓	NA
				\blacklozenge	\blacklozenge	✓	NA
			-	\blacklozenge	\blacklozenge	✓	NA

Table 1. Security model

Future work In most of the remaining threat models the adversary can vote on the voter’s behalf and this goes undetected because either the voting device or

voting server is dishonest, there are also eight threat models where the registrar can break verifiability for voters who do not audit their code sheet. These lines of attack seem impossible to prevent without introducing new players (or new channels between existing players). Future work, particularly if it has a concrete deployment situation in mind, should revisit if it is feasible to introduce the new players or channels. For example, the voting server could send an SMS to the voter acknowledging that her ballot was received.

As we have noted, the formal definition of “verifiability” in [6] is intuitively much closer to integrity (than verifiability) since it does not model verification checks that fail. We leave as future work updating the formal definition of end-to-end “verifiability” in the symbolic model to catch verification failures; we, also, leave as future work formally showing that our refined sufficient conditions satisfy the (as yet nonexistent new) formal definition.

Source code The ProVerif source files are available at <https://github.com/gerlion/Improved-Verifiability-for-BeleniosVS>.

Acknowledgments

Thomas Haines was supported by Research Council of Norway and the Luxembourg National Research Fund (FNR), under the joint INTER project SURCVS (INTER/RCN/17/11747298/SURCVS/Ryan).

References

1. B. Adida. Helios: Web-based open-audit voting. In *In Proceedings of the 17th USENIX Security Symposium (Security '08)*, 2008.
2. G. Barthe, F. Dupressoir, B. Grégoire, C. Kunz, B. Schmidt, and P. Strub. Easy-crypt: A tutorial. In *FOSAD*, volume 8604 of *Lecture Notes in Computer Science*, pages 146–166. Springer, 2013.
3. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS*, pages 62–73. ACM, 1993.
4. Y. Bertot, P. Castéran, G. Huet, and C. Paulin-Mohring. *Interactive theorem proving and program development : Coq'Art : the calculus of inductive constructions*. Texts in theoretical computer science. Springer, 2004.
5. B. Blanchet. Modeling and verifying security protocols with the applied pi calculus and proverif. *Found. Trends Priv. Secur.*, 1(1-2):1–135, 2016.
6. V. Cortier, A. Filipiak, and J. Lallemand. Belenios-VS: Secrecy and verifiability against a corrupted voting device. In *2019 IEEE 32nd Computer Security Foundations Symposium (CSF)*, pages 367–36714. IEEE, 2019.
7. V. Cortier, G. Fuchsbauer, and D. Galindo. Beleniosrf: A strongly receipt-free electronic voting scheme. *IACR Cryptology ePrint Archive*, 2015:629, 2015.
8. V. Cortier, D. Galindo, S. Glondu, and M. Izabachène. Election verifiability for helios under weaker trust assumptions. In *ESORICS (2)*, volume 8713 of *Lecture Notes in Computer Science*, pages 327–344. Springer, 2014.
9. V. Cortier, D. Galindo, and M. Turuani. A formal analysis of the neuchatel e-voting protocol. In *EuroS&P*, pages 430–442. IEEE, 2018.

10. V. Cortier, P. Gaudry, and S. Glondu. Belenios: A simple private and verifiable electronic voting system. In *Foundations of Security, Protocols, and Equational Reasoning*, volume 11565 of *Lecture Notes in Computer Science*, pages 214–238. Springer, 2019.
11. D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Trans. Inf. Theory*, 29(2):198–207, 1983.

Provably Improving Election Verifiability in Belenios

Sevdenur Baloglu¹[0000-0002-3971-9683], Sergiu Bursuc¹[0000-0002-0409-5735],
Sjouke Mauw²[0000-0002-2818-4433], and Jun Pang²[0000-0002-4521-4112]

¹ SnT, University of Luxembourg, Esch-sur-Alzette, Luxembourg

² DCS, University of Luxembourg, Esch-sur-Alzette, Luxembourg
{sevdenur.baloglu,sergiu.bursuc,sjouke.mauw,jun.pang}@uni.lu

Abstract. Belenios is an online voting system that provides a strong notion of election verifiability, where no single party has to be trusted, and security holds as soon as either the voting registrar or the voting server is honest. It was formally proved to be secure, making the assumption that no further ballots are cast on the bulletin board after voters verified their ballots. In practice, however, revoting is allowed and voters can verify their ballots anytime. This gap between formal proofs and use in practice leaves open space for attacks, as has been shown recently. In this paper we make two simple additions to Belenios and we formally prove that the new version satisfies the expected verifiability properties. Our proofs are automatically performed with the Tamarin prover, under the assumption that voters are allowed to vote at most four times.

Keywords: Electronic voting · Formal verification · Verifiability.

1 Introduction

Election verifiability aims to ensure that the outcome of an election, relying on a given electronic voting protocol, correctly reflects the votes of eligible voters. One of its important features is that it should be software independent and end-to-end: even if an adversary corrupts (the software on) voting platforms, election authorities, or voting servers, the public information published on the bulletin board should be sufficient to verify that the election outcome correctly reflects voter choices. This verification is performed by honest parties, which are typically a subset of voters and election auditors. Especially for voters, the verification procedure should also be easy to use, in order to achieve widespread adoption and security guarantees.

Helios is an internet voting system that targets this notion of end-to-end verifiability [1,6,7]. However, an important assumption is that the voting server is honest. Otherwise it could stuff ballots, allowing the adversary to add illegitimate votes, most easily for voters that have not voted. In general, for usability, *revoting is allowed* and *voters can verify their ballots anytime* after voting. In that case ballot stuffing is possible even for voters that have verified their ballots successfully. For example, the server can let some time elapse after a ballot was

S. Baloglu et al.

cast, and cast a new ballot in the name of the same voter. This looks like revoting to observers and will not be noticed by voters verifying their ballots right after voting. The so-called clash attacks allow ballot stuffing in a more surreptitious way [22,23,26]: the adversary gives the same credential to two voters, one single vote is cast for them, and the adversary can cast an additional ballot with no change in the total number of ballots. If revoting is disallowed or ballot verification is after the voting phase, this requires voting platforms to be corrupted, since the adversary needs to supply the same ballot for two voters. Otherwise, it was shown in [9] that corrupting the voting platform is not needed: one voter can verify one ballot and another voter can subsequently verify another ballot for the same credential.

Belenios extends Helios in order to get stronger election verifiability [2,16]. There is no single party that has to be trusted: verifiability holds as soon as either the voting server *or* the voting registrar is not corrupted. The registrar generates public credentials, publishes them on the bulletin board, and distributes the respective private credentials to each voter. The public credential is the verification key of a fresh signing key pair, while the private credential is the corresponding signing key. Ballots are signed and election authorities can verify on the bulletin board that all ballots have been cast by the expected legitimate party. A second advantage of Belenios is that it was proved to satisfy a formal notion of election verifiability, both in the symbolic model [15] (for a particular variant) and in the computational model [14]. This adds confidence that verifiability is satisfied by the protocol specification. Nonetheless, several problems of Belenios and of verifiability definitions in [14,15] were shown in [9], leading to weaker guarantees than expected. In the typical scenario when revoting is allowed and voters can verify their ballots anytime, attacks on verifiability are still possible, most damaging in the case when the registrar is corrupted. Even in the ideal case when both the server and the registrar are honest, ballot reordering attacks are possible, breaking individual verifiability. These attacks are outside the scope of proofs in [14,15], since they do not consider the typical scenario of revoting.

Usability, Everlasting Privacy and Verifiability. There are two main features that, put together, allow these attacks on verifiability in Belenios. The first feature is that, in practice [2], revoting is allowed and voters can verify their ballots anytime. This is important for usability and, eventually, also for coercion-resistance [11,21]. The second feature is that the voting server does not know the link between the public credentials and the corresponding voter identities. Only at ballot casting time does the voter reveal this link, and the server ensures its consistency, e.g. that the same public credential does not correspond to two different voters. Revealing minimal information about the association between voters and their public credentials is important in order to ensure everlasting privacy: even if an adversary may break the underlying encryption scheme and penetrate the private logs of the server, the connection between voters and the corresponding votes should remain private. A similar pattern underlies all attacks in [9]: a corrupted voter can be used by the adversary to cast a ballot for a public credential corresponding to an honest voter. Even if honest voters

Provably Improving Election Verifiability in Belenios

successfully verified their ballots, revoting allows the adversary to undetectably replace them with its own ballots (when the registrar is corrupted), or with earlier ballots submitted by the same voters (when the registrar is honest).

Our Contributions. We propose two simple additions to Belenios and we prove that election verifiability of the resulting system, that we call Belenios+, is strictly stronger in all three scenarios that are subject to attacks in [9]. Each scenario is defined by the corruption abilities of the adversary: \mathcal{A}_1 - both the server and the registrar are honest; \mathcal{A}_2 - the server is corrupt and the registrar honest; \mathcal{A}_3 - the server is honest and the registrar corrupt. In all cases, we assume the adversary may corrupt the secret key of the election, any number of voters and the communication network. For voters, the proposed additions do not require any change in the voting and verification procedures, maintaining the same usability as Belenios. We do not communicate any new information to the voting server regarding the link between voter identities and public credentials. We simply enforce the veracity of the information the voter already communicates. This means that our additions should not affect everlasting privacy in Belenios (everlasting privacy has not been formally proved for Belenios, but it is thought to hold when revoting is not allowed [16]). Our security proofs are in the symbolic model, automatically performed with the Tamarin prover [24], although we need to make some further abstractions, as explained below. We use the verifiability definition of [9], which is more general than [14,15], accounting for revoting and different corruption scenarios.

Belenios relies on a zero-knowledge proof in order to verifiably attach a label to each ballot cast. The label is the public credential of the voter who constructs the ballot and the ballot cannot be detached from the intended label. The goal of this construction is to ensure that each ballot is consistently cast for the intended public credential. Our techniques enrich the structure of the label in order to ensure stronger consistency properties. The first problem that we tackle is a ballot reordering attack, which is possible in all three corruption scenarios, i.e. even for the weakest adversary \mathcal{A}_1 . Omitting some details (presented in Section 2.2), the attack is as follows: an honest voter with public credential cr , may submit two successive ballots b_1 and b_2 ; then, relying on a corrupt voter, the adversary can cast b_2 before b_1 , for the same public credential cr . The honest voter may then verify b_2 and expect it to be tallied, whereas b_1 is tallied instead. The solution we propose for this problem is to augment the label in the zero-knowledge proof such that each new ballot can also be verifiably linked to the ballot that was cast just before for a given public credential. This proof is publicly verified on the bulletin board, thus it also helps in the scenario \mathcal{A}_2 .

The second problem in Belenios relates to the scenario \mathcal{A}_3 and is at the root of several attacks in [9]: because the voting server does not know in advance the connection between voter identities and public credentials, an adversary corrupting the registrar and a voter may submit any ballot for any public credential cr , and claim it corresponds to that corrupt voter. In particular, this may be a ballot b constructed by an honest voter that received the public credential cr at registration. This leads to the fact that the honest voter may successfully

S. Baloglu et al.

verify \mathbf{b} on the bulletin board, while afterwards the adversary is able to cast its own ballot \mathbf{b}_A for the credential \mathbf{cr} . The solution we propose for this problem is to further augment the label in the zero-knowledge proof such that the voting server can ensure that the cast ballot is intended for the corresponding voter. However, we need to make sure that only the server can verify the link between a ballot and the voter identity. That is why the label does not directly contain the identity \mathbf{id} of the voter, but a commitment to \mathbf{id} , for which the server learns the randomness from the voting platform. The randomness can be discarded by the server after reconstructing the commitment and verifying the proof. To hide the identity from an all-powerful adversary against the bulletin board, we can use standard commitment schemes that are perfectly hiding, for example the Pedersen commitment [25].

Abstraction. In practice, the two additions we make do not significantly affect the complexity of running Belenios. However, the fact that we need to recursively link every new ballot with a previously cast ballot significantly affects the running time of Tamarin. To overcome this difficulty, we assume that each voter casts at most four ballots, in effect allowing revoting only thrice (all attacks of [9] occur in scenarios with at most two ballots per voter). We leave as open the problem of formally proving (or disproving) the validity of this assumption. We note that formal results that bound the number of agents or voters for verification have a similar flavour [8,12,13].

Paper Structure. Section 2 contains preliminaries about election verifiability and attacks on Belenios. In Section 3 we describe our improvements and in Section 4 we describe the protocol specification and automated verification with the Tamarin prover.

2 Preliminaries

We describe Belenios in more detail in Section 2.1. The formal notion of election verifiability and the attacks on Belenios are described in Section 2.2.

2.1 Introduction to Belenios

Apart from voters (\mathbf{V}), the parties in the Belenios protocol [16,2] are:

- *Administrator* (\mathbf{A}): determines the list of eligible candidates and the list of eligible voters.
- *Bulletin Board* (\mathbf{BB}): public ledger containing election information: the public key, the list of candidates, the list of public credentials for eligible voters, the list of cast ballots, the final outcome and proofs of correctness. We denote specific portions of \mathbf{BB} with suffixes. In particular, \mathbf{BBkey} contains the public key of the election, \mathbf{BBcast} contains the list of ballots cast for each public credential, and $\mathbf{BBtally}$ contains the list of ballots chosen for tally. \mathbf{BB} can only be changed by writing new information on it; previously written information cannot be changed.

Provably Improving Election Verifiability in Belenios

- *Trustees* (T): generate the secret key of the election, publish the corresponding public key on BB, compute the final outcome.
- *Registrar* (VR): for each eligible voter, it creates a fresh signing key pair $(vk, skey)$; vk is the public credential, which is also denoted by cr in the following; it publishes the list of all public credentials on BB.
- *Voting Server* (VS): receives ballots cast by authenticated voters and publishes them on BB; voter authentication is done via passwords.
- *Voting Platform* (VP): constructs ballots for voter choices; authenticates voters with respect to VS and transmits ballots to VS; each ballot contains a ciphertext encrypting the vote, a signature of the ciphertext with respect to $skey$ of the corresponding voter, and zero-knowledge proofs.
- *Election Auditors* (EA): perform audit and verification of proofs on BB. The validity of the ballot is verified by VS at ballot-casting time, and can also be verified by EA at any time afterwards on BBcast.

Setup Phase. A determines the list of eligible voters id_1, \dots, id_n , and sends the list to VR and VS. VR generates the public and private credentials for each voter, while VS generates login passwords. Each voter id receives the tuple $\langle cr, skey, pwd \rangle$ during setup phase and BB is updated by the following:

$$\text{BBkey: } pk; \quad \text{BBcand: } v_1, \dots, v_k; \quad \text{BBreg: } cr_1, \dots, cr_n.$$

Voting Phase. In this phase, voters interact with their voting platform VP to construct a ballot b , which is sent together with their public credential cr to VS. Upon authentication of the voter and validity checks with respect to cr , the ballot is published on BBcast.

$$\begin{aligned} \text{VP: } & c = \text{enc}(v, pk, r); \quad s = \text{sign}(c, skey); \quad pr_R = \text{proof}_R(c, r, \langle v_1, \dots, v_k \rangle); \\ & pr_L = \text{proof}_L(c, r, cr); \quad b = \langle c, s, pr_R, pr_L \rangle; \end{aligned}$$

$$\text{VS: } \text{authenticates } id \text{ with } pwd; \text{ receives } b \text{ and the public credential } cr; \\ \text{verifies } s, pr_R \text{ and } pr_L; \text{ and stores } (id, cr) \text{ in } \text{Log};$$

$$\text{BBcast: } (cr, b).$$

The signature ensures the voter holds the private part of the public credential cr . The zero-knowledge proof pr_R ensures that the ciphertext contains a vote in a valid range $\langle v_1, \dots, v_k \rangle$. The proof pr_L ensures that the ballot (and the ciphertext) is verifiably linked to the label cr , and cannot be cast for any other credential cr' . In the cryptographic construction, the underlying zero-knowledge proof system takes the arguments of proof_R and proof_L and returns pr_R and pr_L [14,16]. Moreover, the following consistency property is ensured by VS for the Log storing the association between voter identities and public credentials:

$$\begin{aligned} (id, cr) \in \text{Log} \wedge (id, cr') \in \text{Log} &\Rightarrow cr = cr' \quad \text{and} \\ (id, cr) \in \text{Log} \wedge (id', cr) \in \text{Log} &\Rightarrow id = id'. \end{aligned}$$

This prevents a corrupt voter to use a public credential already used by an honest voter, and also to cast ballots for more than one public credential. In addition

S. Baloglu et al.

to ensuring basic integrity properties, consistency of the log also prevents ballot copy attacks like in [17]. The individual verification procedure enables voters to check their ballots on BB anytime during the election. Specifically, they should check that the expected ballot \mathbf{b} is published next to their public credential \mathbf{cr} on BBcast.

Tally Phase. The ballots which will be tallied are selected and marked as input for the tally procedure. Selection typically chooses the last ballot cast by each \mathbf{cr}_i and we have $\text{BBtally}: (\mathbf{cr}_1, \mathbf{b}_1), \dots, (\mathbf{cr}_n, \mathbf{b}_n)$. $\mathbf{b}_i = \perp$ if no ballot was cast for \mathbf{cr}_i . Based on the homomorphic properties of ElGamal encryption [18,20], ciphertexts corresponding to non-empty ballots on BBtally are combined into a ciphertext \mathbf{c} encoding the total number of votes for each candidate. Then, \mathbf{c} is decrypted by trustees to obtain the result of the election.

2.2 Election Verifiability and Attacks on Belenios

We consider the symbolic definition of election verifiability from [9], which is an extension of the symbolic definition introduced in [15]. Election verifiability is modelled as a conjunction of properties $\Phi_{\text{iv}}^h \wedge \Phi_{\text{eli}} \wedge \Phi_{\text{cl}} \wedge \Phi_{\text{res}}^\circ$, where:

Individual verifiability: Φ_{iv}^h ensures that if an honest voter successfully verified the last ballot they cast, then the corresponding vote should be part of the final tally.

Eligibility: Φ_{eli} ensures that if a voter successfully verified a ballot, then the corresponding public credential should be recorded at registration on BB. Moreover, any tallied ballot should correspond to a public credential recorded at registration.

No clash: Φ_{cl} ensures that no two voters can successfully verify their ballot for the same public credential.

Result integrity: Φ_{res}° ensures that the adversary can cast a ballot for a given public credential only if the corresponding voter is corrupted or has not performed the individual verification procedure for any of the ballots cast. A stronger notion of result integrity, denoted by $\Phi_{\text{res}}^\bullet$, prohibits the adversary to cast a ballot even if the voter has not verified any of the ballots cast.

A violation of Φ_{res}° is called ballot stuffing; a violation of Φ_{cl} is a clash attack. Belenios is expected to satisfy election verifiability in the following adversarial scenarios: \mathcal{A}_1 - both the server and the registrar are honest; \mathcal{A}_2 - the server is corrupt and the registrar honest; \mathcal{A}_3 - the server is honest and the registrar corrupt. Security should be ensured by private signing keys - when the registrar is honest, and by private passwords and server logs - when the server is honest. However, [9] shows several attacks resulting from the fact that the server does not know the association between a public credential and the identity of the corresponding voter. A corrupt voter can then cast a ballot for any public credential, as soon as the adversary manages to obtain ballots signed with the corresponding private credential.

Ballot Reordering Attack by \mathcal{A}_1 , \mathcal{A}_2 or \mathcal{A}_3 . Assume an honest voter id with public credential cr casts ballots b_1 and b_2 , in this order, and only verifies b_2 . Then b_2 should be counted for the respective public credential. However, the adversary can cause b_1 to be counted instead. The attack scenario is as follows:

$V(id, cr)$: casts b_1 and b_2 , which are blocked by \mathcal{A} ;
 \mathcal{A} : casts b_2 for cr (relying on a corrupted voter or voting server);
 $BBcast$: (cr, b_2) is verified by the voter $V(id, cr)$;
 \mathcal{A} : casts b_1 for cr ;
 $BBtally$: (cr, b_1) .

In a normal execution, the reception of b_1 or b_2 from id would link cr to id , thus the adversary cannot cast b_1 after b_2 when the server is honest - unless it corrupts the password of id . The crucial point of the attack by \mathcal{A}_1 is that b_2 is cast for the same public credential cr by a distinct corrupted voter.

Ballot Stuffing Attack by \mathcal{A}_3 . When an honest voter id_1 with cr_1 casts a ballot b , the adversary can block and cast it in the name of a corrupt voter id_2 , for the same public credential cr_1 . The voter id_1 successfully verifies b . Subsequently, relying on a corrupt registrar, the adversary can cast another ballot $b_{\mathcal{A}}$ for cr_1 . This violates result integrity Φ_{res}^o and individual verifiability Φ_{iv} , since an adversarial ballot $b_{\mathcal{A}}$ is cast for cr_1 , even though the corresponding voter is honest and has successfully verified the ballot b .

\mathcal{A} : corrupts VR and $V(id_2)$ to obtain $(cr_1, skey_1, pwd_2)$;
 $V(id_1)$: casts b , which is blocked by \mathcal{A} ;
 \mathcal{A} : casts b with (cr_1, pwd_2) , and VS stores (id_2, cr_1) in Log;
 $BBcast$: (cr_1, b) is verified by $V(id_1)$;
 \mathcal{A} : casts $b_{\mathcal{A}}$ with (cr_1, pwd_2) , which is accepted and published;
 $BBtally$: $(cr_1, b_{\mathcal{A}})$.

If the voter id_2 verified the cast ballot b , this also counts as a clash attack in the definition from [9], as it requires resistance to clash attacks even for corrupted voters. A variation of this attack can also lead to a weaker form of ballot stuffing: the adversary can submit $b_{\mathcal{A}}$ before id_1 has a chance to cast a ballot. In that case, the voting server will not accept any further ballot from id_1 , since this would break the consistency of the log for cr_1 . Formally, this is a violation of Φ_{res}^\bullet . Our techniques in the following protect against (strong) ballot stuffing, ballot reordering, and the clash attack. They do not protect against the weaker form of ballot stuffing, i.e. the violation of Φ_{res}^\bullet .

3 Towards Improved Election Verifiability

In Belenios, the aim of the zero-knowledge proof $pr_L = \text{proof}_L(c, r, cr)$ in a ballot $b = \langle c, s, pr_R, pr_L \rangle$ is to verifiably link the ciphertext $c = \text{enc}(v, pk, r)$, and therefore the ballot b , to the public credential cr for which b is cast. We denote the corresponding verification procedure by $\text{ver}_L(pr_L, c, cr)$. A valid proof can only

S. Baloglu et al.

be constructed by the party who constructs the ciphertext, by proving knowledge of the corresponding randomness r with the label cr . This is called labeled encryption in [14]. The idea is that the ciphertext cannot be detached from the label: the adversary cannot copy c , or create a ciphertext related to the encoded vote, and cast it for a different credential cr' . This is required in order to protect from attacks against privacy like in [17]. Concretely, the labeled encryption in Belenios is based on ElGamal encryption with a Chaum-Pedersen proof of knowledge, where the label cr is part of the input to a hash function (SHA256) that computes the challenge for a non-interactive zero-knowledge proof.

We enrich the structure of the label in order to also protect against the attacks presented in Section 2.2. The elements of the new label structure can be given as inputs to the hash function along with cr in the Chaum-Pedersen proof, thus we can rely on the same labeled encryption construction as Belenios. Moreover, we prove in Section 4 that no further attacks are possible on election verifiability in the resulting system. We present the new structure of the label stepwise: first a label structure that protects against ballot reordering attacks by $\mathcal{A}_1, \mathcal{A}_2$ or \mathcal{A}_3 ; then a label structure that protects against other attacks by \mathcal{A}_3 (in particular ballot stuffing); finally, combining the two labels protects against all attacks by $\mathcal{A}_1, \mathcal{A}_2$ or \mathcal{A}_3 .

3.1 Protection Against Ballot Reordering

We assume initially there are empty ballots next to eligible public credentials on BB. Moreover, a specific portion of BB is reserved for displaying the last ballot cast for each credential:

$$\begin{array}{l} \text{(Before voting)} \quad \text{BBlast} : (cr_1, \perp), \dots, (cr_n, \perp) \\ \text{-----} \\ \text{(During voting)} \quad \text{BBlast} : (cr_1, b_1), \dots, (cr_n, b_n) \end{array}$$

When the voting platform VP constructs a new ballot for a voter with public credential cr , it fetches from BBlast the last ballot b' next to cr . Then, in the construction of the proof pr_L , instead of cr , VP uses the label $h(cr, b')$, where h is a collision-resistant hash function mapping the pair (cr, b') into the appropriate domain for labels:

$$\ell = h(cr, b'); \quad pr_L = \text{proof}_L(c, r, \ell); \quad b = \langle c, s, pr_R, pr_L, \ell \rangle.$$

BBcast records all ballots cast for cr , and their order cannot be changed on BB. Election auditors can look at any two consecutive ballots b' and b cast for a credential cr and verify that

$$\text{ver}_L(pr_L, c, h(cr, b')) = \text{ok},$$

thereby ensuring that the party constructing b indeed expects it to follow b' . In particular, if an honest voter casts b_2 after b_1 , the adversary cannot cast b_2 first, since it would have to generate a proof linking b_2 to an earlier ballot b_0 , which is impossible since the adversary does not know the randomness in the ciphertext corresponding to b_2 . This label structure ensures election verifiability in corruption scenarios when the registrar is honest, i.e. \mathcal{A}_1 and \mathcal{A}_2 .

3.2 Protection Against a Corrupted Registrar

The main cause of the ballot stuffing and clash attacks, in the scenario with a corrupted registrar, is that the adversary can block a ballot \mathbf{b} of an honest voter and cast it under the identity of a corrupt voter, while maintaining the same public credential associated to \mathbf{b} . Subsequently, after the honest voter verified \mathbf{b} , the adversary can override it with an own ballot $\mathbf{b}_{\mathcal{A}}$. In order to prevent this, we enrich the label attached to \mathbf{b} so that it includes a commitment to the identity of the voter. More precisely, during ballot casting for a voter id , VP generates a fresh randomness \mathbf{t} , constructs the label $\langle \text{cr}, \text{com}(\text{id}, \mathbf{t}) \rangle$ and sends \mathbf{t} together with the ballot to the voting server VS . Since the label cannot be reconstructed publicly by election auditors, we explicitly include it in the ballot. We have:

$$\begin{array}{l} \text{VP} : \ell = \langle \text{cr}, \text{com}(\text{id}, \mathbf{t}) \rangle; \quad \text{pr}_{\text{L}} = \text{proof}_{\text{L}}(\mathbf{c}, r, \ell); \quad \mathbf{b} = \langle \mathbf{c}, \text{s}, \text{pr}_{\text{R}}, \text{pr}_{\text{L}}, \ell \rangle, \\ \text{-----} \\ \text{VS} : \text{receives } (\text{cr}, \mathbf{b}, \mathbf{t}) \text{ from VP for a given id; } \ell' = \langle \text{cr}, \text{com}(\text{id}, \mathbf{t}) \rangle, \\ \text{casts } \mathbf{b} \text{ if and only if } \ell' = \ell \text{ and } \text{ver}_{\text{L}}(\text{pr}_{\text{L}}, \mathbf{c}, \ell) = \text{ok}. \end{array}$$

In the attack scenario described above, the adversary cannot construct a proof pr'_{L} so that \mathbf{b} is cast by VS under the identity of a corrupt voter. Indeed, the ciphertext in \mathbf{b} cannot be detached from the identity of the honest voter. More generally, we prove that this structure of the label is sufficient to ensure election verifiability in the corruption scenarios when the server is honest, i.e. \mathcal{A}_1 and \mathcal{A}_3 . Election auditors can still check the proof pr_{L} on BB , but they are only able to ensure the ballot is cast for the expected public credential cr and will not have knowledge of the underlying id . Note that we cannot use the id directly in the label, as this would reveal the link between id and cr . Moreover, the commitment scheme should be perfectly hiding, in order to resist an all-powerful, e.g. quantum, adversary.

3.3 Putting the Labels Together

We combine the labels from Section 3.1 and Section 3.2 as follows:

$$\ell_1 = \text{h}(\text{cr}, \mathbf{b}'); \quad \ell_2 = \text{com}(\text{id}, \mathbf{t}); \quad \ell = \langle \ell_1, \ell_2 \rangle.$$

We call $\text{Belenios}_{\text{tr}}$ (from tracking) the variant of Belenios where we augment the label as described in Section 3.1, $\text{Belenios}_{\text{id}}$ the variant where the label is as in Section 3.2 and $\text{Belenios}+$ the variant where the label ℓ is as described in this section. For a protocol P , a corruption scenario \mathcal{A} and a property Φ , we denote by $(P, \mathcal{A}) \models \Phi$ the fact that P satisfies Φ in the corruption scenario \mathcal{A} . Let $\Phi_{\text{E2E}}^{\circ}$ be the election verifiability property $\Phi_{\text{iv}}^{\text{h}} \wedge \Phi_{\text{eli}} \wedge \Phi_{\text{cl}} \wedge \Phi_{\text{res}}^{\circ}$ as described in Section 2.2 and in [9]. In the next section, we describe the specification and automated verification with Tamarin. They allow us to derive the following results:

$$\begin{array}{l} (\text{Belenios}_{\text{tr}}, \mathcal{A}) \models \Phi_{\text{E2E}}^{\circ} \quad \text{for } \mathcal{A} \in \{\mathcal{A}_1, \mathcal{A}_2\}, \\ (\text{Belenios}_{\text{id}}, \mathcal{A}) \models \Phi_{\text{E2E}}^{\circ} \quad \text{for } \mathcal{A} \in \{\mathcal{A}_1, \mathcal{A}_3\}, \\ (\text{Belenios}+, \mathcal{A}) \models \Phi_{\text{E2E}}^{\circ} \quad \text{for } \mathcal{A} \in \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}, \\ \text{while we have } (\text{Belenios}, \mathcal{A}) \not\models \Phi_{\text{E2E}}^{\circ} \quad \text{for } \mathcal{A} \in \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}. \end{array}$$

S. Baloglu et al.

The property Φ_{E2E}° corresponds to the standard verifiability notion used in [14,15]. In particular, this notion ensures that, if an honest voter successfully verified a ballot b for a public credential cr , then b is counted in the final tally as the contribution of cr . A stronger notion of verifiability, denoted by Φ_{E2E}^{\bullet} , was also proposed in [9]: if a ballot is counted for a public credential corresponding to an honest voter, then it must necessarily have been cast by that voter - independently of the individual verification procedure. In the scenario \mathcal{A}_3 , an adversary corrupting the registrar and a voter can cast a ballot $b_{\mathcal{A}}$ for any public credential, violating the strong verifiability notion Φ_{E2E}^{\bullet} , even in Belenios+. The label $\langle h(cr, b'), com(id, t) \rangle$ does not help here, since the adversary can freely combine the identity of a corrupted voter with any credential, sign the ballot and construct valid zero-knowledge proofs. If the honest voter already submitted and successfully verified a ballot b , then the adversary cannot make VS accept $b_{\mathcal{A}}$ for the same public credential under the identity of a corrupt voter. This is due to the fact that the association between the honest voter and the public credential is recorded by the server in the log upon accepting b . That is why Φ_{E2E}° holds for Belenios+.

4 Specification and Verification

4.1 Specifying Protocols in Tamarin

We perform our analysis of Belenios+ using the Tamarin prover, which is based on a multiset rewriting framework. We only illustrate the most relevant features of Tamarin here. For a detailed understanding of Tamarin we refer the reader to [3,24,27]. In Tamarin, messages (or terms) are built from a set of function symbols and properties of cryptographic primitives are modelled by a set of equations. Protocol state information and adversarial knowledge are represented by facts, modelled relying on special fact symbols. Protocol actions are specified by multiset rewriting rules, denoted by $[L] \multimap [M] \multimap [N]$, in which a set of *premise facts* L allows to derive a set of *conclusion facts* N , while recording certain events in *action facts* M .

Example 1. In a voting protocol, the generation of a secret/public key pair can be modelled by the following multiset rewriting rule, that we denote by R_{key} :

$$[Fr(k)] \multimap [!BBkey(pk(k)), Phase('setup')] \multimap [!Sk(k), !BBkey(pk(k)), Out(pk(k))]$$

where $Fr(k)$ denotes the randomly generated fresh key k as a premise. The conclusion facts $!Sk(k)$ and $!BBkey(pk(k))$ record the secret and the public key of the election, respectively; the term $pk(k)$ represents the public key itself, while $!BBkey(pk(k))$ represents the fact that $pk(k)$ is a public key published on $BBkey$. If a fact is preceded by $!$, it means that it can be consumed (i.e. used as premise) any number of times by other protocol rules. Otherwise it can be consumed only once, and it is called a *linear* fact. The fact symbols In and Out are used for communication over the network, controlled by the attacker. The action fact

Provably Improving Election Verifiability in Belenios

$\text{BBkey}(\text{pk}(\mathbf{k}))$ records the event that the public key is published on the bulletin board. The action fact $\text{Phase}(\text{'setup'})$ records that the rule should be executed in the setup phase. The following rules set up candidates v_1 and v_2 and voter identities id :

$$\begin{aligned} R_{\text{cand}} &: [\text{In}(\langle v_1, v_2 \rangle)] \text{---} [\text{Phase}(\text{'setup'})] \text{---} [!\text{BBcand}(v_1), !\text{BBcand}(v_2)] \\ R_{\text{id}} &: [\text{In}(\text{id})] \text{---} [\text{Phase}(\text{'setup'})] \text{---} [!\text{Id}(\text{id})] \end{aligned}$$

To cast a ballot, the voter with identity id makes a choice between the candidates recorded on BBcand and encrypts the vote v using the public key from BBkey together with fresh randomness r . The output including the voter identity id can be sent to the server over the network. To model this action, we define the following rule, where the event $\text{Vote}(\text{id}, v)$ is recorded as an action fact:

$$\begin{aligned} R_{\text{vote}} &: [!\text{Id}(\text{id}), !\text{BBcand}(v), !\text{BBkey}(\text{pk}(\mathbf{k})), \text{Fr}(r)] \\ &\text{---} [\text{Vote}(\text{id}, v), \text{Phase}(\text{'voting'})] \text{---} [\text{Out}(\langle \text{id}, \text{enc}(v, \text{pk}(\mathbf{k}), r) \rangle)] \end{aligned}$$

Cryptographic operations are specified by equations. For example, decryption using the private key \mathbf{k} is specified by:

$$\text{dec}(\text{enc}(v, \text{pk}(\mathbf{k}), r), \mathbf{k}) = v$$

where the term $\text{enc}(v, \text{pk}(\mathbf{k}), r)$ represents the encryption of v with public key $\text{pk}(\mathbf{k})$ and randomness r . It can be decrypted only if the secret key \mathbf{k} is provided.

A *restriction* in Tamarin is a logical formula that constrains the application of protocol rules. For example, the restriction $\forall x, y, i, j. \text{BBkey}(x) @i \wedge \text{BBkey}(y) @j \Rightarrow x = y$ applied to the rule R_{key} in Example 1 means that it is not possible to have two different election keys. The symbol $@$ refers to the timepoints i and j in the execution trace when the rule R_{key} is applied. We can also express a timepoint ordering or equality. For example, the restriction $\forall i, j. \text{Phase}(\text{'setup'}) @i \wedge \text{Phase}(\text{'voting'}) @j \Rightarrow i < j$ means that all setup actions should occur before voting actions. A restriction can also encode the equality predicate, enforcing that u and v are equal in any occurrence of the action fact $\text{Eq}(u, v) : \forall u, v, i. \text{Eq}(u, v) @i \Rightarrow u = v$.

We note that formal verification with Tamarin does not guarantee full-proof security, as Tamarin itself may have bugs. Recently, there is research aiming to underpin fully automated provers like Tamarin with foundations from interactive theorem provers like Coq [4,10,19].

4.2 Specification and Verification of Belenios+

We define a set of equations used for specifying decryption (1), signature verification (2), verification of a range proof (3), and verification of a proof attaching a label to a ciphertext (4):

- (1) $\text{dec}(\text{enc}(x, \text{pk}(y), z), y) = x$,
- (2) $\text{ver}(\text{sign}(x, y), x, \text{pk}(y)) = \text{ok}$,
- (3) $(\forall i) \text{ver}_R(\text{proof}_R(\text{enc}(x_i, y, z), z, \langle x_1, \dots, x_k \rangle), \text{enc}(x_i, y, z), y, \langle x_1, \dots, x_k \rangle) = \text{ok}$,
- (4) $\text{ver}_L(\text{proof}_L(\text{enc}(x, y, z), z, \ell), \text{enc}(x, y, z), \ell) = \text{ok}$.

S. Baloglu et al.

To specify the set of equations (3) in Tamarin, the number of candidates k has to be fixed in advance. We use $k = 2$, but any constant would work. For modelling the actions of participants in the protocol, we define a set of rules and restrictions. For the complete specification, we refer to the Tamarin code online [5]. It is an extension of the code corresponding to Belenios in [9]. In the following, we discuss two of the most important rules in the specification: ballot casting as it happens on the voting platform VP and on the voting server VS. We highlight the difference between Belenios+ and Belenios in red. We use special linear facts in order to track the last ballot cast for each credential: $\text{VPlast}(cr, b_0)$ - to be used by the voting platform, and $\text{BBlast}(cr, b_0)$ - to be used by the voting server. The rule for ballot casting on the voting server makes sure these two facts are in sync. For voter credentials, we use special facts $\text{!Reg}(id, cr, skey)$ and $\text{!Pwd}(id, pwd)$ to store credentials received from the registrar and from the server, respectively. Ballot casting by VP is represented by the following rule:

$R_{\text{vote}}^{\text{VP}}$: **construct a ballot, authenticate and send it to VS**

```

let c = enc(v, pkey, r); s = sign(c, skey);  $\ell = \langle h(cr, b_0), com(id, t) \rangle$ ;
  prR = proofR(c, r, vlist); prL = proofL(c, r,  $\ell$ );
  b =  $\langle c, s, pr_R, pr_L, \ell \rangle$ ; a = h( $\langle id, pwd, cr, b, t \rangle$ ) in
[ !BBcand(v), !BBkey(pkey), Fr(r), Fr(t), !Vlist(vlist), !Reg(id, cr, skey),
  !Pwd(id, pwd), VPlast(cr, b0) ] - [ Vote(id, cr, v), VoteB(id, cr, b) ] ->
[ !Voted(id, cr, v, b), Out( $\langle id, cr, b, a, t \rangle$ ) ]

```

where we use the Tamarin construction `let...in` for assigning terms to variables. The rule abstracts password-based authentication with the help of a hash function, essentially ensuring that only a party knowing the password can cast a ballot for a given id. In reality, the randomness t used for the commitment should be sent on the same secure channel as the password. However, the secrecy of t is not important for verifiability properties, thus we can send it on the public channel. The rule $R_{\text{vote}}^{\text{VP}}$ consumes the linear fact $\text{VPlast}(cr, b_0)$, thus it can be executed only once for any ballot posted on BB. This mechanism is complemented by the ballot casting rule on the server side:

$R_{\text{cast}}^{\text{VS}}$: **authenticate voter, verify and publish ballot**

```

let  $\ell = \langle h(cr, b_0), com(id, t) \rangle$ ; b =  $\langle c, s, pr_R, pr_L, \ell \rangle$ ;
  a' = h( $\langle id, pwd, cr, b, t \rangle$ ) in
[ In( $\langle id, cr, b, a, t \rangle$ ), !BBkey(pkey), !Vlist(vlist), !BBreg(cr), !Pwd(id, pwd),
  BBlast(cr, b0) ] - [ a' = a, ver(s, c, cr) = ok, verR(prR, c, pkey, vlist) = ok,
  verL(prL, c,  $\ell$ ) = ok, Log(id, cr), !BBcast(cr, b) ] ->
[ !BBcast(cr, b), BBlast(cr, b), VPlast(cr, b) ]

```

where we receive a ballot from the voter and perform the corresponding validation steps: verifying the password, the signature and the zero-knowledge proofs. The fact containing the last ballot cast is consumed, and new facts are produced for the new ballot: one to be consumed by the voting platform, and one to be

Provably Improving Election Verifiability in Belenios

consumed by the server when the next ballot is cast. In order to obtain termination, we have a restriction limiting the number of applications of this rule to at most four for each voter. The following rule and restriction model the individual verification procedure, where the restriction ensures that the voter verifies the last ballot cast:

$$\begin{aligned} R_{\text{ver}}^V : [\text{Voted}(\text{id}, \text{cr}, \text{v}, \text{b}), \text{BBcast}(\text{cr}, \text{b})] \text{---} [\text{Verified}(\text{id}, \text{cr}, \text{v}), \text{VerB}(\text{id}, \text{cr}, \text{b})] \text{---} [\text{ }] \\ \text{BBcast}(\text{cr}, \text{b}) @i \wedge \text{BBcast}(\text{cr}, \text{b}') @j \wedge \text{VerB}(\text{id}, \text{cr}, \text{b}) @l \wedge i < l \wedge j < l \\ \Rightarrow j < i \vee \text{b} = \text{b}' \end{aligned}$$

Corruption Scenarios. We have three adversary models \mathcal{A}_1 , \mathcal{A}_2 and \mathcal{A}_3 , as described in Section 2.2. Trustees are corrupted by default: we have a rule that takes the secret key as input from the attacker. For other corruption abilities, we have the following rules:

$$\begin{aligned} C_{\text{corr}}^V : \text{corrupt voter to reveal credentials} \\ [!\text{Reg}(\text{id}, \text{cr}, \text{skey}), !\text{Pwd}(\text{id}, \text{pwd})] \text{---} [\text{Corr}(\text{id}, \text{cr})] \text{---} [\text{Out}(\langle \text{id}, \text{cr}, \text{skey}, \text{pwd} \rangle)] \end{aligned}$$

$$\begin{aligned} C_{\text{pwd}}^{\text{VS}} : \text{corrupt server to determine password} \\ [!\text{Id}(\text{id}), \text{In}(\text{pwd})] \text{---} [\text{ }] \text{---} [!\text{Pwd}(\text{id}, \text{pwd})] \end{aligned}$$

$$\begin{aligned} C_{\text{cast}}^{\text{VS}} : \text{corrupt server to stuff ballots} \\ [\text{In}(\langle \text{cr}, \text{b} \rangle), \text{BBlast}(\text{cr}, \text{b}_0)] \text{---} [!\text{BBcast}(\text{cr}, \text{b})] \text{---} \\ [!\text{BBcast}(\text{cr}, \text{b}), \text{BBlast}(\text{cr}, \text{b}), \text{VPast}(\text{cr}, \text{b})] \end{aligned}$$

$$\begin{aligned} C_{\text{reg}}^{\text{VR}} : \text{corrupt registration of public / secret credentials} \\ \text{let } \text{cr} = \text{pk}(\text{skey}) \text{ in} \\ [!\text{Id}(\text{id}), \text{In}(\langle \text{skey}, \text{cr}' \rangle)] \text{---} [!\text{BBreg}(\text{cr}')] \text{---} [!\text{Reg}(\text{id}, \text{cr}, \text{skey}), !\text{BBreg}(\text{cr}')] \end{aligned}$$

Moreover, when the server is corrupted, in the rule $R_{\text{vote}}^{\text{VS}}$ we only keep the verification actions that can be publicly performed by election auditors. Table 1 contains verification results for the corresponding specifications with Tamarin, obtained with the specifications posted online [5]. We can see that the positive results for Belenios+ are the union of the positive results for Belenios_{tr} and Belenios_{id}, in each of the corruption cases \mathcal{A}_1 , \mathcal{A}_2 and \mathcal{A}_3 . In Table 2, we give execution times for the verification of Belenios+ when we bound the number of ballots per voter accordingly. Tamarin does not terminate without such a bound (it takes more than one hour for five ballots per voter).

5 Conclusion and Future Work

We have introduced a simple extension of Belenios and we have proved with the Tamarin prover that the resulting system improves election verifiability in various corruption scenarios. These additions do not affect usability and efficiency of Belenios. We also claim that (everlasting) privacy is not affected, but this has to be formally proved. The bulletin board has the same structure, but the order in

S. Baloglu et al.

Table 1. Verifiability analysis of the variants of Belenios.

Φ/\mathcal{A}_j	Belenios*			Belenios _{tr}		Belenios _{id}		Belenios+		
	\mathcal{A}_1	\mathcal{A}_2	\mathcal{A}_3	\mathcal{A}_1	\mathcal{A}_2	\mathcal{A}_1	\mathcal{A}_3	\mathcal{A}_1	\mathcal{A}_2	\mathcal{A}_3
Φ_{iv}^h	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
Φ_{eli}	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Φ_{cl}	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
Φ_{res}^\bullet	✓	✓	✗	✓	✓	✓	✗	✓	✓	✗
Φ_{res}°	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓

* : Verification results for Belenios as in [9].

Table 2. Execution times for the verification of verifiability of Belenios+.

#b/ \mathcal{A}_j	Belenios+		
	\mathcal{A}_1	\mathcal{A}_2	\mathcal{A}_3
2 ballots per voter	17 sec	8 sec	57 sec
3 ballots per voter	1 min	33 sec	2 min 47 sec
4 ballots per voter	12 min 6 sec	15 min	15 min 53 sec

which all ballots are cast for a given credential should be clear. Our open problems are related to the formal verification and to the design of electronic voting protocols. Our specification makes certain abstractions that should be lifted or formally justified, for greater confidence in results. The most important abstraction is the one limiting the number of ballots to four for each voter. Concerning the design, our techniques still do not achieve the stronger notion of election verifiability, that prevents the adversary from casting ballots even for honest voters that have not verified their ballots. We also think election verifiability could be achieved in stronger corruption scenarios, e.g. when both the registrar and the server are (partially) corrupted. For example, it could be interesting to achieve public verifiability for the fact that each ballot is associated to an eligible voter, while perfectly hiding the actual identity of the voter. This would limit the corruption abilities of the registrar who generates the public credentials, without relying on the server to perform the verification.

Acknowledgement

This work was supported by the Luxembourg National Research Fund (FNR) and the Research Council of Norway for the joint INTER project SURCVS (No. 11747298).

References

1. Helios - Verifiable online elections, <https://heliosvoting.org/>
2. Belenios - Verifiable online voting system, <https://belenios.org/>
3. Tamarin prover, <https://tamarin-prover.github.io>
4. The Coq proof assistant, <https://coq.inria.fr/>
5. Tamarin specifications for the variants of Belenios, <https://github.com/sbaloglu/tamarin-codes/tree/main/belenios-zkp>
6. Adida, B.: Helios: Web-based open-audit voting. In: van Oorschot, P.C. (ed.) Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008, San Jose, CA, USA. pp. 335–348. USENIX Association (2008), http://www.usenix.org/events/sec08/tech/full_papers/adida/adida.pdf
7. Adida, B., De Marneffe, O., Pereira, O., Quisquater, J.J.: Electing a university president using open-audit voting: Analysis of real-world use of helios. In: 2009 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections. Usenix (2009)
8. Arapinis, M., Cortier, V., Kremer, S.: When are three voters enough for privacy properties? In: Computer Security - ESORICS - 21st European Symposium on Research in Computer Security. Lecture Notes in Computer Science, vol. 9879, pp. 241–260. Springer (2016). https://doi.org/10.1007/978-3-319-45741-3_13
9. Baloglu, S., Bursuc, S., Mauw, S., Pang, J.: Election verifiability revisited: Automated security proofs and attacks on Helios and Belenios. In: 34th IEEE Computer Security Foundations Symposium (2021), <https://eprint.iacr.org/2020/982>
10. Castéran, P., Bertot, Y.: Interactive theorem proving and program development. Coq’Art: The Calculus of inductive constructions. Texts in Theoretical Computer Science, Springer Verlag (2004), <https://hal.archives-ouvertes.fr/hal-00344237>
11. Clarkson, M.R., Chong, S., Myers, A.C.: Civitas: Toward a secure voting system. In: 2008 IEEE Symposium on Security and Privacy (S&P 2008), 18-21 May 2008, Oakland, California, USA. pp. 354–368 (2008). <https://doi.org/10.1109/SP.2008.32>
12. Comon-Lundh, H., Cortier, V.: Security properties: Two agents are sufficient. In: Degano, P. (ed.) Programming Languages and Systems, 12th European Symposium on Programming, ESOP 2003. Lecture Notes in Computer Science, vol. 2618, pp. 99–113. Springer (2003). https://doi.org/10.1007/3-540-36575-3_8
13. Cortier, V., Dallon, A., Delaune, S.: Bounding the number of agents, for equivalence too. In: Piessens, F., Viganò, L. (eds.) Principles of Security and Trust (POST). Lecture Notes in Computer Science, vol. 9635, pp. 211–232. Springer (2016). https://doi.org/10.1007/978-3-662-49635-0_11
14. Cortier, V., Drăgan, C.C., Dupressoir, F., Warinschi, B.: Machine-checked proofs for electronic voting: Privacy and verifiability for Belenios. In: Proceedings of the 31st IEEE Computer Security Foundations Symposium. pp. 298–312. IEEE Computer Society (2018). <https://doi.org/10.1109/CSF.2018.00029>
15. Cortier, V., Filipiak, A., Lallemand, J.: BeleniosVS: Secrecy and verifiability against a corrupted voting device. In: 32nd IEEE Computer Security Foundations Symposium. pp. 367–381 (2019). <https://doi.org/10.1109/CSF.2019.00032>
16. Cortier, V., Gaudry, P., Glondu, S.: Belenios: A simple private and verifiable electronic voting system. In: Guttman, J.D., Landwehr, C.E., Meseguer, J., Pavlovic, D. (eds.) Foundations of Security, Protocols, and Equational Reasoning - Essays Dedicated to Catherine A. Meadows. Lecture Notes in Computer Science, vol. 11565, pp. 214–238. Springer (2019). https://doi.org/10.1007/978-3-030-19052-1_14

S. Baloglu et al.

17. Cortier, V., Smyth, B.: Attacking and fixing Helios: An analysis of ballot secrecy. *Journal of Computer Security* **21**(1), 89–148 (2013). <https://doi.org/10.3233/JCS-2012-0458>
18. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: *Advances in Cryptology, Proceedings of CRYPTO '84*, Santa Barbara. pp. 10–18 (1984). https://doi.org/10.1007/3-540-39568-7_2
19. Hess, A.V., Mödersheim, S., Brucker, A.D., Schlichtkrull, A.: Performing security proofs of stateful protocols. In: *34th IEEE Computer Security Foundations Symposium (CSF)*. vol. 1, pp. 143–158. IEEE (2021). <https://doi.org/10.1109/CSF51468.2021.00006>, <https://www.brucker.ch/bibliography/abstract/hess.ea-performing-2021>
20. Hirt, M., Sako, K.: Efficient receipt-free voting based on homomorphic encryption. In: *Advances in Cryptology - EUROCRYPT 2000*. pp. 539–556. Springer (2000)
21. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, WPES*. pp. 61–70 (2005). <https://doi.org/10.1145/1102199.1102213>
22. Küsters, R., Truderung, T., Vogt, A.: Verifiability, privacy, and coercion-resistance: New insights from a case study. In: *32nd IEEE Symposium on Security and Privacy*. pp. 538–553. IEEE Computer Society (2011). <https://doi.org/10.1109/SP.2011.21>
23. Küsters, R., Truderung, T., Vogt, A.: Clash attacks on the verifiability of e-voting systems. In: *33rd IEEE Symposium on Security and Privacy*. pp. 395–409. IEEE Computer Society (2012). <https://doi.org/10.1109/SP.2012.32>
24. Meier, S., Schmidt, B., Cremers, C., Basin, D.A.: The TAMARIN prover for the symbolic analysis of security protocols. In: *25th International Conference on Computer Aided Verification. Lecture Notes in Computer Science*, vol. 8044. Springer (2013). https://doi.org/10.1007/978-3-642-39799-8_48
25. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*. p. 129–140. CRYPTO '91, Springer-Verlag (1991)
26. Pereira, O., Wallach, D.S.: Clash attacks and the star-vote system. In: Krimmer, R., Volkamer, M., Binder, N.B., Kersting, N., Pereira, O., Schürmann, C. (eds.) *Electronic Voting - Second International Joint Conference, E-Vote-ID*. *Lecture Notes in Computer Science*, vol. 10615, pp. 228–247. Springer (2017). https://doi.org/10.1007/978-3-319-68687-5_14
27. Schmidt, B., Meier, S., Cremers, C.J.F., Basin, D.A.: Automated analysis of diffie-hellman protocols and advanced security properties. In: *25th IEEE Computer Security Foundations Symposium, (CSF'12)*. pp. 78–94. IEEE Computer Society (2012). <https://doi.org/10.1109/CSF.2012.25>

Internet Voting: Behavioral Aspects

Party Cues and Trust in Remote Internet Voting: Data from Estonia 2005-2019*

Piret Ehin (✉)^{1,2}[0000–0001–9398–1730] and Mihkel Solvak^{1,3}[0000–0003–0179–4036]

¹ Johan Skytte Institute of Political Studies, University of Tartu, Lossi 36, 51003, Tartu, Estonia

² piret.ehin@ut.ee

³ mihkel.solvak@ut.ee

Abstract. Trust is crucial for the adoption and use of new technologies. This paper seeks to advance our knowledge of why people trust or distrust disruptive electoral technologies such as remote internet voting. It argues that because of the complexity of the systems in question, most potential users are unable to form independent opinions on the system’s trustworthiness and are likely to rely on cues provided by trusted social actors such as their preferred political parties. The paper develops a set of hypotheses from this conjecture, and tests these with survey data on approximately 5200 Estonian voters in the context of 11 elections held between 2005 and 2019. The findings suggest that partisan attachments are an important determinant of trust in e-voting and that the partisan gap in trust cannot be reduced to differences in socio-demographic voter profiles. Our results, however, do not support the conjecture that less educated individuals are particularly likely to take cues from their preferred parties when assessing the trustworthiness of e-voting.

Keywords: e-voting · internet voting · trust

1 Introduction

In recent decades, trust has become an important focus in technology studies. As a precondition for the adoption and use of new technologies [35, 26], trust can make or break specific innovations, with potentially far-reaching and cumulative macro-societal effects. In the context of the ongoing digital transformation affecting all spheres of life, it is vital to understand the nature, sources and effects of trust in the context of technological change.

The growing literature on the subject has clarified important conceptual questions and produced notable empirical findings. The conceptual work has focused on the role of uncertainty and vulnerability in trust situations, differences between various objects of trust, such as people, organizations or technologies, as well as the relevant properties of the trustor, trustee, and the broader institutional context. In terms of explaining trust in new technologies, the literature

* The work for this paper has received funding from European Union’s Horizon 2020 research and innovation programme under grant agreement No. 857622

P. Ehin and M. Solvak

has tended to prioritize user perceptions of the functionality and reliability of specific technologies. In doing so, the literature on trust converges with technology acceptance models which emphasize perceived usefulness and perceived ease-of-use [10]. However, the literature has to date paid limited attention to how cognitively constrained individuals form opinions and beliefs about highly complex technological systems.

This paper focuses on the proposition that when forming beliefs about the trustworthiness of new technologies, potential users rely on cognitive shortcuts, taking cues from trusted social actors. Grounded in well-established theories of bounded rationality, this approach postulates that when forming judgments about new technologies, people behave as cognitive misers who rely on heuristics in order to reduce the time and effort associated with making up one's mind. Considering the complexity of new digital technologies as well as the rapid pace of technological replacement, the cue-taking approach has potential to lead to new insights about the determinants and dynamics of trust.

We use cue-taking theory to explain popular trust in remote internet voting (e-voting) in Estonia. E-voting is a disruptive technology that significantly alters the calculations and behavior of stakeholders in the electoral process, including voters, parties, candidates and electoral authorities [23]. Estonia introduced remote internet voting in 2005 and has used it since then in all local, national and European Parliament elections. Usage rates have grown rapidly, with e-votes constituting almost a half of all votes cast in national and European Parliament elections in 2019. While high and growing usage rates are suggestive of high levels of trust, our data shows that Estonian voters differ greatly in terms of the extent to which they trust e-voting. We derive a set of hypotheses about partisan attachments and voter trust in e-voting, and test these with survey data on approximately 5200 voters in the context of 11 elections held between 2005 and 2019.

This paper is organized in six sections. The next section examines the concept of trust in the context of technological innovation. The third section revisits the literature on cognitive shortcuts in opinion formation, and presents the argument that voters take cues from their preferred political parties in forming beliefs about e-voting. The fourth section introduces the research design, data and methods. The fifth section describes the positions of Estonia's main political parties on internet voting. The sixth section presents the results of the analysis, focusing on the level, correlates and predictors of trust in e-voting. We conclude with a brief discussion of the implications of our findings.

2 The Concept of Trust in the Context of Technological Innovation

Trust is generally understood as belief in the reliability, truth, or ability of someone or something. Trust has been defined in various ways in different disciplines, and the copious literature on the nature, causes and effects of trust has suffered from several problems including the lack of conceptual clarity and specificity. An

Party Cues and Trust in Remote Internet Voting

Integrative Model of Organizational Trust that stands out for conceptual rigor and underlies a large body of subsequent scholarship defines trust as “willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party” [28]. This model distinguishes between the characteristics of the trustor (propensity to trust) and the characteristics of the trustee (factors of perceived trustworthiness, including ability, benevolence and integrity). It argues that trust refers to the trustor’s willingness to enter into a risk-taking relationship with the trustee, and makes an important distinction between trust as a belief and trusting behavior [28].

While most of the scholarship on trust is concerned with trust in people or organizations, a recent strand of research focuses on trust in technology. Several studies have proposed relevant definitions and measures, arguing that we need a better understanding of what makes technology itself trustworthy, “irrespective of the people and human structures that surround the technology” [29, p. 2]. This approach has strong affinities with technology acceptance models which emphasize the inherent characteristics of specific technologies, such as perceived usefulness and ease-of-use [10] as well as performance expectancy and effort expectancy [38].

Focusing on technology as the object of trust calls for specifying how trust in technology differs from trust in people, and what the implications of these differences are. To trust a person is to trust “a volitional and moral agent” while to trust technology means to trust “a human-created artifact with a limited range of capabilities” that lacks free will and moral agency [29, p. 5]. However, these differences do not challenge the basic definition of trust as a belief that the trustee has the attributes necessary to perform as expected in a situation. Both types of trust are compatible with definitions that emphasize vulnerability and willingness to assume risks as being central to trust. Both types of trust are affected by contextual conditions such as situational normality and structural assurance which refer to the belief that risks will not materialize because the situation is “normal, favorable, or well-ordered” and because “promises, contracts, regulations and guarantees are in place” [29, 30]. Furthermore, it is important to understand that the diverse objects of trust may form complex systems in which technologies, people, organizations, and contextual conditions such as institutional and legal settings are intertwined and interdependent. Whether people distinguish among the different components of the system, whether they trust some components more than others, and how trust in specific components affects trust in the system as a whole remain questions for empirical inquiry.

The above clarifications enable us to spell out what we mean by trust in remote internet voting. The trustor is the potential user – i.e. a person eligible to vote. The object of trust is a system consisting of people, organizations, institutions, laws, rules, norms and specific technologies. The system is highly complex, consisting of multiple interconnected components each of which can constitute a separate object of trust. For instance, a user could have different levels of trust in each of the specific technologies used (e.g. ID cards, authentication,

P. Ehin and M. Solvak

e-voting software, vote encryption systems, protocols and algorithms, servers, etc), as well as in people and organizations involved in the design, production, testing, operation, control, promotion and evaluation of these technologies (including developers, tech companies, governments, lawyers, electoral authorities, etc). Importantly, a lack of trust in one specific component of the system may undermine trust in the system as a whole. Remote internet voting entails a plethora of potential vulnerabilities and risks, including the risk that the vote is not cast as intended, that the vote cast does not remain secret, and that by downloading e-voting applications, users infect their devices with viruses and malware. Beyond personal risks, e-voting can be associated with a range of macro-level risks (e.g. failure to conduct free and fair elections, a crisis or breakdown of democracy). Trust in remote internet voting thus means willingness to rely on the diverse components of such a voting system based on the expectation that the system performs its declared functions (secure and fast location-independent voting in free and fair elections) irrespective of the voters' ability to monitor or control the system.

While the growing literature on trust in technology has done much to clarify the concept and illuminate the sources and effects of trust, it has not yet paid sufficient attention to the question of how people form beliefs about highly complex (systems of) objects. Arguing that the literature on trust in technology would benefit from insights from the broader literature on opinion and belief formation on complex issues, the next section revisits the literature on cognitive heuristics and contemplates the role of political parties in shaping public beliefs about the trustworthiness of e-voting.

3 Trust or Not to Trust Technology? Taking Cues from Political Parties

For decades, scholarship on opinion formation and decision-making has emphasized the cognitive limitations of human judgment, arguing that individuals tend to rely on cognitive heuristics in order to reduce informational and computational costs [32, 34, 17, 6]. A heuristic is a mental shortcut that leads to fast, frugal and mostly accurate decisions in many situations characterized by uncertainty [18]. However, reliance on cognitive heuristics is also associated with errors and reduced accuracy, cognitive bias, stereotyping and prejudice [34].

Cue-taking is one type of heuristic that individuals use in order to reduce cognitive effort involved in problem-solving, opinion formation and decision-making. Because of the cognitive and temporal costs of rational reasoning, individuals look to other trusted social actors, such as political elites, for signals suggesting what to think or how to behave [27, 41]. The likelihood that individuals rely on elite cues when forming opinions and making decisions increases when information is scarce or difficult to obtain, when issues are complex, uncertainty is high, when time is constrained, and when the ability to process information is low.

In the context of competitive democracies, a theory of opinion formation based on elite cues must take into account partisanship. There is a large and

Party Cues and Trust in Remote Internet Voting

diverse literature focusing on party cues and the effects of party attachments on individual opinions and decision-making [9, 22, 3, 7, 8]. It is argued that citizens follow the lead of the party they sympathize with the most in forming policy opinions, and are particularly likely to do so when the issues in question are complex. There is significant evidence that individuals rely on party cues when making up their minds on issues such as European integration [2, 20, 19, 31], the state of the national economy [5], climate change [12], foreign policy [4] or nuclear energy [25]. While much of the literature on partisanship effects has focused on the United States, party attachments have been shown to affect policy opinions in a variety of contexts, including multi-party systems and new democracies [7].

Despite the prominence of the party cues theory in political research, it seems that this approach has not been applied to explaining opinion formation on new digital technologies. Widely used models of technology acceptance and use, such as UTAUT [38] include social influence as one of the explanatory factors. Social influence is defined as "the degree to which an individual perceives that important others believe he or she should use the new system" [38, p. 451]. However, the category of social influence in these models is concerned with subjective norms, culture, image and reputation, not with cue-taking as a form of cognitive shortcut.

There are three interrelated reasons why opinion formation on e-voting is highly likely to involve cue-taking from political parties. First, because remote internet voting is quick and convenient, saving voters time and money, many voters will want to use it. Before doing so, however, they need to determine whether the system can be trusted. In other words, there are strong incentives to form an opinion on e-voting in the first place. Second, the issue is highly complex: few voters have the time and ability to form independent opinions on the trustworthiness of the technological, legal and institutional aspects of remote internet voting. Thus, voters are highly likely to look for and rely on informational and computational shortcuts in forming opinions. Third, political parties can be expected to be important cue-givers because they have much at stake in the introduction of new technologies that transform the electoral process. Technological innovations that alter both the cost-benefit calculations involved in the act of voting, as well as perceptions, norms and understandings related to elections, have the potential to differentially impact electoral support for specific parties. Thus, parties are likely to form positions, corresponding to their perception of how e-voting affects their electoral prospects and those of their contenders. They may frame new technologies in particular ways, seeking to legitimize or de-legitimize their use. In sum, the combination of these three factors makes it highly likely that parties engage in cue-giving and voters in cue-taking regarding e-voting.

We derive the following hypotheses from the above discussion:

H1: *Citizens who vote for parties that endorse remote internet voting are more likely to trust remote internet voting than citizens who vote for parties that criticize this voting mode.*

P. Ehin and M. Solvak

H2: The partisan gap in trust in remote internet voting cannot be reduced to differences in the socio-demographic profiles of party voters.

H3: The effect of party cues on an individual's trust in remote internet voting is conditioned by the individual's level of cognitive sophistication.

4 Research Design, Data and Methods

The hypotheses specified above are tested using individual-level survey data as well as party-level data from Estonia from 2005 to 2019. During these fifteen years, eleven nation-wide elections with an e-voting option have been held, including four local, four national and three European Parliament elections. Estonia remains the only country in the world that offers all of its voters the opportunity to cast a vote online. E-voting has been available in all nation-wide elections since 2005, and the share of e-votes has grown steadily, reaching almost 50 per cent of all votes cast in 2019. In Estonia, e-voting is highly institutionalized and has become part of the regular framework for conducting elections. In the context of a study focusing on trust in e-voting, this means that the object of trust is an existing, widely used system that all voters have the option of using. This differentiates the Estonian case from all other currently existing electoral contexts in the world. For more information on the organization and uptake of internet voting in Estonia see [1, 33, 36].

In this context, Estonian political parties have had more reason and more time to form positions on e-voting than their counterparts around the world. Party positions have evolved together with the Estonian e-voting system, and have both reflected and influenced societal and expert debates on the matter. As there have been no initiatives to systematically collect data on Estonian parties' positions on internet voting, such as a survey or manifesto study, this study infers party positions from a range of available sources, including votes in the parliament, party manifestos and campaign materials, statements by party leaders and officials, as well as media and social media coverage of party activities. This analysis focuses on the positions of six largest parties, three of which have, at various times, expressed skepticism towards e-voting.

To examine voter trust in e-voting, we use individual-level survey data from the Estonian electronic voter study 2005-2019, which is comprised of 11 post-election cross-sectional surveys covering all elections in which the option of remote internet voting has been available. Each survey had a sample size of roughly 1000 respondents, and the samples are representative of eligible voters in terms of age, gender, ethnicity and region. We focus only on self-reported voters, resulting in a dataset of roughly 5200 respondents.

We use the following measure of trust in e-voting: "Do you trust the procedure of internet voting?". Answers to this question were recorded on a four-category Likert scale between 2005 and 2011 and on a 0-10 scale since 2013. In both cases, we split the responses mid-scale and turned the variable into a binary trust variable (0 - do not trust; 1 - trust) to be able to compare effects across

Party Cues and Trust in Remote Internet Voting

the years. Respondents who chose category 5 on the 0-10 scale were randomly assigned to either side.

To investigate the hypotheses we employ the following approaches. After describing the level of trust in e-voting over the years, we turn to the question of whether trust in e-voting differs from trust in political institutions as well as trust in online transactions. To answer this question, we examine correlation matrices of various survey items. Second, we examine the dynamics of trust in e-voting over time according to party choice. Given that some Estonian parties have changed their stances on internet voting over time, an examination of whether and how voter attitudes have followed these changes provides particularly compelling evidence of cue-taking. Third, we run eleven separate logit models in the following setup:

$$\ln \left\{ \frac{Pr(trust_t = 1)}{1 - Pr(trust_t = 1)} \right\} = \beta_0 + \beta_1 demographics_t + \beta_2 partychoice_t + \beta_3 trust_t \quad (1)$$

The dependent variable is trust in e-voting in election t and the independent variables are standard socio-demographics (age, gender, income, education), weekly internet usage frequency, self-reported computer skills of the voter, party choice in the given election, average trust in other state institutions and trust in internet transactions. We run a separate model for each election and include independent variables stepwise in order to assess whether and to what extent party cues override the effects of other factors. Given that we include trust in internet transactions, internet usage intensity as well as self-reported skills as controls, this approach should constitute a rigorous test of the party cues and non-reducibility hypotheses (H1 and H2). Finally, to test the sophistication hypothesis (H3) we examine the predictive margins of trust by party choice and education level.

Below, we will first elaborate on the positions of the Estonian political parties before turning to an analysis of voter attitudes.

5 The Positions of Estonian Political Parties on E-voting

Despite a sustained political commitment to developing internet voting that spans two decades and ten coalition governments, there has been significant partisan conflict over e-voting in Estonia. Out of the six main parliamentary parties, three (Pro Patria, the Reform Party and Social Democrats) have endorsed and promoted e-voting. Pro Patria and the Reform Party, both on the center-right, were leading government parties in the early 2000s when expert and political discussions on e-voting were first launched, the decision to deploy internet voting was taken, and the necessary legislation prepared. The liberal, pro-market Reform Party has been the dominant government party during the observed period, leading coalition governments from April 2005 to November 2016. For most of this period, it was in coalition with Pro Patria and the Social Democrats. Throughout this period, the three parties' positions on internet voting – along

P. Ehin and M. Solvak

with that of the government as a whole – have been highly positive, depicting e-voting as an important element of the Estonian e-state, a long-standing priority of Reform-led governments.

Three other major parties - the Center Party, the People’s Union, and its successor, the Conservative People’s Party – have adopted critical stances on internet voting at various points of time and with varying levels of intensity. For most of the observed period, the three parties were in opposition (with the exception of the Center serving as a junior partner in a Reform-led coalition government from April 2005 to April 2007, and the People’s Union being included in government from April 2003 to April 2005). In November 2016, however, Center became the leading government party, ruling, initially, together with Pro Patria and the Social Democrats, and then, following March 2019 elections, with the Conservative People’s Party and Pro Patria. Below, we summarize available evidence about negative cuing by the Center Party, the People’s Union, and the Conservative People’s Party.

The Center Party, a liberal centrist force with a reoccurring populist streak, was one of the two parties that voted against the introduction of internet voting in 2005. Between 2005 and 2013, it voiced occasional criticism of e-voting. For instance, following the 2011 national elections, MP Ando Leps claimed that the Estonian e-voting system was ”completely untrustworthy,” rendering the election legally invalid [37]. The Party stepped up criticism of e-voting after the October 2013 local elections. Party Chairman Edgar Savisaar published an article in the party newspaper *Kesknädal* in which he claimed that right-wing parties won elections by forging election results [39]. In spring 2013, an NGO connected to the Center Party ran a street campaign in Tallinn, featuring 68 posters with slogans such as “They may delete your vote,” “Every e-vote is a potential threat to Estonia’s independence” and “They can give your vote to whoever they want” (ibid.) In both 2011 and 2014, the Party helped fund visits of foreign experts who produced critical reports of the e-voting system. In 2014, the Center Party Board sent a letter to Estonian and EU top officials requesting immediate cancellation of e-voting due to “fundamental security problems” [40]. In April 2015, the Party’s Council adopted a resolution which claimed that e-voting was a security risk, and argued that e-voting violates the requirement of uniformity and secrecy [24]. The resolution said that even if government parties had not abused e-voting to date, such abuse may occur in the future (ibid).

However, the Center Party appears to have discontinued its criticism of e-voting after it became the leading government party in November 2016. Still, in March 2017, the party proposed a bill which foresaw shortening the e-voting period from seven days to three [39]. However, Party Chairman and Prime Minister Jüri Ratas publicly confirmed that the government endorses internet voting. In September 2017, the government led by Ratas had to manage the most serious crisis in the history of Estonian e-government which occurred after foreign scientists found a vulnerability affecting hundreds of thousands of ID cards used in Estonia [21]. With the reputation of the Estonian e-government system at stake, the government led by Ratas worked hard to solve the crisis and control dam-

Party Cues and Trust in Remote Internet Voting

age. Since the event, the Center Party has refrained from criticizing e-voting. In sum, the Center Party was critical of remote internet voting from 2005 until late 2016, while its position since November 2016 can be characterized as neutral or favorable.

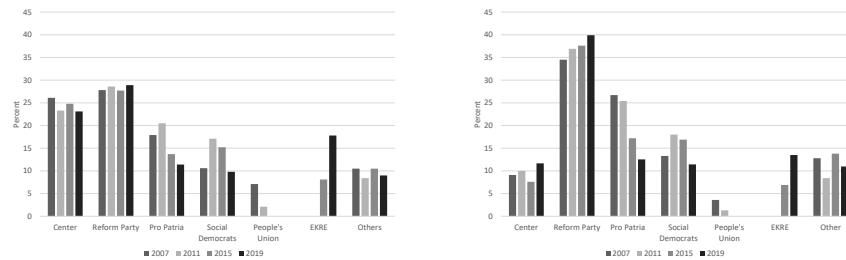
The Estonian People’s Union, a socially conservative rural party, was founded in 1999 and ceased to exist in 2012. It was one of the two parties that voted against the introduction of e-voting in 2005. Furthermore, its former Chairman Arnold Rüütel, then President of the Republic, twice refused to proclaim the law instituting internet voting, arguing that the provision that allows the voter to alter his or her e-vote violates the principle of uniformity of elections. He also asked the Supreme Court to declare the law invalid [13]. In 2006, Jaak Allik, Deputy Chair of the party group in the parliament, argued that e-voting is in principle not observable and electoral authorities are not able to ascertain whether the person who voted with a particular ID card is the legal holder of the card [11]. The party suffered major electoral losses in general elections held in spring 2007. This decline seems to coincide with the subsiding of negative rhetoric directed at e-voting.

The third major party that has criticized e-voting is the Estonian Conservative People’s Party (EKRE). Founded in 2012, the populist far-right party first gained parliamentary representation in 2015 and entered the governing coalition in spring 2019. While the party seems to have kept a low profile on e-voting during the first four years of its existence, it has, over time, turned into a vocal critic of the system. In spring 2017, Henn Põlluaas, Deputy Chairman of the party group in the parliament, called for an international audit of the Estonian e-voting system. Half a year later, EKRE filed a complaint with the Electoral Committee, demanding that internet voting in upcoming local elections be cancelled due to security vulnerabilities affecting ID cards [14]. In March 2019, Deputy Chair of the party, Martin Helme, claimed that for him, the trustworthiness of e-elections was “non-existent” because the integrity of elections “cannot be monitored or verified” [16]. Over the course of 2017-2019, EKRE’s news portal *Uued uudised* published 26 articles expressing various doubts about e-elections, pointing at shortcomings in procedures and emphasizing the need to evaluate and improve the security of the system. After EKRE joined the governing coalition in April 2019, it was assigned the portfolio of the Minister of Foreign Trade and Information Technology. Kert Kingo, who held the position for half a year in 2019, convened an e-voting working group to assess the verifiability, security and transparency of Estonia’s electronic voting system – a move that many interpreted as being politically motivated [15]. In sum, between 2017 and 2019, EKRE’s position on e-voting can be characterized as highly critical.

Differences in party positions appear to reflect the differential utilities that parties derive from e-voting. Voter uptake of e-voting varies by party choice as shown in Figure 1. While the vote shares of the two largest parties (Reform and Center) have been fairly comparable, hovering between 23 and 29 per cent in national elections (Figure 1a), the Reform Party gets about four times as many e-votes as its main political opponent (Figure 1b). Also, Pro Patria and

P. Ehin and M. Solvak

the Social Democrats are clearly more successful in attracting e-votes than the Center Party. Although previous studies have demonstrated that e-voting does not increase turnout or mobilize non-voters [33], it is clear that the importance of this voting channel varies greatly across parties.



(a) Party vote shares in national elections 2007-2019 (b) Party e-vote shares in national elections 2007-2019

Fig. 1: Total vote shares and e-vote shares in national elections 2007-2019

6 Voter Trust and Its Correlates: Results Based on the Estonian Electronic Voter Study

Data from the Estonian electronic voter study suggests that e-voting has enjoyed high levels of trust in Estonia since its inception. According to the first survey conducted in 2005, a few months after the first e-enabled election, about 80 per cent of the voters said that they trusted the system. The level of trust has ebbed and flowed, reaching the lowest level of 54 per cent in 2013, but recovering after that and hovering around 69-70 per cent in the two elections held in 2019.

Before proceeding to analyze the predictors of trust in e-voting, it is important to establish whether and how this type of trust is related to trust in other institutions. Running bivariate correlations (coefficients not shown due to space limitations but available from authors upon request) between trust in internet voting and trust in the parliament, government, politicians, the state and internet transactions show that trust in e-voting is correlated with trusting other institutions but the correlations are systematically weaker compared to correlations between trust in different state institutions. This is a strong indication that trust in e-voting is substantively different from trust in other state institutions and that respondents are able to distinguish e-voting from other objects and evaluate its trustworthiness separately.

Also, it is important to establish whether and how much trust matters when it comes to the decision whether to use the system or not. Figure 2 shows the

Party Cues and Trust in Remote Internet Voting

association between trust and usage of e-voting, extracted from a regression model where usage is the dependent variable and trust is an independent variable alongside conventional socio-demographic measures. The results confirm that trust is a persistent and potent predictor of usage.

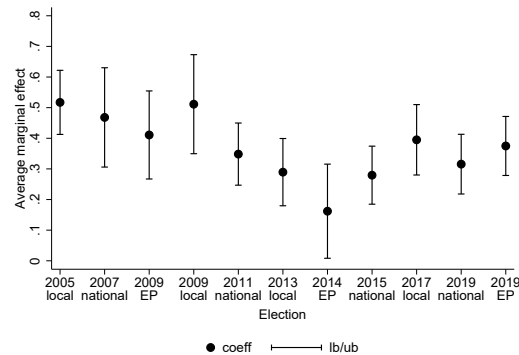
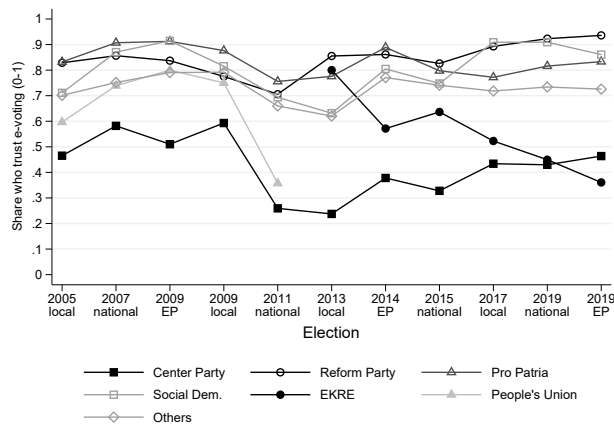


Fig. 2: The effect of trust on usage of e-voting (2005-2019)

Turning to the question of how party cues affect voter attitudes, Figure 3 shows the level of trust in e-voting by party choice over time. Multiple things stand out. First, the figure shows that party supporters fall into two distinct groups, the low-trust (Center, EKRE voters) and the high-trust group (all others). Second, the share of trustors fluctuates over the observed period for two parties. Trust among Center Party supporters starts out high, then plummets and then grows again. These fluctuations reflect the temporal evolution of the Center Party position on e-voting: the Party was initially indifferent towards this voting mode, then started to heavily oppose it and finally switched to positive rhetoric after becoming the leading government party in late 2016. EKRE supporters have moved from a high-trust group to a low-trust group almost linearly as the party leadership's opposition to e-voting has grown more vocal.

Next, we ran a regression model in order to ascertain the effects of party choice on trust in e-voting, controlling for socio-demographic variables as well as computer literacy, internet usage and trust in political institutions as well as internet transactions. Table 1 presents a part of the regression model output (effects of control variables not shown). A number of findings stand out. First, in the early years of e-voting, party choice was not a significant predictor of trust. Statistically significant effects of party choice appear from 2011 onwards and persist when internet usage, PC skill level, trust in internet transactions as well as socio-demographics are controlled for. Second, we see how the explanatory power of these models, especially their ability to classify low-trust voters, improves over time and then diminishes again. These fluctuations correspond to shifts in the Center Party's stance on e-voting. Third, as the reference group are Center Party

P. Ehin and M. Solvak

**Fig. 3:** Level of trust in e-voting by party choice

voters, we can conclude that the supporters of the Reform Party, Pro Patria and the Social Democrats are clearly more trusting of e-voting than Center Party supporters. This finding is in line with the significantly larger share of internet votes accruing to these three parties compared to the Center Party.

Table 1: The effects of party choice on trust in e-voting (2005-2019)

	2005 local	2007 national	2009 EP	2009 local	2011 national	2013 local	2014 EP	2015 national	2017 local	2019 national	2019 EP
Reform Party (ref: Center)	0.027 (0.058)	0.061 (0.052)	0.064 (0.050)	0.029 (0.064)	0.224 (0.127)	0.273*** (0.071)	0.238** (0.081)	0.357*** (0.081)	0.149* (0.063)	0.301*** (0.062)	0.311*** (0.073)
Pro Patria	0.041 (0.058)	0.109 * (0.050)	0.119 ** (0.047)	0.077 (0.063)	0.259 * (0.127)	0.198 ** (0.070)	0.307 *** (0.084)	0.406 *** (0.081)	0.075 (0.079)	0.162 ** (0.075)	0.229 ** (0.085)
Social Democrats	-0.074 (0.081)	0.063 (0.056)	0.100* (0.048)	0.014 (0.071)	0.226 (0.127)	0.110 (0.068)	0.237** (0.075)	0.397*** (0.073)	0.227** (0.074)	0.281*** (0.067)	0.241*** (0.075)
People's Union	-0.101 (0.086)	0.001 (0.081)	-	-	0.234 (0.184)	na	na	na	na	na	na
EKRE	na	na	na	na	na	0.161 (0.217)	0.080 (0.129)	0.324*** (0.089)	-0.102 (0.088)	-0.011 (0.073)	-0.176 (0.098)
Other party	-0.011 (0.059)	0.052 (0.053)	0.027 (0.047)	0.079 (0.056)	0.232 (0.120)	0.217*** (0.058)	0.178* (0.077)	0.325*** (0.077)	0.101 (0.058)	0.123 (0.072)	0.129 (0.079)
Sensitivity	98.45	99.59	98.38	98.94	96.87	85.8	93.42	88.89	92.03	91.07	92.92
Specificity	16.67	2.00	27.66	10.64	34.21	78.33	68.89	63.83	53.73	60.26	52.14
Pseudo R^2	0.176	0.198	0.453	0.171	0.383	0.619	0.584	0.422	0.478	0.514	0.440
Observations	459	532	479	426	395	472	318	530	548	554	484

Average marginal effects with standard errors in parentheses.

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Finally, we turn to the question of whether the effect of party cues on trust in e-voting is moderated by the voter's cognitive sophistication. Figure 4 shows the predictive margins of trust in e-voting according to the highest level of education attained. Overall, the results suggest that trust in e-voting does not depend

Party Cues and Trust in Remote Internet Voting

on sophistication. However, regardless of which party we focus on, a pattern emerges where in the first years of e-voting lower education was associated with higher levels of predicted trust, while in the last four or five elections, the highly educated are more prone to trust e-voting than the less educated. It is not clear why such a reversal has occurred - because confidence intervals for the educational categories overlap, we cannot substantively interpret these results. What is clear, however, is that when trust in e-voting declines among supporters of a particular party, it does so across all educational categories.

7 Conclusions

This study sought to contribute to the burgeoning literature on trust in new technologies by systematically evaluating the proposition that voters take cues from political parties when evaluating technologically complex voting systems such as remote internet voting. It derived three hypotheses from the discussion about the correlates and predictors of trust, and tested these with individual and party-level data from Estonia, covering the period 2005-2019.

The results lend support to the party cues hypothesis, which postulated that citizens who vote for parties that endorse e-voting are more likely to trust e-voting than citizens who vote for parties that criticize this voting mode. In the case of Estonia, this means that voters who cast a vote for the Center Party, the People's Union or the Conservative People's Party have been less likely to trust e-voting than voters who voted for other parties. The second hypothesis, which posited that the partisan gap in trust cannot be reduced to differences in the socio-demographic profiles of party voters, was also confirmed. The effects of party choice on trust for e-voting persisted when a variety of socio-demographic controls, along with general trust in political institutions, computer literacy, internet usage and trust in internet transactions were controlled for. The third hypothesis which expected the effect of party cues on an individual's trust in e-voting to be conditioned by the level of political sophistication was not confirmed.

These results confirm the potential of societal actors to shape mass perceptions of new technologies, with consequences for the uptake and use of such technologies. The fact that political parties have 'skin in the game' in debates about voting modes increases the risk that e-voting will become politicized. To the extent that the voters' propensity to use e-voting technology varies by party choice, parties derive differential utility from the availability of this voting mode. Feedback effects among such utility, the cues parties send to their voters, and the resulting differences in usage rates have the potential to lead to a growing polarization of trust in and usage of e-voting along party lines.

References

1. Alvarez, R., Nagler, J.: Likely consequences of internet voting for political representation. *The Loyola Los Angeles Law Review* **34**, 1115–1153 (2000)

P. Ehin and M. Solvak

2. Anderson, C.J.: When in Doubt, Use Proxies: Attitudes Toward Domestic Politics and Support for European Integration. *Comparative Political Studies* **31**(5), 569–601 (1998)
3. Bartels, L.: Beyond the running tally: Partisan bias in political perceptions. *Political Behavior* **24**, 117–150 (2002)
4. Berinsky, A.J.: Assuming the costs of war: events, elites, and American public support for military conflict. *Journal of Politics* **69**(4), 975–997 (2007)
5. Bisgaard, M., Slothuus, R.: Partisan elites as culprits? how party cues shape partisan perceptual gaps. *American Journal of Political Science* **62**, 456–469 (2018)
6. Bobadilla-Suarez, S., Love, B.: Fast or frugal, but not both: Decision heuristics under time pressure. *Journal of Experimental Psychology, Learning, Memory and Cognition* **44**(1), 24–33 (2018)
7. Brader, T., Tucker, J.A.: Following the party’s lead: Party cues, policy opinion, and the power of partisanship in three multiparty systems. *Comparative Politics* **44**(4), 403–420 (2012)
8. Bullock, J.: Party Cues. In: Suhay, E., Grofman, B., Trechsel, A.H. (eds.) *The Oxford Handbook of Electoral Persuasion*. Oxford University Press (2019)
9. Campbell, A., M. Converse, P.E., Miller, W.E., Stokes, D.E.: *The American Voter*. New York: Wiley (1960)
10. Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly* **13**(3), 319–340 (1989)
11. Delfi: Rahvaliid jätkab sõda e-hääletuse vastu. <https://www.delfi.ee/news/paevauudised/eesti/rahvaliid-jatkab-soda-e-haaletuse-vastu?id=13112352> (2006), [Online; accessed 03-November-2019]
12. Ehret, P., Van Boven, L., Sherman, D.K.: Partisan barriers to bipartisanship: Understanding climate policy polarization. *Social Psychological and Personality Science* **9**(3), 308–318 (2018)
13. ERR: Rüütel sai e-valimistega lüüa. <https://www.err.ee/435363/ruutel-sai-e-valimistega-luu> (2005), [Online; accessed 03-November-2019]
14. ERR: Ekre vaidlustas e-valimiste korraldamise. <https://www.err.ee/617818/ekre-vaidlustas-e-valimiste-korraldamise> (2017), [Online; accessed 03-November-2019]
15. ERR: It minister convenes inaugural e-voting working group. <https://news.err.ee/958188/it-minister-convenes-inaugural-e-voting-working-group> (2019), [Online; accessed 03-November-2019]
16. ERR: Valimisteenistus ekre-le: e-valimised on vaadeldavad ja kontrollitavad. <https://www.err.ee/921784/valimisteenistus-ekre-le-e-valimised-on-vaadeldavad-ja-kontrollitavad> (2019), [Online; accessed 03-November-2019]
17. Gigerenzer, G.: *Rationality for mortals: How people cope with uncertainty*. New York: Oxford University Press (2008)
18. Gigerenzer, G., Gaissmaier, W.: Heuristic decision making. *Annual Review of Psychology* **62**, 451–482 (2011)
19. Hobolt, S.B.: Taking cues on Europe: Voter competence and party endorsements in referendums on European integration. *European Journal of Political Research* **46**(February), 151–182 (2007)
20. Hooghe, L., Marks, G.: Calculation, community and cues: Public opinion on European integration. *European Union Politics* **6**(4), 419–443 (2005)
21. Information System Authority: ROCA Vulnerability and eID: Lessons Learned. <https://www.ria.ee/sites/default/files/content-editors/kuberturve/roca-vulnerability-and-eid-lessons-learned.pdf> (2019), [Online; accessed 03-November-2019]

Party Cues and Trust in Remote Internet Voting

22. Jacoby, W.G.: The Impact of Party Identification on Issue Attitudes. *American Journal of Political Science* **32**, 643–61 (1988)
23. Kersting, N.; Baldersheim, H.: *Electronic Voting and Democracy: A Comparative Analysis*. New York: Palgrave Macmillan (2004)
24. Keskerakond: E-riigis on suurepärase kõik peale e-valimiste. <https://www.keskerakond.ee/et/530-keskerakonna-volikogu-avaldus-e-riigis-on-suurepaerane-koik-peale-e-valimiste> (2015), [Online; accessed 03-November-2019]
25. Latre, E., Thijssen, P., Perko, T.: The party politics of nuclear energy: Party cues and public opinion regarding nuclear energy in Belgium. *Energy Research and Social Science* **47**, 192–201 (2019)
26. Lippert, S.K., Davis, M.: A conceptual model integrating trust into planned change activities to enhance technology adoption behavior. *Journal of Information Science* **32**(5), 434–448 (2006)
27. Lippmann, W.: *Public Opinion*. Harcourt, Brace and Co (1922)
28. Mayer, R.C., Davis, J.H., Schoorman, F.D.: An integrative model of organizational trust. *The Academy Management Review* **20**(3), 709–734 (1995)
29. McKnight, D.H., Carter, M., Thatcher, J.B., Clay, P.F.: Trust in a specific technology: An investigation of its components and measures. *ACM Transactions on Management Information Systems* **2**(2) (2011)
30. McKnight, D.H., Cummings, Larry, L., Chervany, N.L.: Initial trust formation in new organizational relationships. *The Academy Management Review* **23**(3), 473–490 (1998)
31. Pannico, R.: Parties are always right: the effects of party cues and policy information on attitudes towards EU issues. *West European Politics* **43**(4), 869–893 (2020). <https://doi.org/10.1080/01402382.2019.1653658>
32. Simon, H.A.: Rational choice and the structure of the environment. *Psychological Review* **63**(2), 129–138 (1956)
33. Solvak, M., Vassil, K.: *E-voting in Estonia: Technological Diffusion and Other Developments Over Ten Years (2005 - 2015)*. University of Tartu (2016)
34. Tversky, A., Kahneman, D.: Judgment under uncertainty: Heuristics and biases. *Science* **185**(4157), 1124–1131 (1974)
35. Vance, A., Elie-Dit-Cosaque, C., Straubl, D.W.: Examining trust in information technology artifacts: The effects of system quality and culture. *Journal of Management Information Systems* **24**(4), 73–100 (2008)
36. Vassil, K., Solvak, M., Vinkel, P., Trechsel, A.H., Alvarez, R.M.: The diffusion of internet voting. Usage patterns of internet voting in Estonia between 2005 and 2015. *Government Information Quarterly* **33**(3), 453–459 (2016). <https://doi.org/10.1016/j.giq.2016.06.007>
37. Veiserik, I.: Ando leps: E-hääletus riigikogu valimistel õigustühine. *Kesknädal* (23 March) (2011)
38. Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D.: User acceptance of information technology: Toward a unified view. *MIS Quarterly* **27**(3), 425–478 (2003)
39. Vester, L., Olup, N.M.: Ülevaade: Keskerakonna võitlused e-valimiste vastu. *Postimees* (5 September) (2017)
40. Villmann, Anna-Liisa.: Keskerakond nõuab Euroopa Parlamendilt e-valimiste tühistamist. <https://www.err.ee/512935/keskerakond-nouab-euroopa-parlamendilt-e-valimiste-tuhistamist> (2014), [Online; accessed 03-November-2019]
41. Zaller, J.R.: *The Nature and Origins of Mass Opinion*. Cambridge: Cambridge University Press (1992)

P. Ehin and M. Solvak

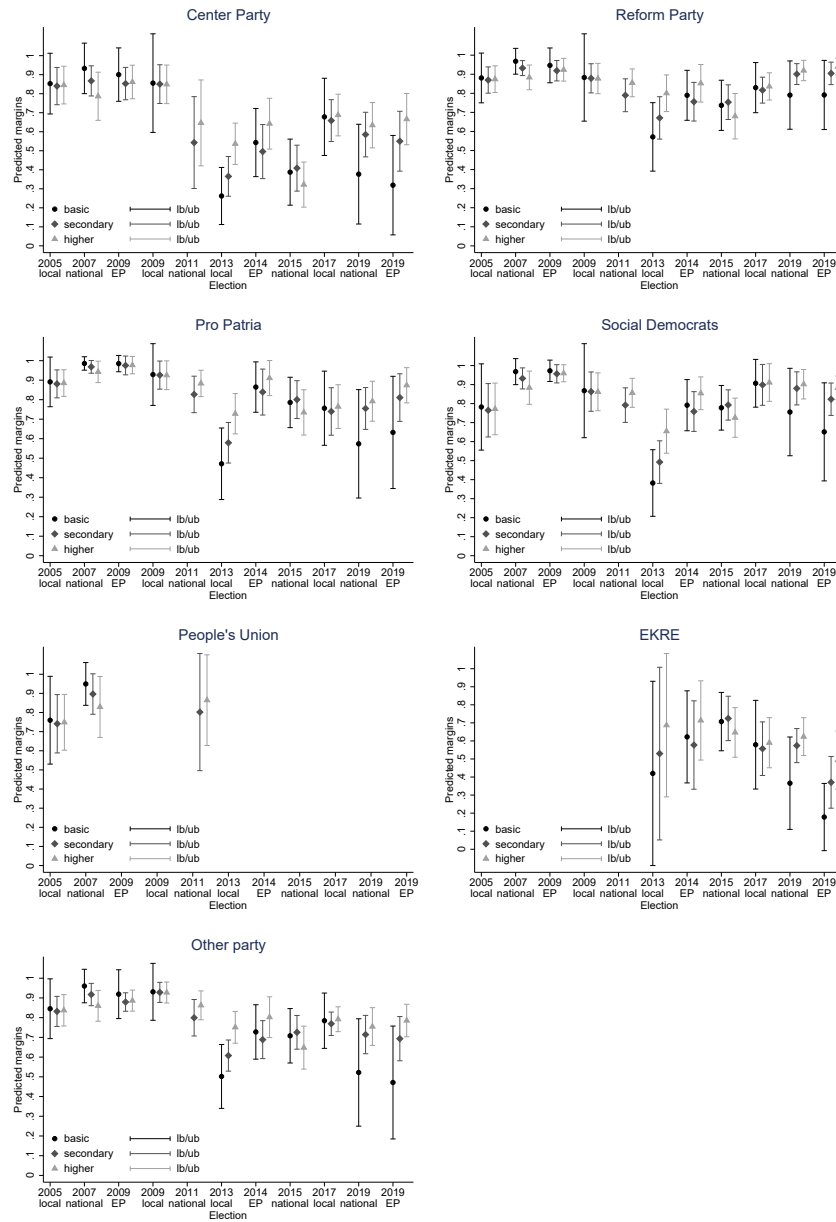


Fig. 4: The effect of education on trust in e-voting by party choice (predictive margins)

To i-vote or not to i-vote: Drivers and Barriers to the Implementation of Internet Voting

Nathan Licht¹[0000-0002-6699-9879], David Duenas-Cid^{2, 3}[0000-0002-0451-4514], Iuliia Krivonosova¹[0000-0001-7246-1373] and Robert Krimmer²[0000-0002-0873-539X]

¹ Tallinn University of Technology, Ehitajate tee 5, 12616 Tallinn, Estonia

² University of Tartu, Ülikooli 18, 50090 Tartu, Estonia

³ Kozminski University, 57/59 Jagiellonska, 03-301 Warsaw, Poland

nalich@taltech.ee

dduenas@kozminski.edu.pl

iuliia.krivonosova@taltech.ee

robert.krimmer@ut.ee

Abstract. This paper investigates the drivers and barriers of internet voting and the implications of a global pandemic for the development of the respective technology. In contrast to the expected uptake in the early 2000s of internet voting, the technology is still rather seldomly used in election systems around the world. The paper at hand explores the different forces that drive or impede internet voting adoption from a political, social, legal, organizational, contextual, economic and technological perspective. In an exploratory approach, 18 expert interviews and extensive complementary desk research were conducted.

The findings identified 15 general drivers and 15 general barriers for the process of internet voting adoption. The evidence suggests that for a large part, the political features, trust and perception are the most pivotal factors to internet voting development.

Keywords: Internet Voting, Drivers and Barriers, Framework of Internet Voting, Technology adoption, e-Democracy

1 Introduction

From Richard Buckminster Fuller [1] in the mid 20th century over Bill Gates [2], who predicted in his book *The Road Ahead* that “voters will be able to cast their ballots from home or their wallet PCs” to Apple’s CEO, Tim Cook that “dream[s] of [voting on phones]” [3], the idea of deploying remote electronic voting has been envisioned by technology leaders since the first half of the last decade. A vision that was increasingly voiced at the beginning of the early 2000s as the interest in the internet and information and communication technologies (ICT) grew bigger.

Bill Gates’ quote translated into present understandings probably refers to what is nowadays called internet voting (i-voting), which is a form of remote voting that is conducted in unsupervised environments such as one’s home. If one compares his quote with the quote by Tim Cook, it does not sound very different, despite being said around 26 years earlier. In fact, the technology has been around for over two decades and has not diffused as it was expected that it would be. During the early 2000s, a great interest in novel technology existed, and much investment occurred alongside the general developments of ICTs to enhance democratic processes. Experts and politicians back then were convinced that in the course of the following 20 years, every democratic election would be conducted via electronic voting and even using the internet [4]. Although today that is still not the reality that we live in, the quote by Tim Cook seems to reflect a still present vision for contemporary leaders to be able to conduct elections online.

Therefore, the question can be raised why i-voting has not adopted as it had been expected and what are factors that drive internet voting. Moreover, due to the current global COVID-19 pandemic, several elections that were meant to take place were postponed, and discussions about whether to implement novel, sustainable and long-term voting solutions in response to the current events have appeared [5]. Remarkably, the interest in i-voting technology has heightened due to the global developments in response to the COVID-19 pandemic [6] which makes our research more timely and relevant. The understanding of i-voting’s diffusion, its driving as well as impeding forces seem to be common questions that have been raised in academia and yet lack a holistic overview and common first understanding, which this paper aims to provide.

This paper solely focusses on i-voting, which is a specific form of electronic voting (e-voting), but for a better understanding of research intersections between these two topics, the following section depicts previous work related to both issues.

Previous works on e-voting have investigated diffusions of e-voting in Europe and drivers and barriers around e-voting [7] on adoption factors of e-voting by young people [8] the evolution of e-voting [9], the global e-voting status [10] and to provide an e-voting framework [11]. On i-voting, previous studies examined the global status quo [12, 13], studied the origins of remote online voting [4], aimed at providing a historical overview on i-voting usage [14, 15] and facilitating conditions for i-voting implementation on the examples of Estonia and Switzerland [16]. Furthermore, i-voting adoption was explicitly investigated for the Estonian case [17], and respective adoption phases were identified for the Estonian case [18]. Last, another work looked at the adoption stages and on what levels internet voting will occur [19]. This respective paper

identified two levels and five adoption stages of internet voting diffusion on which this paper is building on to investigate the respective drivers and barriers that impact the technology acceptance on these levels.

In conclusion, previous research either looked at part drivers and barriers or facilitating conditions in specific contexts. However, no comprehensive study has been conducted so far that investigates general drivers and barriers that are observable along the various adoption and trialed contexts. In line with that identified research gap, this paper poses the following research question: *What is driving internet voting and what barriers exist to further adoption?* In order to answer this question, the work at hand conducted in an exploratory way some 18 expert interviews and extensive complementary desk research. The applied methodology used for this paper, is explained subsequently.

2 Methodology

In order to study what hinders or benefits the implementation of internet voting, we want to identify its drivers and barriers. To do so, we conducted a qualitative empirical study with a nonexperimental design including expert interviews, as promoted by Brown & Hale [20]. This research was conducted using an inductive epistemological approach to acquire knowledge. The inductive process, as opposed to the deductive method, is a “bottom-up [technique in which] evidence is collected first, [from the observation of the world] and knowledge and theories built from this” [21]. In order to guide the data analysis, a conceptual model was created *ad hoc*¹, integrating propositions included in five innovation diffusion theories. This model (see Figure 1) explains how different dimensions are embedded into one context that shapes the process of diffusion of internet voting, in an evolutionary process that is impacted by perceptions, adopter categories and discourses. Furthermore, it establishes the differentiation of internet voting adoption on two levels: political and individual. The model presents five dimensions, various stakeholders and factors that impact the technology acceptance process within societies.

In order to make this paper better readable, we will briefly introduce some necessary stakeholders. First, the *relevant social groups* [23] which have a need or specific interest in the new innovation which creates a demand within society for the respective technology. Second, *change agents* or *opinion leaders* [24] shape public debate around an innovation due to their privileged position in society. Third, *individual drivers* are the citizens themselves who would be accepting technology based on the expected utility against the expected effort [25]. The following empirical research will explore the drivers and barriers and their allocation on the respective level of adoption

¹ For a better understanding, see: 22. Licht, N.: Insights into Internet Voting: Adoption Stages, Drivers & Barriers, and the Possible Impact of COVID-19. Ragnar Nurkse Department of Innovation and Governance. Tallinn University of Technology (2021)

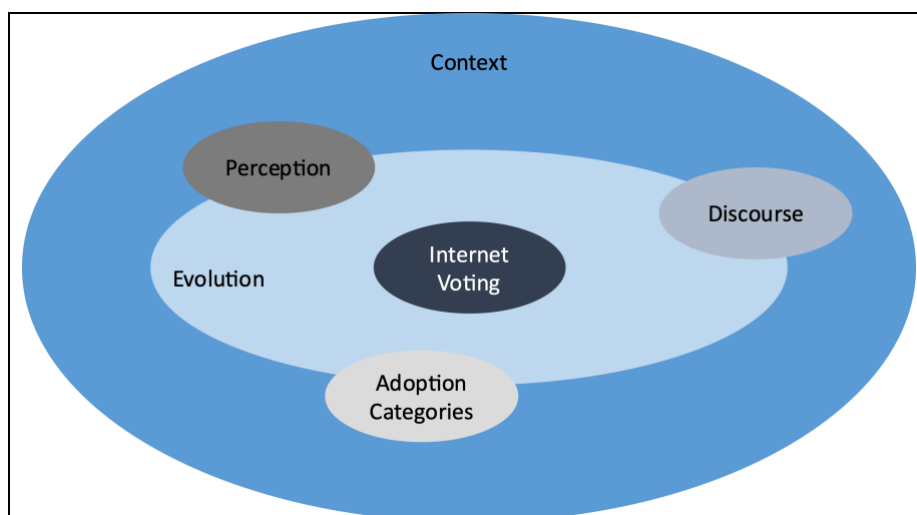


Fig. 1. Framework of Internet Voting

The data collection of this research was conducted via semi-structured expert interviews and complemented by desk research, allowing cross checking experts opinions with other sources. The study followed the framework provided by Krimmer's mirabilis [9] that aids to identify the respective stakeholders involved in the implementation process of e-voting technology. In the context of this research, it was limited to three stakeholders: i) Media/observer, ii) election management and iii) inventors or vendors of voting technology. More precisely, it was focused on practitioners/EMBs/policymakers, scholars and election observers, as well as vendors or inventors of i-voting technology. A total of 18 interviews were conducted, transcribed, confirmed and analyzed in NVivo, via a deductive codification approach proposed by Mayring [26]. Data triangulation is granted through confirming cross-checking answers against either statement of other interviewees or findings from the literature [27]².

This research has natural limitations with regard to its research design. Primarily, the finding of appropriate experts can limit the findings of the study to the extent that either not the most applicable experts might have been identified or that specific experts did not confirm to participate in the research [28]. In particular, it was more challenging to achieve an even distribution among gender and geographics. Also, during the interview process, issues may arise, mainly due to the lack of testing the human language, which may cause ambiguity and hence distort the originally intended meaning of words by the expert. Furthermore, qualitative research as such, as to their lack of generalizability as it would be the case in quantitative research [29].

² The empirical findings will be cited as in-text citations with the interview number in brackets, in the following format: e.g., single citation (1), multiple citations (1;2; 3...).

3 Analysis & Discussion of Drivers and Barriers

Given the dual nature of the process of adoption of technology we divide the information obtained from the expert interviews as well as the desk research in two main framing contexts, on the one hand, the context referring to the Political and Socioeconomic situation and, on the other hand, the Technological one.

3.1 Political and Socio-Economic Context Dimension

In line with the theory, the context is very influential in the establishment of election systems [30]. The findings further resemble the supporting framework and can be divided into social, economic, cultural/historical, political, organizational, legal and procedural elements.

Civil Society.

The different processes of construction of a society favor or disfavor the discussion, critique, and proposition of i-voting technology. A more diverse society consisting of academia, civil society organizations (CSO) and experts, enable a more varied discourse about i-voting and can be either driving or impeding diffusion. These groups are drivers if they, for example, promote the inclusion of excluded voter groups through i-voting or might be barriers if they voice security or transparency concerns. Furthermore, regions with a high number of IT-related content creation and the communication thereof, due to strong CSOs and expert groups, are somewhat reticent to adopting new voting technologies as they have stronger groups driving the discourse around the risks (5;8;10;13&14). However, the presence of solid lobby groups within society, fighting for the rights of visually impaired persons and expatriate voters, have been, on the other hand, identified as strong drivers for internet voting adoption on the political level (7;9;10;11&15).

Vendors.

Also, the lack of expert communities and hence a lack of expertise within society tends to make these contexts more susceptible to be targeted by vendors. High-level lobbying by vendors is very effective when no counterparties contribute to expertise to the debate (2;8). Technology in elections is considered because of the commercial implications and strong lobbying efforts by vendors that persuade governments to adopt new technologies in their elections (1;2). One of the interviewees (1) specifically mentioned the push of the commercial drive and its implications for voting technology adoption. Moreover, contexts with less regulated procurement methods, and the lack of civil opposition that is run by non-governmental actors, who are knowledgeable in that field, tend to faster purchase new voting technologies (NVTs) and in less sustainable way (1;2). Academia and expert groups have been identified as vital stakeholders in the adoption discussion due to their ability to aid in overcoming suspicions or doubts through investigating challenges, proposing solutions and creating prototypes (5;14).

Economic Situation.

Internet voting systems (IVS) and the respective infrastructure that is necessary to promote i-voting can be very costly in short-term consideration, not only in terms of purchasing but also maintenance of an IVS (4;6;16). From a long-term perspective, the associated costs per vote via IVS are remarkably lower than conventional votes and some cases have considered internet voting for the reason of cost reduction (1;4;11) [31, 32]. However, most cases that have introduced i-voting still provide traditional paper voting, i.e., postal voting, as an alternative option to prevent vote coercion, which in fact adds additional costs (2;6).

Culture and History.

Our findings suggest the existence of differences in the interpretation of vote secrecy and universal suffrage depending on the cultural context, which influences the perception of IVS (6). In more detail we observed that a relatively relaxed understanding of secrecy and a strong approach towards universal access might lead to enhanced i-voting efforts. On the contrary, where a particular emphasis on secrecy is present, further i-voting diffusion might be rejected if not enough proof is given via universal verifiability of how a vote is cast, counted and kept secret. Last, an increased emphasis on universal suffrage, and therefore, a strong focus on the inclusion of diaspora voters or visually impaired people might lead to higher IVS uptake (6;15).

Elections are, in some contexts, seen as a community-based exercise in which the electorate follows their duty to go and vote. That exercise might be perceived as an act of physically convening and voicing one's opinion and would culturally not accept to replace that with technology (5). This case does not describe the opposition of technology per se but the predominant proposition of tradition (3;6). Regarding historical influences, our interviews conclude that post-crisis situations or the newly gained independence of regions impact the creation of new voting systems (1). Often, the act of removing old election systems is an act of trust-building and demonstration of recent ruling in which NVTs are perceived as neutral third party that politicians and administrations have no influence over (1;3;5).

Political Context.

. In nearly all interviews, the political will was identified as both a powerful driver as well as a strong barrier. First, governments use i-voting technology as political agenda to demonstrate modernity and progress in their political activity (17). Some contexts have attributed electoral affairs to a ministry and restructuring the state alongside the electoral system is used for political campaigning purposes (2;18). In essence, political actors aim to appear progressive and modern and wish to use tools like IVS to prove also tech-savviness (18). Significant technological developments can be traced back to politically motivated events and decisions. If technology is perceived to be beneficial for the incumbent party, it is promoted; if not, the same party may become the greatest opponent to NVT development (1;2;3;5;10;15). This observation, also known as the "middleman paradox", refers to the phenomenon that incumbents

resist the move towards e-democracy because they perceive that the altered election system might lead to a decrease of their own political power and control [33]. In line with further evidence, change of government was named to be another influential factor. Two scenarios were identified which have been concrete barriers to IVS diffusions: 1) the election of a new governing party, also ascribable to the middleman paradox (6;14); and 2) a civil conflict in which the transformation of the election system is put on halt (2). Regarding the first scenario: If certain political actors identify that their electorate is opposing the idea of i-voting and that their competitor might benefit from online voting more than they expect to do, evidence shows that this actor tends to discontinue i-voting for purely political reasons [34] (6;11;14). Furthermore, the findings show that i-voting is a highly sensitive subject with attached political risks, associated costs and resources needed; therefore, unless a concrete need requires it, governments tend to refrain from touching that subject (4;6;11;14;15).

The second dimension refers to accessibility and universal suffrage, which have been identified to be among the strongest general drivers for i-voting adoption. Accessibility refers to the idea that “people with disabilities should be able to use all public spaces and services in the same way as other people” [35]. Online voting can enfranchise disabled people as they can more easily register and authenticate themselves and cast their vote from their home (3;7;9;10;15). The provision of universal suffrage identified by the OSCE [36] entails, further, the idea to integrate the entire electorate into the elections. Universal suffrage can be interpreted in different ways, and countries, as well as semi-autonomous regions, have been considering for a significant part to introduce i-voting because of their aspiration to include overseas or territorially challenged voters into their elections more efficiently. Nearly all conducted interviews mentioned the aspect of voting provision for the diaspora, overseas diplomats, consular staff, general populations in extreme territorial conditions or overseas soldiers. Essentially, the intrinsic motivation is political and only promoted if the incumbent expects to gain from including these groups of voters, as sometimes the diaspora consists of political opponents and hence its exclusion from electoral matters is deliberate (5;6;8;9;10;18). Another impact of diaspora voters concerns their foreign impact through campaign donations and exercising of their often-strong socioeconomic status and power on domestic political debate (2).

Organizational Context.

Another element to mention is, that as populations increase and administrative capacities need to be restructured to enable higher procedural efficiency, new technologies allow better election management and further ease electoral processes, especially regarding cumbersome remote voting processes such as postal voting (4;5;8;15;16). And yet, from the study, it is clear that voter coercion and vote-buying in remote and uncontrolled election environments still remain to endanger the integrity of elections, and for that, specific contexts that initially have seen technology as a practical solution refrain from particularly adopting i-voting (4). Also, the context’s set-up, procedural traditions and hurdles as well as the degree of digital governance and

the understanding of digital services play a substantial role in driving i-voting adoption due to the spill-over effect that tends to occur in digital ecosystems (2;7;9;14;17).

Legal Context.

The obtained results present evidence that legal frameworks need to be established for an effective i-voting introduction (14;16). Passing appropriate legislation, however, tends to be rather difficult because the law is rigid in nature, and ICT is relatively flexible and needs to be evaluated regularly. Law, once passed, will remain as a reference text for future considerations and cannot simply be changed on demand (14). Specific contexts experience the already written law to be a barrier, and lawmakers would need to pursue passing actively or amending the law, which allows for IVS considerations.

Furthermore, empirical data shows that law is subject to interpretation and that certain regions may therefore understand the legal text differently and hence court interpretations can be essential in the development of IVS (6;7;8). Cases were identified in which important court decisions prevented further NVT adoption and influenced third parties not to adopt (6;8), or judgements existed that paved the way for i-voting to be adopted (15). In the interviews, it was further identified that there is a lack of a general legal and technical framework/design that describes and defines the appropriated provisions of i-voting systems. This lack becomes a barrier because the standard according to which a potentially suitable system would be compared against does not exist, and hence the debate is less structured (9;10;11). The other scenario was described that a legal framework exists, but it is impossible to comply with the requirements, and it makes it merely impossible to proceed with i-voting development (9).

3.2 Technological Context Dimension

The following issue concerning technology and security features mainly concern the adoption process on the political level but is influenced by the narratives and discourses on the individual level. Although, during the interviews, it was mentioned that various technology designs exist, we generically refer to ‘the technology’ as such in order to enable a more holistic discussion. Besides the existing technological capabilities to host and conduct elections using i-voting, a threshold for many countries in terms of technology and security is the concrete definition of what technology should be used for the elections in form of a concrete framework (10;14). Furthermore, certain contexts lack respective experts that know how the systems work and that are able to provide the right guidance for it to be successfully implemented (11;13). Hence, a legal framework could also become a barrier, not just a facilitator for sustainable implementation. Legal frameworks can be worded in various ways, promoting or demoting the usage of remote online voting components (9).

Furthermore, technology is considered so complex that most citizens tend not to understand how the vote is being cast, counted, kept secret and how they can verify that their vote was counted as intended (3;16). Therefore, it is technically possible but often not viable to exchange a functioning system that is operating with paper (e.g. postal voting) with a new system that needs to provide transparency, secrecy and integrity proof to all stakeholders. Hence, the complex nature, in cases, is seen to be a barrier (1). It is, moreover, important to differentiate hereby between full-scale adoption and partial adoption. In contexts of partial adoption, technical failures and security breaches seem less concerning than if they were to occur in full-scale adoption contexts. Therefore, imposing the task of expanding with i-voting diffusion is a more complex endeavor than offering it for a share of the eligible electorate (2;15).

One of the biggest challenges from the technology side is to provide either individual or universal verifiability (1). The technical abilities exist to provide these features in a reliable way, but need to be acknowledged by the decision-making party in order to be fully useful (10). Although the demand for such verifiability feature to be present in the election system has increased, barely any state legislator has acknowledged and integrated such features into their requirements which can be both a barrier as well as a driver (14). On the one hand, it facilitates eased implementation efforts as they need to meet fewer requirements. On the other hand, the system is also more vulnerable to criticism of transparency and integrity.

Furthermore, internet voting does require not only the technology but also the infrastructure that would facilitate the execution of the election. Such infrastructure would be broadband networks with high penetration rates, especially in remote areas. If no internet access exists in remote areas, there is no utility gain from adopting IVS for the purpose of including remote areas better into elections (5;16;18). The mentioned issue is subject to the geographical context and is related to the digital divide, which is a term used to describe the gap between contexts that benefit from digital technology and those who do not [37]. The empirical findings suggest that the digital divide, which had been more so visible in the early 2000s, was a barrier to many non-Western contexts (4;16) [38-40].

Hence, these findings suggest that while none sufficient ICT infrastructure seemed to have been a barrier for IVS in non-Western contexts, the increase in broadband penetration with the beginning of the second decade drove IVS development to see the first advent of IVS cases in non-Western contexts [38]. Still, the digital divide remains to exist and further is a barrier to IVS development in certain regions (16;18) [41]. The following section analyzes and discusses the perception and discourse dimension.

3.3 Perception and Discourse Dimension

One of the major findings from the interviews in terms of perception is regarding the issue of trust. Although trust is hard to measure and still subject to ongoing academic investigations, certain parameters could have been identified. The public perception is

mostly referring to the drivers and barriers that impact the diffusion that occurs on the individual level after the political decision has been made to introduce IVS in society.

The findings support the assumption that election systems are as much trustworthy as the people who erected and proposed them. Hence, if people mistrust the government and or EMBs who implement IVS, they tend to mistrust the technology (5). Furthermore, regardless of the previous trust given to one election system, it is not granted that this trust is simply transferable to any novel election system. On the contrary, it seems that strong trust in EMBs in primarily Western democracies might be one of the bigger barriers to i-voting adoption as the primary assumption is to question whether new technology is necessary and simultaneously to endanger a well working system (1;10;14). This may be further supported by the concept of path dependency, which states that individuals would decide to trust and use a system based on previous experiences, decisions and preferences that they made [42, 43]. That phenomenon exists along with all fields of social spheres and might certainly affect the choice of usage of election systems.

Internet voting technology requires a great amount of trust from the electorate since its technological setup is relatively complex, and very few experts do understand the system entirely (1). Whether one may trust in one particular aspect or not is rather incoherent with objective measurements. Regardless of objectively measured and reliable evidence that would suggest that appropriate i-voting technology exists, many cases experience one of the biggest barriers to be the lack of trust (1;3;5;11;14). Additionally, objectivity and trust tend to be fragmented by public discourse and the strong presence of social media that influences public opinion on electoral matters [44]. Moreover, specific expert groups and CSOs have made it their duty to detect and inform about vulnerabilities in i-voting systems particularly, since the 2016's US presidential election, increased interest in cybersecurity around elections (6;7;9;18). Although public discourse has been identified to be a barrier in many instances, there are also cases in which pressure by CSOs and media on politicians have paved the way for the introduction of IVS (15).

Although certain risks had been already present in the early 2000s and cyber hacking and lobbyism against the introduction of i-voting existed since the first hour (10), it was, however, on a much smaller scale. In comparison to nowadays, there was less awareness of the entirety of cyber-risks and also less internet usage penetration in general (6) which can nowadays be seen as a barrier to further diffusion. The perception of technology its potentials and risks has shifted. Common cyber threats and dangers have been put more in focus around the discussion for i-voting introduction than it was the case in the early 2000s. That is mostly due to the fact that the technology was relatively novel and less experimented with than it is nowadays. Hence, more threat and risk awareness exist as common knowledge in the electorate, and hence success stories back then might not be as successful today (6;7).

Since i-voting technology is to a degree somewhat intangible for the large share of people, i-voting demonstrations are used to build trust in the system (1;10;14). Including rhetoric and competence demonstration seem to be useful in convincing the electorate about the system, as suggested by the findings. These demonstrations can be of bureaucratic nature, in which the focus is rather on the institutions and has been proven to be successful in contexts in which a history of malfunctioning of institutions exists. In a context in which previously technical failures in election systems had occurred, trust-building via technology demonstrations have proven to be successful (14). Perception, then, may be impacted by security breaches and technical failures. The identified cases in which that occurred show different results for the degree of usage (6;7;14). Hereby, a necessary differentiation has to be made between the roles that academia or CSOs play and the media. These stewards of discourse certainly have identified to be impacting the diffusion process and certainly media on the individual diffusion level. However, more data is needed to look into the issue impact of trust in election systems as a result of technical failures.

From the empirical findings, we identified the drivers for the political decision level, to be universal access and accessibility for disabled voters, the pursuit of a contactless democracy, they wish to appear modern, the vendor's push, the process improvements, the perception of technology to be a neutral third party, the perception of increased administrative integrity, cost reductions, strong lobby groups, expected increase in voter turnouts and the presence of high socioeconomic power and well-established technical infrastructure. On the individual adoption level, we presented evidence that drivers exist such as convenience voting, spill-over effects within a digital society and the socioeconomic status of voters. Following barriers were identified for the political level adoption process: the middleman paradox, political crisis, change of government, security concerns, theoretical technical vulnerabilities, strong opposition from CSOs and academia, lack of a framework, lack of technological infrastructure, lack of verifiability, procedural barriers and the change of legal requirements. Barriers to adoption on the individual level have been identified as path dependency, cultural traditions, mistrust in technology and mistrust in EMBs and governments.

Table 1. Overview of the Drivers of Internet Voting

Drivers
Political level
Universal access (Expatriate & overseas staff voting, voting in territorially challenging locations)
Accessibility
The political will to appear modern and innovative
Contactless democracy
Vendor's commercial drive
Increase turnout/prevent further decline

Strong lobby groups
Perception of technology as neutral third party
Cost reductions
Process improvements
Integrity improvements in administrative operations
Socioeconomic status and high technological infrastructure (geographics)
Individual level
Convenience voting
Spill-over effect within already digitised societies and their ecosystem
Socioeconomic status of the voter

Table 2. Overview on the Barriers of Internet Voting

Barriers
Political level
Middleman Paradox
Political crisis
Change of government (related to middleman paradox)
Security concerns
Theoretical technical vulnerabilities
Strong opposition from academia & CSOs
Lack of a framework
Lack of technological infrastructure/Digital divide
Lack of verifiability
Procedural barriers
Change of legal requirements
Individual level
Path dependency
Cultural traditions
Mistrust in technology
Mistrust in government and EMBs

4 Conclusion

In order to answer the question on what drivers and barriers exist that prevent further internet voting diffusion, subsequently, the discussion occurs first on the political level and then on the individual level.

The driving or lobbying stakeholders on the political decision level, are the diaspora, territorially challenged voters or disabled voters which resemble the described relevant social groups. Further the groups lobbying for these relevant social groups on the political level and hence driving stakeholders as for example lobby groups, academia, CSOs or vendors have a resemblance to the change agents and opinion leaders identified in the conceptual model. Further findings suggest that the political will is a major driver for i-voting adoption on the political level as to prevent decreasing voter turnouts or the urgency to provide an appropriate election system for the context of an evolving contactless democracy or to appear modern through the introduction of NVTs. Last, the degree of the socioeconomic status, influences whether the political level even considers the move towards NVTs to be feasible or not.

On the individual adoption level, although, the aspect of convenience voting is still under further academic investigation, the empirical findings suggest that the proposed theory of relative utility in regard to effort can be confirmed for the individual level. Furthermore, the findings have also identified that, although an early interest might exist for i-voting, individuals tend to not maintain that interest if they experience no further usage of the infrastructure than for merely voting online from time to time. In the case of Estonia, this steady interest was achieved through the wider usage avenues of the e-ID for bank transactions for example [45]. In contrast, the Austrian case failed to mobilize enough supporters for its online voting systems because it had no further utility to its voters than to vote [4]. Ergo, a wider-context deployment of ICT technology and the practicality of a digital ecosystem might create a spill-over effect and hence drive i-voting technology for the technology acceptance on the individual level.

From the finding, a central part that impedes further global i-voting adoption has been the middleman paradox. This is a central barrier for many regions as the first adoption decision is made on the political level and later transferred to the individual level. However, the fear of losing one's own power that could only be bypassed if an urgent need for the election reform would appear, impedes further i-voting in many contexts around the world. Further contextual barriers were identified to be security concerns, lack of verifiability and theoretical vulnerabilities. Moreover, mistrust and in combination with public discourse are opposing forces to the development of NVTs as CSOs, academia and expert groups in many cases actively oppose the idea of i-voting implementation due to security and verifiability concerns. Their ability to provide expertise, facilitate communication, to have access to prototypes and further resources such as data and expert knowledge makes them to effective change agents and opinion leaders that frequently lobby against IVS diffusion.

A particular barriers to adoption on the individual level has identified to be path dependency [43, 46]. It being a purely social issue, cultural norms and values amplify the problem of path dependency and confirm the cultural explanation for why technology is adopted. The social construction of society and perception of technology

are decisive in explaining adoption and would be confirmed by the issue of path-dependency. Mistrust in technology is strongly depending on perception and consists of the fear that the technology might not be secure, which mostly is related to the fact that the technology is too complex for the average person to understand fully. Furthermore, the mistrust might also exist towards the decision-makers generally, and therefore the technology might not be accepted.

In conclusion, the research question can be answered through the depicted evidence showing that in total, 15 drivers, 12 on the political and three on the individual level and 15 barriers, with 11 on the political and four on the individual level, have been identified. Strong driving and impeding forces alike were found on the political level to be the absence or presence of political will, necessity and the so-called middleman paradox. Even if the list of drivers and barriers is balanced, the reality shows that the implication of them is not following the same pattern, since the reduced number of adopters of i-voting brings to the conclusion that barriers play a more important role in the process of adoption than drivers. Further detailed case studies in selected countries could shed new light on how these drivers and barriers interact in particular administrative and political contexts and bring to the final decision of implementing or not i-voting. Additional research would be necessary in the field of trust in elections and specifically in election technology as well as the respective roles attributed to building or harming trust through the two discourse drivers that are academia or CSOs and on the other side the media. From the interviews it became apparent that these groups another study is merited but in which their roles especially in the individual diffusion process is further investigated. Possible questions to consider could be how can trust be measured and how can trust-building of new voting technologies be formed and what roles do media and academia play in that process? Last, in order to understand how various contexts, deal with electoral crises and why certain regions stopped their internet voting, while others remain to deploy IVS in their elections, a follow-up study on Estonia's foreign cyber interference, France's discontinuation in 2017 and Norway's case of their technical vulnerabilities may be appropriate. In this proposed study, it would be sensible to look at the positioning of academia and CSOs and the reasons why that may be the case and under what circumstances that might change and impact the adoption and diffusion of internet voting. In summary, internet voting has been around for more than two decades and identified to be a logical tool for democracy and yet lacks large-scale adoption. In this paper we analyzed and presented general drivers and barriers that impact the adoption and diffusion process and illustrated further research areas that merit further investigation. Internet voting, being a process in a political process is also highly impacted by political factors itself and therefore significant qualitative differences between the respective drivers and barriers for the respective contexts might exist.

Acknowledgements

This work received support from mGov4EU and eceps grants - 857622 and 959072 and, in the case of Dr. David Duenas-Cid, also from Polish National Research Center

grant (Miniatura 3 - 2019/03/X/HS6/01688 “Zaufanie do technologii w e-administracji: Powtórna analiza nieudanego wdrożenia elektronicznych maszyn do głosowania w Holandii (2006-07)”

References

1. Fuller, R.: Buckminster. No More Secondhand God and Other Writings. Carbondale and Edwardsville: Southern Illinois University Press (1963)
2. Gates, B., Myhrvold, N., Rinearson, P., Domonkos, D.: The road ahead. (1995)
3. Cook, T.: Apple’s C.E.O. Is Making Very Different Choices From Mark Zuckerberg. In: Swisher, K. (ed.). New York Times (2021)
4. Krimmer, R.: Internet Voting in Austria: History, Development, and Building Blocks for the Future. WU Vienna University of Economics and Business (2017)
5. IDEA, <https://www.idea.int/news-media/multimedia-reports/global-overview-covid-19-impact-elections>
6. Krimmer, R., Duenas-Cid, D., Krivonosova, I.: Debate: safeguarding democracy during pandemics. Social distancing, postal, or internet voting—the good, the bad or the ugly? *Public Money & Management* 41, 8-10 (2021)
7. Kersting, N., Baldersheim, H.: Electronic voting and democracy: a comparative analysis. Springer (2004)
8. Schaupp, L.C., Carter, L.: E-voting: from apathy to adoption. *Journal of Enterprise Information Management* (2005)
9. Krimmer, R.: The evolution of e-voting: why voting technology is used and how it affects democracy. Tallinn University of Technology Doctoral Theses Series I: Social Sciences 19, (2012)
10. Vegas, C., Barrat, J.: Overview of Current State of E-Voting Worldwide. *Real-World Electronic Voting: Design, Analysis and Deployment* 51 (2016)
11. Risnanto, S., Abd Rahim, Y.B., Herman, N.S., Abdurrohman: E-Voting Readiness Mapping for General Election Implementation. *Journal of Theoretical and Applied Information Technology* 98, (2020)
12. Gibson, J.P., Krimmer, R., Teague, V., Pomares, J.: A review of e-voting: the past, present and future. *Annals of Telecommunications* 71, 279-286 (2016)
13. Krimmer, R., Triessnig, S., Volkamer, M.: The development of remote e-voting around the world: A review of roads and directions. In: *International Conference on E-Voting and Identity*, pp. 1-15. Springer, (Year)
14. ACE - The Electoral Knowledge Network, http://aceproject.org/ace-en/focus/e-voting/countries/mobile_browsing/onePag
15. Khutkyy, D.: Policy paper Internet Voting: Challenges and Solutions. European Digital Development Alliance (2020)
16. Górny, M.: I-voting—opportunities and threats. Conditions for the effective implementation of Internet voting on the example of Switzerland and Estonia. *Przegląd Politologiczny* 133-146 (2021)

17. Vassil, K., Solvak, M., Vinkel, P., Trechsel, A.H., Alvarez, R.M.: The diffusion of internet voting. Usage patterns of internet voting in Estonia between 2005 and 2015. *Government Information Quarterly* 33, 453-459 (2016)
18. Vinkel, P., Krimmer, R.: *The How and Why to Internet Voting an Attempt to Explain E-Stonia* (2017)
19. Anonymous: Anonymous. In: *International Joint Conference on Electronic Voting*. Springer, (Year)
20. Brown, M., Hale, K.: *Applied research methods in public and nonprofit organizations*. John Wiley & Sons (2014)
21. Ormston, R., Spencer, L., Barnard, M., Snape, D.: The foundations of qualitative research. *Qualitative research practice: A guide for social science students and researchers* 2, 52-55 (2014)
22. Licht, N.: *Insights into Internet Voting: Adoption Stages, Drivers & Barriers, and the Possible Impact of COVID-19*. Ragnar Nurkse Department of Innovation and Governance. Tallinn University of Technology (2021)
23. Pinch, T.J., Bijker, W.E.: The social construction of facts and artefacts: Or how the sociology of science and the sociology of technology might benefit each other. *Social studies of science* 14, 399-441 (1984)
24. Rogers, E.: *Diffusion of Innovation* Fifth edition New York. NY: Free Press (2003)
25. Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D.: User acceptance of information technology: Toward a unified view. *MIS quarterly* 425-478 (2003)
26. Mayring, P.: *Qualitative content analysis: theoretical foundation, basic procedures and software solution*. (2014)
27. Flick, U.: *Triangulation - Eine Einführung* VS Verlag. (2008)
28. Flick, U.: *An Introduction to Qualitative Research*. SAGE, Hamburg (2014)
29. Ochieng, P.: An analysis of the strengths and limitation of qualitative and quantitative research paradigms. *Problems of Education in the 21st Century* 13, 13 (2009)
30. Derichs, C., Heberer, T.: *Wahlssysteme und Wahltypen: politische Systeme und regionale Kontexte im Vergleich*. Springer (2007)
31. Krimmer, R., Duenas-Cid, D., Krivonosova, I.: New methodology for calculating cost-efficiency of different ways of voting: is internet voting cheaper? *Public Money & Management* 41, 17-26 (2021)
32. Krimmer, R., Duenas-Cid, D., Krivonosova, I., Vinkel, P., Koitmaa, A.: How much does an e-Vote cost? Cost comparison per vote in multichannel elections in Estonia. In: *International Joint Conference on Electronic Voting*, pp. 117-131. Springer, (Year)
33. Mahrer, H., Krimmer, R.: Towards the enhancement of e-democracy: identifying the notion of the 'middleman paradox'. *Information systems journal* 15, 27-42 (2005)
34. Postimees. Tallinn, <http://s3-eu-west-1.amazonaws.com/pdf.station.ee/epl/epl/2020/11/17/p2.pdf?AWSAccessKeyId=AKIAJMCAZLEYS3TMYI7Q&Expires=1619606705&Signature=5yLZkaC0Jyt2IOGQ6E2NrzA7oS8%3D>

35. OSCE/ODIHR: A Booklet about: Watching Elections and Helping People with Disabilities take part in Elections. OSCE/ODIHR. Warsaw (2017)
36. OSCE: Document of the Copenhagen Meeting of the Conference on the Human Dimension of CSCE. In: OSCE (ed.). OSCE.Copenhagen (1990)
37. Hilbert, M.: The end justifies the definition: The manifold outlooks on the digital divide and their practical usefulness for policy-making. *Telecommunications Policy* 35, 715-736 (2011)
38. Ronquillo, C., Currie, L.: The digital divide: Trends in global mobile and broadband Internet access from 2000-2010. *Nursing informatics ... : proceedings of the ... International Congress on Nursing Informatics 2012*, 346 (2012)
39. UNCTAD: The Digital Divide Report: ICT Diffusion Index 2005. United Nations (2005)
40. Norris, P.: *Digital divide: Civic engagement, information poverty, and the Internet worldwide*. Cambridge university press (2001)
41. <https://cipesa.org/2021/03/south-africas-parliament-rejects-plan-to-introduce-e-voting/>
42. Investopedia, <https://www.investopedia.com/terms/p/path-dependency.asp>
43. David, P.A.: Clio and the Economics of QWERTY. *The American economic review* 75, 332-337 (1985)
44. Krimmer, R., Rabitsch, A., Kuzel, R.o., Achler, M., Licht, N.: *Elections & Internet, Social Media and Artificial Intelligence (AI): A Guide for Electoral Practitioners*. UNESCO (2021 forthcoming)
45. Martens, T.: Electronic identity management in Estonia between market and state governance. *Identity in the Information Society* 3, 213-233 (2010)
46. Gross, R., Hanna, R.: Path dependency in provision of domestic heating. *Nature Energy* 4, 358-364 (2019)

Turnout in e-voting pilots in the 2021 presidential elections in Ecuador

Régis Dandoy¹[0000-0003-3593-0024]

¹ Universidad San Francisco de Quito, Ecuador
rdandoy@usfq.edu.ec

Abstract. Based on a quasi-experimental design of the 2021 Ecuadorian presidential elections, I investigate the effect of postal voting, on-site electronic voting (DRE voting) and Internet voting on non-resident citizens' effective voter turnout. This short paper shows that, while DRE voting has no significant impact on turnout, turnout among non-resident citizens using Internet voting and postal voting is significantly higher compared to neighboring electoral districts.

Keywords: E-voting, Internet voting, Turnout, Ecuador

1 Introduction

In preparation of the 2021 elections in Ecuador, the electoral management body decided to run pilots testing different voting modalities for Ecuadorian voters living abroad. In these pilots, three different voting modalities were tested in three overseas electoral districts: postal voting (Ottawa district), on-site electronic voting / DRE voting (Buenos Aires district) and Internet voting (Phoenix district). In all other overseas electoral districts, non-resident voters cast their votes in-person in consulates and other diplomatic venues. In the form of a quasi-natural experiment, these pilots constituted a unique opportunity to investigate the impact of different voting modalities on the decision of some voters to participate in elections. This paper explores the impact of postal voting, DRE voting and Internet voting on turnout for voters living abroad in an overlooked country (Ecuador) at the occasion of the two rounds of the 2021 presidential elections and to contribute to the burgeoning literature on the consequences of different e-voting modalities for voter's behavior. In particular, recent scholarly works have empirically investigated turnout among non-resident voters using Internet voting, yet mostly in Western countries [1-3].

2 E-voting and effective turnout in Ecuador

Compared to traditional measures of turnout, my operationalization of effective turnout takes into account the invalid votes (i.e., the blank and null votes) in the calculation of turnout. It is measured as the number of valid votes divided by the total number of registered voters. The analysis of effective turnout is particularly relevant

in the case of Ecuador as DRE and Internet voting systems provide the voter with two additional options: two buttons are available on the screen – usually at the bottom of the list of parties or candidates – allowing the voter to express a blank or a null vote instead of a vote for a specific party or candidate. Moreover, the share of null and/or blank vote is traditionally high in Ecuadorian elections, usually around 10% for the presidential elections.

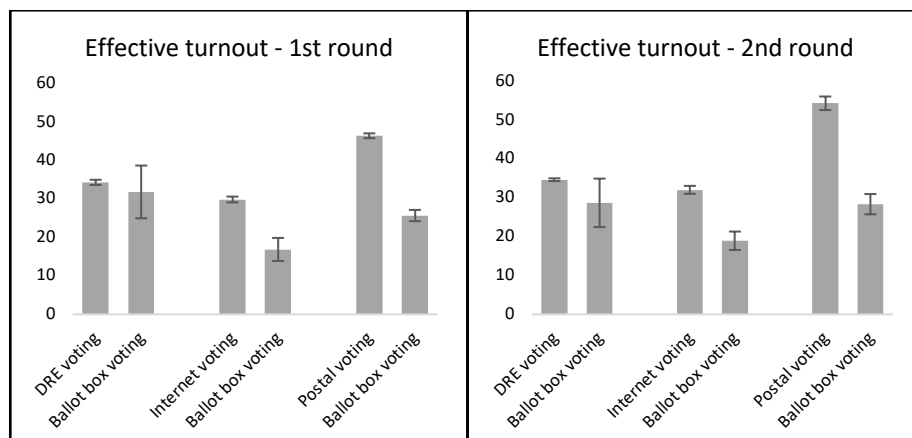
Based on the analysis of turnout figures for the two rounds of presidential elections in Ecuador (7 February and 11 April 2021), I compare the measure of effective turnout across four different types of ballots: paper ballots emitted in-person in the consulate building (i.e., ballot box votes), postal paper ballots (i.e., postal votes), electronic ballots filled in-person in the consulate building (i.e., DRE votes) and internet-based ballots (i.e., Internet votes). I include in the analyses the three electoral districts where pilots have been running by the Ecuadorian electoral management body and the two geographically closest electoral districts using ballot box voting. For instance, the Phoenix district (Internet voting) is compared with the Houston and Los Angeles districts (ballot box voting). This geographical proximity allows to limit the socio-demographic variation of the characteristics of the electorate in various abroad districts and to control for the potentially differentiated impact of the Covid-19 situation on voter mobilization. Data originates from official elections results [4] and figures are based on average turnout in both gender groups.

I observe that the introduction of alternative modalities of voting affected turnout figures compared to neighboring districts (see Fig. 1 and 2). In the electoral district where DRE voting was used, the average effective turnout for the presidential elections is slightly higher than in neighboring districts using ballot box voting. Differences are within the margin of standard errors, meaning that there are no significant differences in turnout between districts using DRE voting and districts using ballot box voting. This finding is in line with global South scholars that found no statistically significant effect of DRE voting on turnout in Brazil and in India [5-6].

Contrary to DRE voting, the impact of Internet voting on turnout is significantly more important and positive. On average, effective turnout is higher by 13% in the electoral district using Internet voting compared to neighboring districts using ballot box voting. It therefore seems that Internet voting has a significant and positive impact on turnout in this electoral district located abroad, confirming previous studies on non-resident voters in the USA and Switzerland [1-2]. Finally, postal voting has a similar positive impact on turnout as the electoral district implementing this voting modality displays an effective turnout more than 20% superior to turnout in neighboring districts based on ballot box voting.

The Covid-19 pandemic placed elections in many countries between a rock and a hard place, and election officials worldwide have been considering the implementation of alternative methods of voting. By demonstrating that turnout among Ecuadorian non-residents using remote voting (i.e., postal and Internet voting) is significantly higher than turnout among those casting their votes in the polling stations, this short paper provides an argument for election officials that envisage additional voting modalities for non-resident voters in future elections.

Fig. 1 and 2. Average effective turnout per voting modality in the 2021 elections in Ecuador



References

1. Fowler, A.: Promises and perils of mobile voting. *Election Law Journal* 19(3), 418–431(2020).
2. Germann, M.: Internet voting increases expatriate voter turnout. *Government Information Quarterly* (published online) (2021).
3. Dandoy, R., Kernalegenn, T.: Internet Voting from Abroad: Exploring turnout in the 2014 French consular elections, *French Politics* (published online) (2021).
4. National Electoral Council (CNE) webpage, <https://app01.cne.gob.ec/Resultados2021>.
5. Fujiwara, T.: Voting Technology, Political Responsiveness, and Infant Health: Evidence from Brazil. *Econometrica* 83(2), 423–464 (2015).
6. Desai, Z., Lee, A.: Technology and protest: the political effects of electronic voting in India, *Political Science Research and Methods* 9(2), 398–413 (2021).

On the Offensive: IT Threats

Penetration Testing a US Election Blockchain Prototype*

Shawn M. Emery^{1,2}, C. Edward Chow^{1,3}, and Richard White^{1,4}

¹ University of Colorado, Colorado Springs USA

² semery@uccs.edu

³ cchow@uccs.edu

⁴ rwhite2@uccs.edu

Abstract. With electronic voting (e-voting) systems under increased cyber-attack by malicious agents, it is critical that the security of these systems be thoroughly evaluated. This article describes techniques used to comprehensively analyze a prototype mobile voting system utilizing blockchain technology. For identified vulnerabilities, an attack method is described in order to exploit these issues and suggestions are made in order to help resolve the security implications of the attack. This analysis considers multiple layers of the network stack, including the voting application suite of software, as attack vectors. From this, the lessons learned can be used to improve future electronic voting systems by identifying the various attack surfaces regardless if they were successfully exploited or not. This in itself will help add to specific domain knowledge of attacking e-voting systems to utilize blockchain technology.

Keywords: electronic voting · blockchain voting · penetration testing

1 Introduction

Due to recent revelations about electronic voting system attacks by Russia [6] and, ironically, in light of Russia's own blockchain voting system vulnerabilities [3], there is an urgent need to understand the security posture of existing and future voting systems. This paper examines a US election system prototype that utilizes blockchain technology. A future voting system may or may not be based on this prototype, but by thoroughly testing the security posture of such a system we will have a better understanding of the risks exposed or secured by this design. Democratic societies depend upon the integrity of free and fair elections. Cyber-attacks on voting systems undermine confidence in electoral results and present a serious concern to representative democracies.

1.1 Background

A US government organization, that has requested to remain unnamed, plays an important role in national elections. In order to improve voter experience and

* This work was supported in part by Colorado State Bill 18-086.

S. Emery et al.

ease election administration, this organization developed a mobile application compatible with both Apple and Android operating systems. The organization approached the Colorado National Cybersecurity Center (NCC) about field testing this R&D concept. The NCC, in turn, introduced this organization to the University of Colorado, Colorado Springs (UCCS), who agreed to conduct a field test in the form of a mock election. In preparation for the mock election, the team at UCCS offered to replicate the prototype architecture and evaluate its deployability, and also conduct both active and passive testing to evaluate its security.

Note that the system tested was a conceptual design, to help prove that a digital system could meet the requirements of a voting system. As designed it was never intended to go into a production electoral system at any level; local, state, or federal, let alone the US 2020 presidential election.

There have been penetration tests performed and the results published on other internet based e-voting prototypes [14], but the authors are unaware of any other prototypes that utilized blockchain infrastructure in the US. There have been many warnings against using blockchain in e-voting systems dating back a few years, including recent analysis [8].

1.2 Cases against blockchain utilization

The architecture of this prototype is predicated on blockchain technology in order to register and verify elections, register voters, and record submitted ballots. Initially, blockchain capabilities seemed to intersect with a number of properties critical to a secure voting system, however, a report from The National Academies of Sciences, Engineering, and Medicine (NASEM) itemized a number of weaknesses in employing blockchain technologies for voting systems [7]. The authors state that the blockchain could utilize other subsystems that could alleviate some of these disadvantages but that these subsystems could be used without the blockchain to begin with. The purpose of this paper is to comprehensively explore the consequences of deploying blockchain systems and their dependencies, which introduces additional attack surfaces for malicious actors to drop some vote transactions and favor others.

1.3 Testing methodology

At the request of the organization, and with help from their contractor, UCCS replicated the prototype architecture to evaluate its deployability, and also conducted active and passive testing to evaluate its security. Due to certain proprietary constraints, testing was necessarily limited to those components that were made available to UCCS. So for instance, UCCS was unable to conduct a white hat inspection of the source code due to intellectual property rights. For similar reasons, penetration testing was also limited, and the team was unable to perform a complete in-to-outside evaluation. Still, the team was able to perform a fairly comprehensive outside-to-in evaluation by attacking the blockchain smart

contract bytecode, network stack, application dependent libraries, operating system, containers, system/application configuration, and voter privacy. Our team was only allowed to test one instance of the implementation.

2 Voting System Architecture

The voting system targeted in the attack consisted of a mobile application (app) for Apple iOS and Android, a web server for the prototype app and EO APIs, and middleware servers between the application level APIs and the blockchain related servers (middleware servers), which coupled the back-end functionality of the system: Key Management Service, Identity Management Service, Database Servers, and Blockchain Nodes.

The prototype is a scalable architecture designed for independent operation at state and local levels with support from the organization. In deployed configuration, the mobile application, web server, middleware servers, and key management infrastructure are controlled by the organization, whereas the identity management system, database servers, and blockchain nodes are under the control of state and local EOs.

Prototype testing was planned to culminate in a large-scale mock election. In preparation for the mock election, UCCS was responsible for instantiating and configuring all components that would fall under the control of state and local EOs, with the exception of the Key Management Interoperability Protocol (KMIP) and identity management services. The organization retained control and responsibility over its components, plus the simulated state identity management service system. UCCS used Linux cloud compute resources for all of the deployed systems under their control. This allowed for relatively inexpensive deployment of the prototype architecture, and had the further advantage of allowing UCCS to clone these systems for authorized independent testing when the organization needed exclusive access to the primary system.

The functional flow of a prototype election begins when an EO sets up an election database and sends notices to voters via mail inviting them to download the app. Following the instructions received in their mailed invitation, prospective voters can then download the app through conventional means onto their iOS or Android device. Voters then use the app to sign-up with an existing account to an election by either scanning an activation QR code or manually entering the code that's on their invitation. Voters then log into the mobile app to cast their electronic ballot using conventional navigating and selection techniques. Voters then affix their signature by signing on the device touch-screen before submitting their ballot to be counted. An EO verifies the ballot signature before consigning it to be registered in the blockchain. Only after it is verified in the blockchain will the vote be tabulated.

The operational flow of a prototype election begins when the designated EO is added to the election database and blockchain. The EO then authenticates through the secure voting web site. The EO can subsequently instantiate a new election and voter registry by submitting an election spreadsheet through the

S. Emery et al.

web interface. The spreadsheet contains districts, precincts, parties, the various contests, ballot types, ballot types to precinct mappings, and voter activation codes. The middleware server processes this request and generates the corresponding database requests. The middleware server deploys the associated voter registry, election registry, election, and election verification smart contracts. The middleware server then adds the election and voter registry as an election instance by constructing and submitting a blockchain transaction for the election instance. When this is completed, a ballot activation code is sent to the voter via mail. This code was created when the election was instantiated. The activation code is correlated to the voter's registration ID number in the database and an account is created for the voter on the blockchain. Once the voter authenticates through the application, by supplying a voter supplied PIN and verification code, then the voter is allowed to select the contests and sign the ballot before submitting the completed ballot. With ballot submission the database records the meta data; election ID, affidavit, submission date, status, election contract address, and voter registration ID. The voter transaction is then recorded on the blockchain as having voted. The ballot itself is then recorded on the blockchain from the voter's account. Through the web interface the EO looks for the submitted ballots and either approves or rejects submitted ballots based on affidavit, etc. If approved then the voted ballot's metadata is marked as approved in the database and a transaction is sent to the blockchain indicating that the ballot submission by the voter has been approved. The EO can also call election verification smart contract functions to audit the specified election. The essential components of the system are detailed in the following sections.

2.1 Mobile Application

The prototype app allows the user to register as a voter in the election. In a real election, mail is sent to the voter for account setup for the targeted election. However, in preparation for the mock election, accounts were already provisioned for each e-mail address for the participants of the mock election. For the mock election, e-mails were sent containing activation codes (QR codes) for activating their accounts for an election whereas in a real election, a separate mail would be sent to the voter with a QR code for the targeted election. The voter then scans the QR access code in order to register for the election. The user then provides their e-mail address, mobile phone number, and creates a five digit PIN. Once a user is registered for an election the user is allowed to sign-in by providing their mobile phone number and the PIN that they created. A verification code is sent, which allows the user to sign-in. Once signed in, the user is allowed to select the ballot and vote for the corresponding candidates and propositions for that election. Before submitting the ballot the voter is required to draw their signature on the touch-screen.

2.2 Web Server

The web servers provide the user interface for both the voter registration and voter ballot submission, as well as the EO responsible for creating, registering, and verifying elections and submitted ballots. This functionality is provided by the middleware server on the back-end. In the test configuration the web server was not login accessible for white hat testing during this project. Network connections were secured with HTTPS.

2.3 Middleware Servers

The middleware servers are a crucial system that correlates registered elections, registered voters, and submitted ballots to the various back-end software subsystems as previously described. The middleware servers were not login accessible for white hat testing during this project. The middleware servers were configured for load balancing.

2.4 Database Servers

The database servers, running MongoDB (v4.0.10), store each of the registered election templates, voters, voters registered, and submitted voter ballots. For testing, three database servers were deployed on Ubuntu cloud compute resources. A replica set was configured with one primary server and two secondary servers. Network connections were secured using TLS and encryption-at-rest was configured for the collection with the master key residing on a KMIP server.

2.5 Blockchain Nodes

The middleware servers validate and send voter and EO transactions to the blockchain nodes. The blockchain nodes were running Parity-Ethereum (v2.6.0), which were instantiated from Docker images. The wallets are stored on the middleware servers in order to sign and submit transactions on the blockchain. For testing, five blockchain servers were deployed on Ubuntu cloud compute resources. The configuration employed simulated the decentralization advantages of a blockchain system, lending to the unique environment of local, state, and federal governments in this problem space. The consensus protocol configured was Proof of Authority (PoA) with Authority Rounds (Aura) for the private chain. The step duration was set to four seconds. Network connections were secured with the respective node's provisioned secret key.

3 Attacking the System and Application Configuration

While deploying the prototype architecture, UCCS encountered a number of deployment instructions that introduced vulnerabilities exposed by penetration testing: file permissions/ownership, disclosing passwords/keys through command arguments, and minimal firewall rule-sets.

S. Emery et al.

3.1 Permissions/ownership on sensitive files

On a number of Unix based file systems the default `umask(2)` is `0022`, which means that a file created by the user will be readable/writable by user, and readable by group and other, assuming ACLs are not involved. With blockchain infrastructure there are several sensitive files to be aware of:

- The node-key and account password can be stored in the blockchain node configuration file
- The account password can be stored in a separate file on the blockchain node
- Files that contain passwords used to decrypt certificate files (with key pairs)

Administrators should ensure that these files are only readable by a user that has the least amount of privileges for the required access that it needs of the system. During penetration testing we found that the voting system certificate file and some configuration files were initially readable by "other" as some deployment instructions did not consider default file permissions and ownership.

The files that were inadvertently left open would allow both passive and active attacks on the MongoDB and blockchain nodes. Eavesdropping of voting data could be possible. The worst case scenario could entail a man-in-the-middle (MITM) attack that would modify voter data information to alter the outcome of the election. Even with forward secrecy (FS), if the attacker can manipulate the session key then the cipher suite, e.g. Ephemeral Diffie-Helman (DHE), is still vulnerable. Guidance on hardening server systems is discussed in [10].

3.2 Disclosing sensitive information to unprivileged users

Another common mistake we found was providing passwords, secret keys, etc. as process arguments. This will allow unprivileged users to view sensitive information by executing `ps(1)` or searching the system's `proc(5)` structure. Well designed software will allow a configuration file version of utilizing this argument. Keeping sensitive information in the corresponding file readable only by the user of the process will prevent further exploitation of the system. During penetration testing of the voting system, node-key material was exposed by starting the blockchain virtual machine with this argument.

With the node-key an attacker can impersonate the node in order to create a denial of service (DoS) at the least. If an unprivileged attacker can access the majority of the nodes' `proc` structure then the attacker would be able to launch a 51% attack on the consensus nodes. If a 51% attack is successful then the attacker could allow transactions that vote for the candidate of their favor and drop all other votes.

3.3 Minimal set of firewall rules

It is imperative that the firewall rule-set be as restrictive as possible between peers and other dependent subsystems. A better way to do this is to start with essential ports, e.g. SSH (in order to log into the system), and add rule-sets as

dependencies are discovered through ingress dropped packets. Even blockchain nodes will have different ports that are required to be open, so each node type (e.g. authority node, user node, etc.) should have a separate firewall group. During penetration testing there were extraneous ports that were exposed on the various systems/subsystems due to the superset of ports used in the firewall group. For example, the blockchain types of user node and authority nodes were grouped into one firewall rule-set.

Luckily none of these opened ports had other processes that were bound to them. If they had, then at the least a DoS attack could be launched against the associated service. This could be accomplished by a flood attack or through a memory handling error of the service. Worst case scenario would involve a memory handling error with privilege escalation.

4 Attacking the Blockchain

As previously enumerated, blockchain systems incorporate a number of capabilities that lend themselves very well to the requirements of a voting system. However, blockchain's strengths can also be its weakness.

4.1 Breaking the consensus of the node set

If the consensus protocol of the blockchain nodes can be broken, then the attacker has essentially performed a DoS attack on the infrastructure. Breaking consensus protocol can occur in multiple ways, depending which consensus algorithm is used. For example, when using authority rounds, there are strict rules that each of the step nodes have to adhere to during their round. It is time sensitive to the step. Therefore, attacking the timing system of each of the nodes or a shared time sync service can break a node's turn. For example, if the node's system-time, which is based on the time sync protocol, is skewed a couple of seconds then that node would either observe that their peer nodes are out-of-step or that the other nodes would notice that the compromised server is out of step with themselves. As a result, they would either fail or worse, fork a new chain. This could affect the consensus of the nodes and in web3's (Ethereum's client library) current state, a client that binds to the forked chain's node would have issues with freshness, as the smart contracts or transactions would unexpectedly not be in sync. During penetration testing, controlling the time sync caused the targeted blockchain node to break consensus with its peers. With the total set of three authority nodes, after the second targeted node was compromised, consensus was broken. Therefore, when utilizing remote time sync protocols, such as Network Time Protocol (NTP) consider using underlying security protocols (e.g. TLS or IPsec) to secure time sync communication.

4.2 Deriving the private key of the sender account

Each block in a blockchain has digital signatures which is used to verify the authenticity of the transaction as the associated sender of the corresponding

S. Emery et al.

transaction. Some blockchain implementations, such as Ethereum Virtual Machines (EVMs), use Elliptic Curve Digital Signature Algorithm (ECDSA) to verify transaction signatures. Unfortunately client libraries when constructing these signatures had incorrectly generated uniform values [2], k -values, leaving the keys vulnerable to attack.

For the voting blockchain system, there were no duplicate r -values found that could be used to derive the sender's private key. Duplicate r -values would indicate the same k -value because $r = g^k \bmod q$, where g is the generator under the prime field p , and q is the order of the base point \mathbb{G} . From this an attacker can find the signer's private key by first calculating the k -value. Once the attacker has k then solve for x from s :

$$s = k^{-1}(e + xr) \bmod q \quad (1)$$

$$ks = e \bmod n + xr \bmod q \quad (2)$$

$$xr \bmod n = (ks - e) \bmod q \quad (3)$$

$$x = r^{-1}(ks - e) \bmod q \quad (4)$$

where (r, s) is the signature, e is the hash of the message, and x is the sender's private key.

It is recommended that implementations use best practices when deriving uniform k -values. RFC 6979 is one such methodology that ensures stronger security properties are met for ECDSA signatures.

4.3 Brute forcing encrypted wallets

Brute forcing encrypted wallets is also possible, especially if weak cryptographic parameters are specified and deployed, for instance:

- Number of hash iterations: 10240
- Key Derivation Function (KDF): PBKDF2
- Wallet account address: e.g. 0xbadcafedeadbeef...

During penetration testing, weak parameters were utilized by the deployed wallet but the password used to encrypt the wallet was very strong. However, if the password was shorter and had the minimal set of character classes then the associated password could be brute-forced by even a moderate GPU in a relatively short period of time. Better security parameters recommended for this deployment would be:

- Significantly increase hash iterations to at least 262144
- Use a stronger KDF, such as SCRYPT
- Remove the unnecessary account address for privacy reasons. For instance, in Ethereum, the account addresses can be derived from the private key. The private key can be used to derive the public key. The public key can be used to derive the account address.

Penetration Testing a US Election Blockchain Prototype

Just increasing hash iterations from 10240 to 262144 increases the estimated computational time from 15 days to 1 year and 244 days on a modest Nvidia GeForce GT 710 with 2GB of memory at a hash rate of 1,800 H/s. There were not enough resources on the system tested when using SCRYPT as the KDF, vs. PBKDF2.

4.4 Breaking blockchain privacy

Having a decentralized service allows for inherent duplication, but does not provide mechanisms for deduplication. If private data is accidentally incorporated as transaction data to the chain then there is no way of redacting this information, either from soft or hard-forks in the chain. During penetration testing of the voting system's blockchain there were no discernible user identifiers that could correlate a voter with an individual when used as voter ID input to the vote submission smart contract transaction. The sender account address did not reveal any other information about the voter.

In addition, the vote submission transaction does not reveal any of the candidates or propositions selected in the ballot. The reason for this is that the ballot template uses an index for each of the selected candidates or propositions instead of names. However, after correlating the outcome of the election to the submitted ballots, the candidates could be inferred by observing the index # of the ballots. In addition, any write-in for the ballot would be submitted in clear ASCII text as an input argument to the vote submission transaction. An attacker could easily identify voter write-ins simply by decoding the ASCII characters. There is no easy solution for the write-in privacy scenario. A hash would not work given that write-in candidates could be arbitrary and encrypting write-in information would require additional resources which would further increase the attack surface of the system.

4.5 Compromising the smart contracts on the blockchain

There are a number of utilities that can be used to test the security of smart contracts, specifically with the Ethereum framework:

- MAIAN[5]: has greedy, suicidal, and prodigal test cases. The greedy and prodigal test cases deal with cryptocurrency and therefore not relevant to the election smart contracts, accounts, and blockchain reward system. The suicidal test case checks for the EVM kill op-code that could be called by anyone that would disable the associated contract.
- Mythril[4]: contains a dozen test cases that check for insecure delegate calls, deprecated op-codes, insecure low-level calls, integer overflow and underflow, etc. Smart contract source code is not required.
- Echidna[12]: uses fuzz testing of smart contract interfaces. Echidna requires that the targeted smart contracts be modified in order to support invariant testing. This requires a stub function to be created which allows Echidna to assert smart contract logic: always true, sometimes false, and transaction reversion.

S. Emery et al.

- Slither[13]: utilizes a number of test cases in its suite: uninitialized state, local, or storage variables, reentrancy vulnerabilities, unused return values, multiple calls in a loop, unused state variables, etc.

During penetration testing, access to the Solidity smart contract source code was not permitted. This constrained testing to just Mythril and MAIAN. Neither Mythril nor MAIAN found any vulnerabilities/errors in the four voting smart contracts’ byte-code tested: Election Registry, Voter Registry, Election, and Election Verification.

5 Attacking the Voter Application

The voter application front-end and back-end were targeted from both a black-box and white hat approach. The blockchain nodes that were tested were running in Docker containers for the authority and user nodes running on Ubuntu. The database servers that were tested were also running on Ubuntu. There were multiple vectors targeted during penetration testing of the aforementioned systems, including:

- Packet crafting, using two techniques, to induce a memory handling error for either DoS or privilege escalation attack.
- Employing vulnerability scanners to check for any known attacks with the application or dependent libraries
- Static analysis, by scanning dockers images for any known vulnerabilities in the application or dependent libraries.
- Launching a replay attack from the voting mobile application, blockchain, and database traffic.
- Exploiting the privacy of voter information/demographics

5.1 Packet crafting

The goal with this attack was to attempt to exploit memory handling issues with the targeted application and the dependent libraries down the network protocol stack. Tests were created for the packet crafting framework, Scapy[11]. These tests were made in an attempt to trigger a segmentation violation and/or to gain elevated privileges through direct memory manipulation to a privileged operation, e.g. shell access.

During penetration testing of the voting systems both the blockchain and database instances successfully handled fuzz testing and null/zero length messaging. The targeted applications did not have any memory handling issues and gracefully rejected the crafted packet that was sent. However, the fuzz and null messaging was large granular, which means that the crafted payload was contextually significant at the OSI transport layer. In the blockchain and database stack, the associated session, presentation, and application layers were not targeted because of the lack of Scapy modules available, bundled or 3rd party.

Penetration Testing a US Election Blockchain Prototype

Further investigation is required to develop Scapy modules that will support read/writes at JSON, WebSockets, RPC, and blockchain procedure call granularity. This will allow a much more targeted attack on the encoding and data structures of the application and its network stack.

5.2 Vulnerability scanning

In general, there are many areas for vulnerability scanning. Foremost is network scanning, looking for open service ports and testing known vulnerabilities of these services. Another is scanning the operating system for any known vulnerabilities that may be detected. However, administrators do not normally think of Docker containers as vectors for security vulnerabilities. In fact the issue is concerning considering that distributions statically bundle the dependent libraries of their application and forget to update the images once security vulnerabilities have been fixed in the dynamic distribution of the underlying operating system. Trivy[1] is a static scanner that scans docker images looking for any known vulnerabilities of the system libraries. If a vulnerability is found Trivy will report the CVE # and severity of security issues found.

For the Docker containers running the blockchain servers, Trivy reported that the underlying Ubuntu libraries were two major versions behind and four minor releases behind: native 18.04.3 vs. Docker 16.04. The blockchain server's library dependencies, libc6 and libudev1, had 15 vulnerabilities of high, medium, and low severity scores. There were four vulnerabilities that were considered high and possibly relevant for glibc: CVE-2017-18269, CVE-2019-9169, CVE-2018-11236, and CVE-2018-6485. Further investigation is required to see if these high severity issues can be exploited with the blockchain server as a consumer.

Metasploit 7[9] was run remotely against the blockchain nodes and database servers. The ports were open to simulate that a network peer had been compromised. However, there were no exploits found with the blockchain server's network stack:

```
TCP/UDP->HTTP/WebSockets/IPCSocket->RPC->JSON->blockchain server
```

However, this has no specific coverage by Metasploit. The particular database server tested does not have coverage as well. Further investigation is required to develop specific modules for the application protocol and encoding schemes of each of these services in Metasploit.

5.3 Replay attacks on various subsystems

When testing the voting mobile application, network traffic was captured from the mobile application (using a reverse proxy), database, and blockchain servers. The traffic captured was associated with voter registration (sign-up and sign-in) and subsequent ballot submissions. This traffic was subsequently replayed from the respective systems using `tcpreplay(1)`. Tests revealed that there were no memory handling issues on any of the services running, however message latency increased noticeably with the database servers, which deserves further investigation.

S. Emery et al.

5.4 Exploiting the privacy of voter information

During penetration testing a goal was to exploit sensitive voter information. Sensitive voter data could consist of passwords, name, address, SSN, signatures, etc. It was discovered that signatures were stored in the database collection as the affidavit of the submitted ballot of the voter, with the image stored in PNG format. The EO's responsibility is to manually look at the provided voter signature and compare this image with a known signature before approving or rejecting the submitted ballot.

Even though the database servers utilize encryption at rest, the human signatures could be intercepted by a web proxy, discussed in the next section. Given the high target value of signatures the threat of an insider poses some risk.

5.5 Attacking through reverse web proxy

During penetration testing, the mobile application is susceptible to eavesdropping and tampering of voting information. This can be accomplished by using a reverse proxy with a root certificate. An attacker can intercept submitted votes, tamper with votes, or perform a DoS (vote submission dropped). The interceptor scenario can also occur legitimately on corporate devices and networks. User owned devices and their corresponding browser's will detect and alert that interception has occurred, however users are unfortunately trained to click-through warnings in order to gain access to a known service. After all the US voter app must be trustworthy, right? Yes, but the attacker could control the network and the corresponding network proxy. There are a number of techniques to prevent or detect this type of attack, based on the various capabilities of the attacker:

- Browser key pinning could be used to detect proxy issued certificates vs. the actual web server's.
- In recognition of corporate infrastructure, the mobile voter app could disclose to the end user that voting could allow interceptors to observe/change voting.
- The browser could use DNS based Authentication of Named Entities (DANE). However if the attacker also controls the user's DNS access then the attacker could drop DANE requests.
- Traffic could be tunneled through a VPN in order to bypass the attacker's web proxy. However this access may also be denied if the attacker also controls the network's egress access.
- The web applications could utilize the Web Crypto API to encrypt payloads between the voter's mobile device web browser and the web server end point.

Even though an attacker could thwart the voter app's counter-measures, it is better to hard fail than fallback to a mode that would allow an attacker to exploit voter data.

5.6 Attacking mobile app voter sign-up and election registry

The sign-up interface requires an activation QR code or activation number associated with the target election for that user. However, during penetration testing, if another user's activation code was intercepted by an attacker that code could be used as long as the user had a valid account based on e-mail address. After entering a valid activation code the user is prompted for an e-mail address, phone number, and PIN. While entering the e-mail address the string is automatically checked in real-time, which gives an attacker instant verification on whether this is a registered account or not. The phone number can be an arbitrary number chosen by the attacker. The PIN is a sequence of five digits. After submitting, a six digit verification code is sent to the specified mobile number.

Sign-in requires a phone number and the PIN that was previously created by the voter. After entering the phone number and entering the PIN, the app will indicate nothing when entering the data, however when switching focus between the two fields, text under the respective fields will indicate whether either of the two fields is incorrect. This allows a persistent attacker to brute force either the phone number or PIN. Given a PIN that has only five digits this would not take an attacker long to guess with instant feedback. Once submitted a six digit verification code is sent to the specified phone number, for two-factor authentication. If the verification code is entered incorrectly the user will not be allowed to submit the code because the submit button is disabled until the actual code is entered. Again this gives the attacker real-time feedback on whether the verification code is valid or not. With six digits this is also guessable by a persistent hacker or bot.

In order to prevent these types of attacks the text fields should disable real-time checks on the entered data. Only when the data has been submitted should there be feedback on success or failure. If authentication fails, regardless of whether it was due to an invalid account or PIN/password, a generic error message should be displayed. Indicating neither specific account or PIN/password failure. Failures should either trigger an n -strikes algorithm, where the user is no longer able to authenticate for x amount of time or an exponential back-off algorithm used to only allow another attempt after x^y amount of time has expired (where x is a constant and y is the attempt number).

To fix the arbitrary activation code issue the account (e-mail address) must be bound to the activation code rather than pairing the sign-up account with activation code.

6 Recommendations

6.1 Security attestation of integrated blockchain systems

Anytime a new electronic system is employed it is more accessible to legitimate users, but also opens new vectors for malicious users. Given the high stakes of democratic elections it is imperative that the entire election system's security is analyzed. This paper analyzed all available layers in the stack of the blockchain and dependent systems. Criteria for evaluating a potential system includes:

S. Emery et al.

- Determine if blockchain is the right solution for the problem space and that all simpler solutions have been exhausted to meet the same requirements without introducing additional risks.
- Insist that the source code be shared in order to provide transparency and assurances that principles of secure programming have been followed. Including, static and dynamic analysis (e.g. fuzz testing) of the source code.
- Ensure that configuration files have the most restrictive permissions and ownership for the processes that require access to the associated information.
- Key material or passwords are never passed through commands line arguments, regardless if they are short-lived or long-running processes.
- Provide the most restrictive firewall rule-sets that start with restricted access to predetermined ingress and egress traffic, expanding only after bringing up the system.
- Understand how the blockchain establishes consensus and probe against these mechanisms and any of their dependencies for any possible procedural, privilege escalation, and DoS attacks.
- Establish assurances of the cryptographic elements and their dependencies used to construct signatures on the blockchain.
- Check the cryptographic parameters of wallet configurations and establish adequate strengths from the KDF algorithm used, hash iterations, and sufficient privacy in case the wallet is exposed.
- Because of blockchain’s decentralization, take care to ensure that the data contained in the block’s transactions does not expose private data or could be used to infer sensitive information.
- Due to blockchain’s immutable nature and the complexities of transitioning from insecure smart contracts, thoroughly analyze all smart contracts that will be deployed on the blockchain through static and dynamic analysis.
- Think like an adversary by attacking the system through packet crafting, vulnerability scanning, replay attacks, passive/active attacks, and targeting the application and all dependent libraries/protocols/systems.

6.2 Correlating Voting Principles

There are a number of voting principles that systems should adhere to in order to meet the requirements of a free and fair election, regardless if they employ blockchain technology or not:

- Privacy: In order to prevent voter coercion or vote buying, a cast ballot should remain anonymous throughout the voting process. Unfortunately, in the system tested an attacker could look at the blockchain’s transactions in order to find which voting slots were selected by any particular account. If the database were to become compromised the blockchain accounts could be correlated to the voter’s ID. Moving to off-chain transactions for privacy defeats the purpose of including blockchain in the architecture in the first place. This was attacked successfully in penetration testing, see section 4.4.

Penetration Testing a US Election Blockchain Prototype

- Coercion resistance: Having the vote cast on the user’s personal device and transported over the internet allows for coercion on many levels. The voter’s personal device could have malware such as keyloggers that can record the user’s vote for the coercer. Having the vote sent over the internet allows for middleboxes and web proxies to inspect the user’s vote. The user’s mobile device itself can be viewed or screen recorded in order to help ensure the coercer of the desired vote. This was also exploited, refer to section 5.5.
- Verifiability: The system tested did not provide individual verification by the voter that their vote was received/recorded or by any public observers in the tallying and results phase of the election. Verification can be accomplished while still maintaining voter privacy by utilizing a number of cryptographic primitives such as homomorphic encryption, leveraging mix networks, and employing zero-knowledge proofs at the voter verification, tallying, and results phase of the election.
- Fairness: With the system’s vote slot design, blockchain transactions can be observed while the election is in process. This would violate election fairness as votes for candidates could be exposed/published before the tallying and results phase of the election, thereby giving the currently winning candidate a positive view. This vulnerability was discussed in section 4.4.
- Correctness: This encompasses that only valid voters are allowed to vote, voters can only vote once or a vote is counted only once, ballots can not be tampered with without detection, all valid votes are counted in the results, and invalid votes are not counted in tallying nor in the election results. Blockchains do have properties that meet a number of correctness criteria, however there are much more specific and mature technologies, i.e. secure databases, that do not introduce unnecessary attack surfaces, including blockchain’s decentralization, consensus algorithm, and supporting libraries. These potential attacks were referenced in sections 4.1 and 5.2.

6.3 Future Work

Due to a number of factors, there were several work items that we could not explore further during our time with the e-voting system.

- Obtain permission for the smart contract source code to comprehensively test vulnerabilities of the targeted system.
- Implement Scapy modules specifically for the blockchain and database stack.
- Develop Metasploit modules to support various layers of the blockchain stack. Compared to Scapy, testing could go further in exploiting any vulnerabilities found.
- Research exploiting the CVE vulnerabilities found by Trivy.

Given our resources, limited time with the test systems, non-existent modules for the blockchain and database stack, and lack of access to the source code, the authors estimate, with all of the above work to be done, that we have only completed 50% of a comprehensive penetration test.

S. Emery et al.

7 Conclusion

With the intensive efforts to disrupt democratic societies of their inalienable right to have a free and fair election, it is critical that voting systems are vetted holistically and objectively as possible. Deploying blockchain as a framework for the voting system plays to blockchain's strengths of having a decentralized, immutable, verifiable, and robust subsystem. However, our research and related penetration testing of a blockchain based electronic voting system prototype has shown that a blockchain's strengths can also be its own weakness. Having a decentralized ledger allows access to those accounts that may fall victim through various attacks on the wallets and even forms of social engineering. The vectors of attack are numerous on these systems and are only as strong as its weakest subsystem. Vigilance is required to protect the precious right of voting, but not even vigilance will be enough without looking at the system holistically. Technology alone currently does not meet the high assurances required for free and fair elections. Our hope is that this work will add to the body of knowledge of what and how an e-voting system that utilizes blockchain technology could and will invariably be attacked.

References

1. Aqua: Trivy (Oct 2019), <https://github.com/aquasecurity/trivy>
2. Brengel, M., Rossow, C.: International Symposium on Research in Attacks, Intrusions, and Defenses RAID 2018. pp. 623–643 (Sep 2018). <https://doi.org/10.1007/978-3-030-00470-5>
3. Cimpanu, C.: Moscow's blockchain voting system cracked a month before election (Aug 2019), <https://www.zdnet.com/article/moscows-blockchain-voting-system-cracked-a-month-before-election>
4. ConsenSys: Mythril (Oct 2019), <https://github.com/ConsenSys/mythril>
5. Maian: Maian (Mar 2018), <https://github.com/MAIAN-tool/MAIAN>
6. Matishak, M.: What we know about russia's election hacking (Jul 2018), <https://www.politico.com/story/2018/07/18/russia-election-hacking-trump-putin-698087>
7. National Academies of Sciences, Engineering, and Medicine: Securing the Vote: Protecting American Democracy, pp. 103–106. The National Academies Press, Washington, DC, USA (2018). <https://doi.org/10.17226/25120>
8. Park, S., Specter, M., Narula, N., Rivest, R.L.: Going from bad to worse: from Internet voting to blockchain voting. *Journal of Cybersecurity* **7**(1) (Feb 2021). <https://doi.org/10.1093/cybsec/tyaa025>
9. Rapid7: Metasploit: Penetration testing software - rapid7 (Nov 2016), <https://www.rapid7.com/products/metasploit>
10. Scarfone, K., Jansen, W., Tracy, M.: SP 800-123. Guide to General Server Security (2008). <https://doi.org/10.6028/NIST.SP.800-123>
11. SecDev: Scapy (Oct 2019), <https://github.com/secdev/scapy>
12. Trail of Bits: Echidna (Aug 2019), <https://github.com/crytic/echidna>
13. Trail of Bits: Slither (Oct 2019), <https://github.com/crytic/slither>
14. Wolchok, S., Wustrow, E., Isabel, D., Halderman, J.: Attacking the Washington, D.C. Internet Voting System (Feb 2012). https://doi.org/10.1007/978-3-642-32946-3_10

How to Break Virtual Shareholder Meetings: An Empirical Study^{*}

Andreas Mayer^[0000–0001–7055–8876]

Heilbronn University of Applied Sciences
Max-Planck-Str. 39, 74081 Heilbronn, Germany
andreas.mayer@hs-heilbronn.de

Abstract. The Covid-19 pandemic has had a major impact on annual general meetings (AGMs) of shareholders worldwide. In 2020, quickly passed emergency laws prohibited on-site AGMs. Therefore, shareholder meetings shifted from a physical to a virtual voting event.

In this paper, we present our large-scale study on the security of 623 virtual AGMs held by German companies including corporations listed in stock indices such as DAX and MDAX. We found several severe vulnerabilities in six out of eight online voting platforms, breaking confidentiality, integrity, and availability. In 72% of all virtual AGMs at least one of the three security goals was compromised. We responsibly disclosed all weaknesses and helped to develop fixes.

Keywords: annual general meeting (AGM) · online voting · web security

1 Introduction

Due to the Covid-19 pandemic emergency laws have been adopted in several states worldwide. They prohibit annual general meetings of shareholders with physical attendance. Therefore, the DAX-listed company Bayer AG conducted the first purely virtual annual general meeting (vAGM) in Germany on April 28, 2020. Over 5,000 shareholders participated at a total cost of around € 1 million [4].

In 2020, the vAGMs emerged to the new standard for shareholder meetings. The vast majority of German stock corporations (78%) conducted a virtual event in last years AGM season. This trend proliferated in 2021 and in the long term politics already consider vAGMs as permanent alternative to general meetings with physical attendance [13].

An AGM allows shareholders to vote on both company issues and the election of the company’s board of directors. Therefore, the shareholders’ AGM is the foundation for sound corporate governance. Shifting the face-to-face meetings to the virtual world launched manifold new potential security risks. Consequently,

^{*} This article is based on a study first reported in the “Tagungsband zum 17. Deutschen IT-Sicherheitskongress des BSI” [6].

A. Mayer

companies outsource the organisation and execution of vAGMs to specialized vendors that provide an online voting platform (oVP) to the shareholders. Due to the novelty of oVPs and their usage, the security is barely studied. Thus, we raise the following research questions in this paper:

- **RQ1:** Which oVPs exist and how large is their market share in Germany?
- **RQ2:** How secure are those oVPs in respect to well-known web attacks?

To answer these questions, we conducted an empirical study covering 623 vAGMs of German corporations in the period from April 28 to December 31, 2020. In total, we identified 15 AGM service providers with eight different oVPs. On basis of our threat analysis and the attacker model (Section 2) we performed a systematic security analysis. To this, we participated in 46 vAGMs with 71 different user accounts. Our methodology is presented in Section 3.

During our security analysis, we found critical threats in the context of vAGMs which (partially) correlate to well-known vulnerabilities and attacks derived from the Open Web Application Security Project (OWASP) Top 10 list¹ [9]. Our analysis involves information gathering beforehand and the examination of the deployed security best practices. Based on the results, we estimated the potential attack surface.

In summary, six out of eight oVPs, covering 72% of the 623 vAGMs, had critical vulnerabilities that compromised at least one of the three information security goals, namely confidentiality, integrity, and availability. Our attacks allowed the fraudulent alteration of shareholders' votes, the complete takeover of shareholder accounts, effective denial of service attacks, and the large-scale leakage of personal data. By spending approximately € 20 an attacker was able to access the personal data (e.g. name, address, date of birth, etc.) of *all* shareholders, including voting behavior and number of voting rights, from *all* vAGMs conducted on the vulnerable oVP. In total, 161 vAGMs were affected by this privacy issue. Section 4 presents the empirical study including the revealed attack surface and the found vulnerabilities. Afterwards, we discuss our findings in Section 5 and conclude in Section 6.

Contributions. In summary, we make the following contributions:

- We conducted a large-scale study by systematically analyzing the security of 623 vAGMs in Germany. To our best knowledge, we are the first that have done such an evaluation.
- We found several severe vulnerabilities in six out of eight real-world oVPs breaking the information security goals confidentiality, integrity, and availability of the underlying online-voting systems. In 72% of all vAGMs at least one security goal was compromised. The found vulnerabilities affected vAGMs of major companies listed in stock indices such as DAX and MDAX.
- We disclosed all vulnerabilities found to the platform owners in a responsible disclosure process and supported them in developing fixes.

¹ The OWASP Top 10 list represents a broad consensus about the most critical security risks to real-world web applications.

2 Threat Analysis

In the following, we present threats that arise from the assets and the use cases an oVP typically provides. These threats may be exploited by vulnerabilities existing in the oVP itself or in the associated infrastructure. An attacker can use these vulnerabilities to launch attacks that may break the security goals (i.e. confidentiality, integrity, and availability) of the oVP. Therefore, a successful attack may break some or all underlying election principles.

2.1 Assets and Use Cases

From the attacker's perspective, the most valuable asset stored in an oVP is the personal data of the shareholders. This data comprises first name, last name, place of residence, number of shares/votes, voting behavior, and the type of share possession (own/proxy). Furthermore, each shareholder has a unique shareholder number, often used as username to log in to the oVP. In addition, sensitive data such as full address, e-mail address, telephone number, date of birth, nationality and the bank that reported the shareholding is often stored, too.

Looking at the functionality, an oVP typically provides the following use cases:

- Online voting functionality
 - Voting by electronic absentee ballot including changes/revocations
 - Authorization of and instructions to the company's proxies
 - Granting power of attorney to a third party
 - Voting confirmation via a generated pdf document
- Submission of questions to the company (in advance of the AGM)
- Declaration of objections to AGM resolutions
- Content section to provide additional documents such as presentations, annual reports or the list of attendees during the vAGM
- Video Streaming
- Login/logout functionality

2.2 Attacker Model

In our attacker model, the perpetrator possesses the following capabilities:

1. **Access to oVP:** The URL to the oVP is published in the public invitation to the AGM and/or on the company's website. Therefore, the oVP can be accessed by anyone via the Internet.
2. **Log in to the oVP:** We assume, that the attacker is a shareholder of a company using a susceptible oVP for their vAGM. Therefore, the attacker has his own legitimate oVP account with username and password. Please note that buying one share is sufficient to attend the company's AGM.
3. **Victim may click on a link:** Further, the attacker may trick a legitimate vAGM participant (the victim) to click on a link (e.g., by posting on a discussion forum or sending an e-mail). The victim must be logged into the oVP at the same time.

A. Mayer

2.3 Results Threat Analysis

In Table 1, we summarize the results of our threat analysis. We consider three severe real-world threats, each of which compromises one of the three fundamental security goals. For each threat we mapped the corresponding vulnerability classes from the OWASP Top 10 list as shown in the second column. In our empirical study, we used this vulnerability classes to perform a systematic manual black box penetration test (see Section 3). The last column shows well-known attacks that can be used to exploit the vulnerabilities. In the following, we briefly explained the attacks:

1. **Brute force password guessing:** If weak passwords are used to log in to the oVP they may be guessed by an automated brute force attack.
2. **Session fixation attack [5]:** In a successful session fixation attack, the victim uses a session id predetermined by the attacker. In consequence, the attacker can take over the victim’s identity on the oVP.
3. **Broken access control:** Due to missing or incorrectly implemented access control mechanisms authenticated users may be able to access sensitive data (e.g., the personal data of other shareholders) or may trigger unauthorized actions.
4. **Cross-site request forgery (CSRF) [14]:** These attacks arise when a malicious web site causes a victim’s web browser to execute an unwanted action on a trusted site. Regarding the usage of an oVP this can be for example changing the victim’s voting behavior.
5. **Brute force account locking:** A frequently used countermeasure to prevent automated password guessing is to lockout accounts after a defined number of incorrect password attempts. However, this measure may easily be abused by the attacker to lockout hundreds of user accounts. This allows an attacker to exclude individual or all shareholders on attending the vAGM.

Table 1. The results of the threat analysis mapped to OWASP Top 10 vulnerability classes. The resulting example attacks exploit these vulnerabilities and break the security goals.

Threat	OWASP Top 10	Example attacks	Security goal broken
Attacker may get access to personal data and/or the victims’ voting behavior	A2:2017, A5:2017	Brute force password guessing, session fixation, broken access control	Confidentiality
Attacker may manipulate personal data and/or the victims’ voting behavior	A2:2017, A5:2017, A8:2013	Brute force password guessing, session fixation, broken access control, CSRF	Integrity
Attacker may block shareholder accounts and thus prevent participation in the vAGM	A2:2017	Brute force account lockout	Availability

3 Methodology

In this section, we describe the methodology used to carry out our systematic security analysis (cf. Figure 1). The Federal Gazette² publishes all AGM invita-

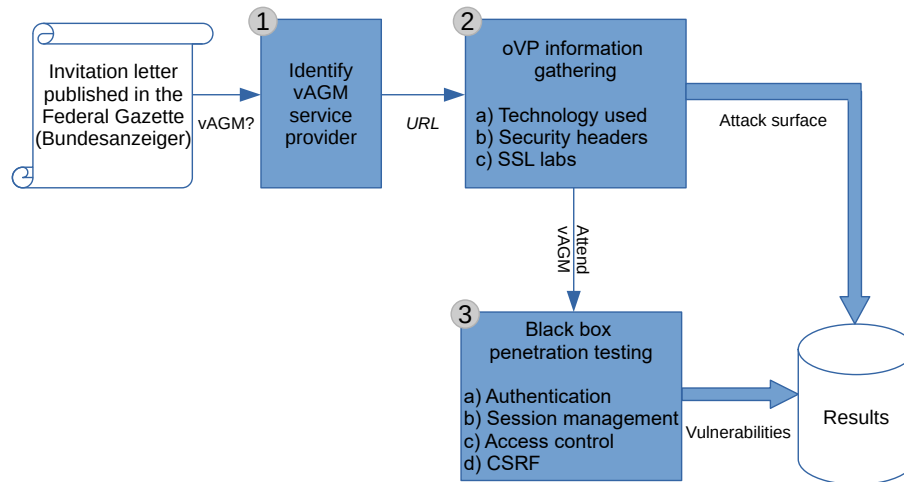


Fig. 1. The methodology we used for our systematic security analysis. First, we identified the existing vAGM service providers. In the second step, we analyzed the attack surface by collecting publicly available information about the different oVPs. Finally, we conducted a black box penetration test.

tion letters and served us as data source. We used these announcements to check if the company wants to conduct the shareholder meeting virtually. If yes, the following steps were carried out:

1. **Identify AGM service provider:** We identified the mandated AGM service provider (source: invitation letter) and the *URL* to the oVP (source: company website and/or invitation letter).
2. **Information gathering:** To estimate the potential attack surface and to gain an impression of the security best practices deployed, we collected the following publicly available information:
 - a) **Technologies used:** We used the free browser extension Wappalyzer [11] combined with manual investigation of HTTP headers to find out the technology stack used (i.e. third-party software libraries, operating system, webserver type, and infrastructure components). Furthermore, we

² The Federal Gazette (Bundesanzeiger) is an official publication of the Federal Republic of Germany that announces all legally relevant company news. All announcements are freely available under <https://www.bundesanzeiger.de>.

A. Mayer

analyzed whether outdated software libraries with known vulnerabilities were used.

- b) Security header rating: Nowadays, a large number of well-proven HTTP security headers (e.g. HSTS³) exist that can be used to increase the security of web applications. The deployment of these security headers was investigated using the free website “Security Headers” developed by Scott Helme [2]. The service assigns ratings ranging from A+ (very good) to F (insufficient).
 - c) SSL Labs rating: Another important cornerstone of the oVPs’ security is the correct deployment of the TLS protocol (formerly known as SSL) to secure the network traffic via HTTPS. The free service “SSL Labs” [10] was used to analyze and rate the security of the TLS configuration. The service assigns ratings ranging from A+ (very good) to F (insufficient).
3. **Black box penetration testing:** To get full access to the oVPs, we bought shares of different companies and registered to the shareholder meetings. Building on the results of our threat analysis (cf. Section 2), we then performed a manual black box penetration test to find exploitable vulnerabilities. We systematically analyzed the following four security critical scopes:
- a) Authentication
 - b) Session management
 - c) Access control
 - d) CSRF

4 Security Evaluation

In this section, we present the results of our empirical study we conducted on the security of German vAGMs throughout the general meeting season 2020. We systematically investigated the oVPs we found in the wild according to the methodology presented in Section 3. To fully carry out the security evaluation, we participated in 46 vAGMs with 71 different shareholder accounts.⁴ The evaluation period starts with the first German vAGM (Bayer AG) on April 28, 2020 and ends on December 31, 2020. In total, 623 vAGMs were held during this period. In two cases we could not identify the mandated AGM service provider. Of the remaining 621 vAGMs, 584 (94%) were conducted by 15 different AGM service providers. In 23 cases (3.7%), a video conferencing system such as Zoom was used without a dedicated oVP. Here, polling was done conventionally by absentee voting, fax or even e-mail. On 14 vAGMs (2.3%), the companies used an oVP software developed on their own.

Figure 2 shows the market share of all discovered AGM service providers in Germany. The AGMs conducted via a video conferencing system and the in-house developed AGM portal solutions are not considered further.

³ HTTP Strict Transport Security (HSTS) [3] is a security header for HTTPS connections that is designed to protect against downgrade attacks and session hijacking.

⁴ To legitimately attend a vAGM you need to buy at least one company share. In most cases we had two accounts per vAGM.

During further investigations of the oVPs, it turned out that the AGM service providers UBJ. GmbH, GFEI AG, ITTEB GmbH & Co. KG, BADER & HUBL GmbH, HV-Management GmbH, AAA HV Management GmbH, Art of Conference, and HV AG use the same underlying platform. Therefore, we summarized them to “BS portal”. In the end, the number of different oVPs discovered reduced to eight platforms.

The oVPs of the three largest AGM service providers Computershare GmbH & Co. KG, Link Market Services GmbH and Better Orange IR & HV AG dominate the market for vAGMs with a total market share of 66.8%. On the fourth place BS HV portal has a combined total market share of 19.2%. The remaining 8% are shared by four service providers, namely C-HV AG (3.1%), ADEUS Aktienregister-HV-Service GmbH (2.9%), HVBest Event-Service GmbH (1.0%), and FAE Management GmbH (1.0%).

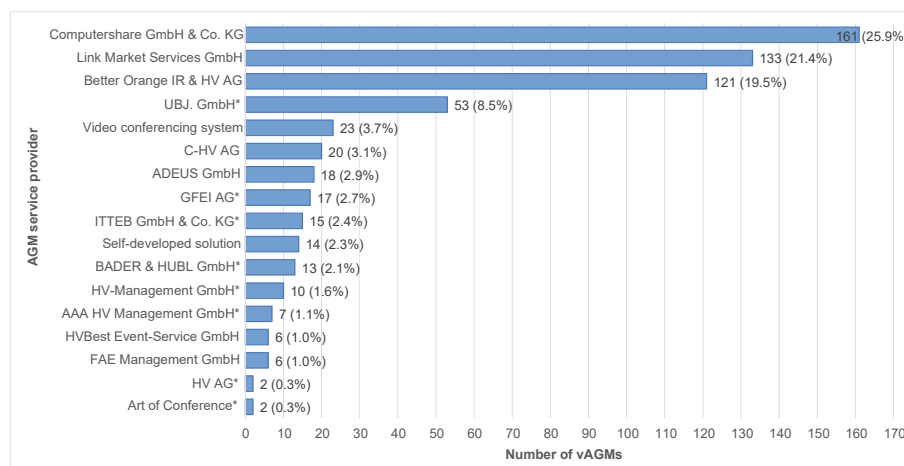


Fig. 2. Number of conducted vAGMs per service provider. The top three service providers dominate more than two thirds of the vAGM market in Germany. Service providers marked with “*” are using the same underlying oVP. Therefore, they are summarized and referred to as “BS portal”.

4.1 Information Gathering

Table 2 summarizes the results of the information gathering. Seven out of eight oVPs examined are classic web applications with server-side application logic. Only Computershare GmbH & Co. KG relies on a modern Javascript-based single page application. The portals are built in various programming languages (1x Javascript, 2x PHP, 2x ASP.net and 3x Java), frameworks (Angular JS, JavaServer Pages, Java Server Faces, Telerik Web UI, and CodeIgniter), and were

deployed on different web servers (3x Microsoft IIS, 2x Nginx and 1x Apache). In two cases, we were unable to identify the web server used.

In total, ten outdated software libraries with known vulnerabilities were found in five of eight oVPs (market share: 52.4%). The vulnerable libraries date from 2015–2019. No software libraries with known vulnerabilities were discovered in the oVPs of Link Market Services GmbH, BS portal, and HVBest Event-Service GmbH.

All oVPs are accessible via HTTPS and deploy the TLS protocol. Regarding the SSL labs rating, four out of eight oVPs achieved a B rating (market share: 49.2%). Link Market Services GmbH, Better Orange IR & HV AG, and FAE Management GmbH (market share: 41.9%) achieved an A rating. Only the oVP of ADEUS GmbH achieved the best possible rating of A+ (market share: 2.9%).

In the security header rating, Computershare GmbH & Co. KG, and ADEUS GmbH performed best with a C rating (market share: 28.8%). HVBest Event-Service GmbH achieved a D rating. The remaining HV portals with a market share of 64.2% achieved a F rating.

Table 2. The results of the information gathering from step 2 of our methodology.

Broken security Goal	Technology	Vulnerable libraries	SSL Labs Rating	Header Rating
Computershare GmbH & Co. KG	Microsoft IIS, Angular JS	5	B	C
Link Market Services GmbH	Web server unknown, PHP	-	A	F
Better Orange IR & HV AG	Nginx, Java Server Faces	1	A	F
BS portal	Microsoft IIS, ASP.net, Telerik	-	B	F
C-HV AG	Microsoft IIS, ASP.net, Telerik	1	B	F
ADEUS GmbH	Web server unknown, Java	1	A+	C
HVBest Event-Service GmbH	Nginx, Java Server Pages	-	B	D
FAE Management GmbH	Apache, PHP, CodeIgniter	2	A	F

4.2 Black Box Penetration Testing

Authentication. First, we examined the password based authentication. The results are shown in Table 3. All AGM service providers deploy numeric usernames, which are assigned in an ascending order. They are usually 4–5 digits long, starting with leading zeros. Due to this type of account assignment, the usernames are easy to guess.

Next, we analyzed the password policies each service provider employs. This analysis is based on 71 legitimate vAGM accounts. With one exception, all service providers assign randomly generated passwords to the accounts. The password length ranges from 5–10 characters. The password space consists of uppercase and lowercase letters and numbers from the ASCII charset. Additionally, in one case, the special character “*” was included in the password space. The HVBest Event-Service GmbH oVP does not use random passwords. Instead, a combination of the number of shares held, zip code, and place of residence of the shareholder must be entered to authenticate.⁵ The password policies of HV AG and Art of Conference could not be investigated as we did not have any accounts (market share: 0.6%).

The oVPs of Better Orange IR & HV AG and HVBest Event-Service GmbH did not respond with a generic authentication error message on wrong login attempts (market share: 20.5%). Instead, they indicate to the user whether either username or password was wrongly entered. This provides an attacker valuable information.

Link Market Services GmbH and HVBest Event-Service GmbH additionally secure their logins via CAPTCHAs⁶ which must be solved in order to login (market share: 22.4%).

The oVPs of Computershare GmbH & Co. KG, C-HV AG, ADEUS GmbH, and FAE Management GmbH locked accounts after entering multiple wrong passwords (market share: 32.9%). While FAE Management GmbH locked the accounts temporarily for approx. 30 minutes the other three oVPs need manual intervention of the support for unlocking. All other service providers, did not lockout accounts after entering multiple wrong passwords.

Session Management. The three oVPs Better Orange IR & HV AG, BS portal, and C-HV AG were vulnerable to session fixation attacks (market share: 41.8%). The logout functionality of the oVPs of Better Orange IR & HV AG and Computershare GmbH & Co. KG was without any effect (market share: 45.4%). In consequence, the user session remained valid after the user had clicked on the logout button.

Access Control. We revealed a severe broken access control vulnerability in Computershare’s oVP (market share: 25.9%). No further broken access control vulnerabilities were found at any of the other oVPs.

CSRF. Solely, FAE-Management GmbH was susceptible to CSRF attacks.

Table 4 summarizes the results of our penetration testing objectives session management, access control, and CSRF.

⁵ HVBest Event-Service GmbH told us that they also support alternative authentication methods.

⁶ A “Completely Automated Public Turing test to tell Computers and Humans Apart” (CAPTCHA) is a challenge–response test used to determine if a human is interacting with the service.

Table 3. The evaluation results of the password based authentication. (n/a: no data available)

AGM service provider	Password policy	Insecure login message	CAPTCHAs	Account lockable
Computer Share GmbH & Co. KG	6 letters [A-Z, a-z, 0-9] or [A-Z, a-z]			✗
Link Market Services GmbH	6 letters [0-9]		✗	
Better Orange IR & HV AG	8 letters [A-Z, a-z, 0-9, *]	✗		
UBJ. GmbH	5 letters [A-Z, a-z, 0-9] or 8 letters [0-9]			
C-HV AG	8 letters [A-Z, 0-9]			✗
ADEUS GmbH	10 letters [0-9]			✗
GFEI AG	5-6 letters [A-Z, a-z, 0-9]			
ITTEB GmbH & Co. KG	6 letters [A-Z, a-z, 0-9]			
BADER & HUBL GmbH	10 letters [a-z, 0-9]			
HV-Management GmbH	8 letters [A-Z, a-z, 0-9]			
AAA HV GmbH	5 letters [0-9]			
HVBest Event-Service GmbH	No. shares, zip, city	✗	✗	
FAE Management GmbH	8 letters [0-9, a-f]			✗ (30 min.)
HV AG	n/a			
Art of Conference	n/a			

Table 4. The results of the security evaluation of session management, access control, and CSRF. (✗: vulnerability found, n/a: not tested)

oVP	Session fixation attack	Session log-out defect	Broken access control	CSRF attack
Computershare GmbH		✗	✗	
Link Market Services GmbH				
Better Orange IR & HV AG	✗	✗		
BS portal	✗			
C-HV AG	✗			
ADEUS GmbH				
HVBest Event-Service GmbH	n/a	n/a	n/a	n/a
FAE Management GmbH				✗
Total successful attacks (\sum : 7)	3	2	1	1

5 Discussion

In this section, we discuss the results of our security evaluation presented in Section 4.

5.1 Information Gathering

Vulnerable libraries. We found ten third-party libraries with known vulnerabilities in five of eight oVPs with a market share of 52.4%. It should be noted, that using a susceptible library does not necessarily mean that the vulnerability can be exploited. For example, vulnerable code that never gets executed cannot be exploited by an attacker. However, today’s security best practice demands that vulnerabilities should be patched as soon as possible. In contrast, the vulnerabilities found were publicly known for 1–5 years and proper security patches exist.

TLS security. The TLS configuration of the oVPs did not reveal any severe vulnerabilities. Due to the server-side support of TLS 1.0 and 1.1, half of the oVPs only obtained a B rating. The Internet Engineering Task Force (IETF) recommends to remove support of both protocol versions because of their inherent weaknesses since mid-2018 [7]. Finally, the IETF deprecated both versions in March 2021 [8]. We would recommend to turn off support of legacy TLS 1.0 and 1.1. This would immediately improve the security of the web traffic without loss of compatibility.

Security headers. The security header analysis showed that a large number of security headers are rarely deployed by the oVPs. This reveals a unnecessarily large attack surface. The adoption of widely supported security headers may prevent or at least hinder many known attacks. From developers perspective these “low hanging fruits” can easily be implemented in a short time.

Overall, the results revealed a broad attack surface providers’ could easily reduce.

5.2 Black Box Penetration Testing

Authentication. Seven out of eight oVPs generate random passwords for the shareholder accounts. This prevents weak passwords, such as “12345” or the use of same passwords for different services. According to the current recommendations of the German Federal Office for Information Security, a secure password should be at least eight letters long and should consist of upper and lower case letters, numbers, and special characters. Additionally, passwords that do not use all four character types should be significantly longer (>12 characters) or multi-factor authentication should be deployed [1].

These password requirements were not met by any of the investigated service providers. AAA Management HV GmbH deployed the weakest password policy with a password space of 10^5 combinations. HVBest Event-Service GmbH did not

use passwords in all examined cases. Instead, for authentication it was necessary to enter username, number of shares held, postal code, and place of residence. Compared to a strong password policy this type of authentication is considerably weak as the required credentials are easy to obtain (zip code and place of residence) or guessable (number of shares).

The AGM service providers argued that complex password policies would lead to higher support costs. Due to the critical threats that may arise by deploying weak authentication mechanisms, we strongly recommend more complex password policies or multi-factor authentication.

To counter brute force password guessing attacks HVBest Event-Service GmbH and Link Market Services GmbH deploy CAPTCHAs. However, the CAPTCHAs prompted to users are simple in structure and not robust enough to satisfy current security best practices (see Figure 3). In 2010 J. Yan and A. S. El Ahmad showed how CAPTCHAs of this type can be broken [12].



Fig. 3. Examples of CAPTCHAs deployed by Link Market Services GmbH (left) and HVBest Event-Service GmbH (right).

To prevent systematic password guessing attacks, four oVPs lockout accounts after multiple incorrect login attempts. Furthermore, FAE Management GmbH tries to mitigate permanent blocking by automatically unlocking accounts after approximately 30 minutes. However, in all cases an attacker may block individual or all shareholder accounts right before a vAGM. Due to the predictable nature of the usernames this can result in an effective denial of service attack (DoS). In addition, Better Orange IR & HV AG and HVBest Event-Service GmbH (market share: 20.5%) allow to identify valid usernames depending on the error responses. Thus, we recommend displaying a generic error message to the user (e.g. “Login failed!”).

Session management. We discovered session fixation attacks in three oVPs (45.8% market share) allowing an attacker to completely take over the victim’s user session. To mitigate this attack, a fresh session id must be assigned to the user after successful login. Finally, the broken logout functionality found in two oVPs (market share: 45.4%) may also allow session hijacking if session data gets compromised. For example, if an attacker gets access to the victim’s browser after clicking on the (defect) logout button.

Access control. We found a severe vulnerability leading to a broken access control attack in the most prevalent oVP operated by Computershare GmbH & Co.

KG (market share: 25.9%). The vulnerability allowed an attacker to get full access to the personal data of *all* shareholders including name, address, date of birth, voting behavior, and number of shares held from *all* vAGMs conducted by Computershare.⁷ The only prerequisite was a legitimate account by buying at least one share and registering for the vAGM. Unfortunately, the attack was not limited to the vAGM the account was created for – it allowed to read out the personal shareholder data of all conducted vAGMs.

CSRF attacks. We found one oVP susceptible to CSRF attacks (market share: 1.0%) that may allow to (stealthily) change the voting behavior of the victim. For this, the victim must click on a link provided by the attacker while having an active user session to the oVP.

Summary. Our results were summarized in Table 5. The table shows which security goals were broken by the attacks found per oVP. We have not considered the ramifications of the weak password policies found as we did not execute brute force password guessing attacks on the oVPs. However, all AGM service providers contacted during the responsible disclosure process classified this as a critical attack vector that has to be mitigated.

In total, we found in six out of eight oVPs (market share: 71.6%) attacks that at least broke one security goal. Solely, the oVP of Link Market Services GmbH exposed no exploitable vulnerabilities. The oVP of HVBest Event-Service GmbH could not be fully tested due to missing login accounts. Nevertheless, the results of the information gathering show that both portals expose a unnecessary large attack surface.

Table 5. Summary of the broken security goals per oVP and the affected market share. (C: confidentiality, I: integrity, A: availability, **X**: broken, n/a: not tested)

oVP	Market share	C	I	A
Computershare GmbH & Co. KG	25.9%	X		X
Link Market Services GmbH	21.4%			
Better Orange IR & HV AG	19.5%	X	X	
BS portal	19.2%	X	X	
C-HV AG	3.1%	X	X	X
ADEUS GmbH	2.9%			X
HVBest Event-Service GmbH	1.0%	n/a	n/a	n/a
FAE Management GmbH	1.0%		X	
\sum Broken security goals (market share)		4/8 (67.7%)	4/8 (42.8%)	3/8 (31.9%)

⁷ According to article 40, section 1 of the German Securities Trading Act, major shareholders are known to the public as they must publish the acquisition/disposal of shares when reaching or falling below major holdings (i.e. 3%, 5%, 10%, 15%, 20%, 25%, 30%, 50%, and 75%). However, changes of voting rights within the prescribed limits and all other shareholders remain anonymous to the public.

6 Conclusion

The Covid-19 pandemic defined vAGMs as the new normal for shareholder gatherings. They have established themselves as a system-relevant crisis instrument within a very short time. However, in the long run they will only be successful if the underlying oVPs provide an adequate level of security. The results of our empirical security study on vAGMs in Germany show that there is room for improvements. In six out of eight oVPs with a market share of almost 72%, we discovered severe vulnerabilities. It is important to stress, that we only investigated the oVPs against well-known web attacks and analyzed the deployment of security best practices in order to measure the potential attack surface. Therefore, we only scratched on the surface of the portals' security. Our findings are a first step to increase the security level at vAGMs and helped to sensitize the AGM service providers.

Acknowledgements

The author would like to thank the anonymous reviewers, Maximilian Westers, and Vladislav Mladenov for their helpful comments.

References

1. Bundesamt für Sicherheit in der Informationstechnik: Sichere Passwörter erstellen. https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Informationen-und-Empfehlungen/Cyber-Sicherheitsempfehlungen/Accountschutz/Sichere-Passwoerter-erstellen/sichere-passwoerter-erstellen_node.html, accessed: 2021-09-09
2. Helme, S.: Security headers. <https://securityheaders.com/>, accessed: 2021-09-09
3. Hodges, J., Jackson, C., Barth, A.: HTTP Strict Transport Security (HSTS). RFC 6797 (Nov 2012). <https://doi.org/10.17487/RFC6797>, <https://rfc-editor.org/rfc/rfc6797.txt>
4. JUVE Verlag für juristische Information GmbH: Aktionärstreffen: Linklaters-Mandantin Bayer spart mit Online-HV 2,5 Millionen Euro. <https://www.juve.de/nachrichten/deals/2020/04/aktionaerstreffen-linklaters-mandantin-bayer-spart-mit-online-hv-25-millionen-euro> (Apr 2020), accessed: 2021-09-09
5. Kolšek, M.: Session fixation vulnerability in web-based applications (December 2002), https://www.acrossecurity.com/papers/session_fixation.pdf, accessed: 2021-09-09
6. Mayer, A.: Virtuelle Hauptversammlungen: Ein sicherer Ersatz für Präsenzveranstaltungen? In: Bundesamt für Sicherheit in der Informationstechnik (ed.) Deutschland. Digital. Sicher. 30 Jahre BSI. Tagungsband zum 17. Deutschen IT-Sicherheitskongress. pp. 233–248. SecuMedia Verlag, Gau-Algesheim (Feb 2021)

7. Moriarty, K., Farrell, S.: Deprecating TLSv1.0 and TLSv1.1. Internet-Draft draft-moriarty-tls-oldversions-diediedie-00, Internet Engineering Task Force (Jun 2018), <https://datatracker.ietf.org/doc/html/draft-moriarty-tls-oldversions-diediedie-00>, work in Progress
8. Moriarty, K., Farrell, S.: Deprecating TLS 1.0 and TLS 1.1. RFC 8996 (Mar 2021). <https://doi.org/10.17487/RFC8996>, <https://rfc-editor.org/rfc/rfc8996.txt>
9. OWASP Foundation Inc.: OWASP Top 10 – 2017. Publication, OWASP Foundation (November 2017), <https://owasp.org/www-project-top-ten/2017/>, accessed: 2021-09-09
10. Qualys: SSL Labs. <https://www.ssllabs.com/ssltest/>, accessed: 2021-09-09
11. Wappalyzer: Wappalyzer - browser add on. <https://www.wappalyzer.com>, accessed: 2021-09-09
12. Yan, J., El Ahmad, A.S.: Captcha robustness: A security engineering perspective. *Computer* **44**(2), 54–60 (2011). <https://doi.org/10.1109/MC.2010.275>
13. Zeit Online: Justizminister wollen dauerhaft virtuelle Hauptversammlung. <https://www.zeit.de/news/2021-06/17/justizminister-wollen-dauerhaft-virtuelle-hauptversammlung>, accessed: 2021-09-09
14. Zeller, W., Felten, E.W.: Cross-site request forgeries: Exploitation and prevention. *The New York Times* pp. 1–13 (2008)

Tales from the Trenches: Case Studies in Election Cybersecurity Preparedness in Texas

Elizabeth Kasongo^{1,3}, Matthew Bernhard², and Chris Bronk¹

¹University of Houston, ²VotingWorks, ³Louisiana Workers' Compensation Corporation

Abstract. Events surrounding the 2016 election violently shook the U.S. elections environment. Since then, numerous policy changes have been implemented. Despite this, the 2020 election was still one of the most contentious elections in U.S. history, up to and including historic-levels of violence and unrest. We conducted post-mortem interviews with three election officials at the county level in Texas to get a better idea about what went well, what when poorly, and what must be addressed going forward. How well did the policy changes post-2016 bolster our confidence in elections in 2020? The answer is quite a lot, but not enough to accommodate new issues like the COVID-19 pandemic and unforeseen levels of domestically-generated misinformation, which overshadowed policy successes in securing systems from outside manipulation by cyberattack.

1 Introduction

The 2016 U.S. presidential election witnessed overt attempts by foreign powers to influence the outcome [23]. Multiple elections jurisdictions across the U.S. experienced attempts to infiltrate the infrastructure they use to conduct elections, with infiltration succeeding in at least two states [9]. In the wake of these attacks, significant effort was made to shore up the defenses of elections in the United States, including significantly more funding for elections [25], the establishment of the Cybersecurity and Infrastructure Security Agency (CISA), and major emphasis on information sharing bodies like the Information Sharing and Analysis Centers and Organizations [11]. Numerous social media companies also significantly revamped efforts to combat misinformation that was widespread during the 2016 elections.

These efforts were largely successful, with the U.S. elections community concluding that the 2020 presidential election was “the most secure in American history” [13]. Despite this assessment and the tremendous amount of money and effort poured into election infrastructure, the U.S. witnessed something extraordinarily rare after the 2020 election: political violence. On January 6th, hundreds of people stormed the U.S. Capitol building, injuring the security force attempting to protect the building and killing one officer [4]. Calls of violence have run rampant after the election, and even elected officials expressed serious skepticism, up to and including voting against certifying the election results [29].

In this paper, we attempt to shed light on how these two seemingly disparate things can both be true at the same time: elections in the U.S. are more secure

than ever before, and yet discontent with the electoral system and concerns about its security are at an all time high. We argue that many of the policy changes made post-2016 did have a significant impact on the robustness of the electoral process, as shown by the deft response to an unforeseen and formidable challenge: the COVID-19 pandemic. Much of the policy changes made in the wake of 2016 empowered information sharing and the acquisition of better election equipment, which made states' efforts to transition to processes that limit the spread of coronavirus significantly easier. Whereas elections in years past had struggled to respond to the ever changing landscape, the increased resources and awareness allowed jurisdictions to respond rapidly.

However, these rapid responses also created a new problem: the spread of misinformation. We conducted interviews with three election officials in Texas, a state that saw dramatic changes to its election policies in response to COVID-19, including extended early voting periods and modified rules around absentee voting [30]. All three interviewees indicated that the election itself went smoother than anticipated, but that misinformation proved to be a much more difficult issue to contend with. In an environment where election logistics were changing rapidly, and one where the spread of misinformation had already proven to be an effective attack vector, voters were bombarded with constantly changing and inconsistent information about when, where and how to vote. Worse, voters were also subject to conspiracy theories heralded at the highest level of government about the insecurity of voting systems and the illegitimacy of the election results [3].

Our interviews with elections officials helped us to better understand that worked and what did not work in Texas in 2020. We find that the additional resources and communications greatly bolstered the security and efficacy of elections. However, our interviewees also expressed great concern over the rise of misinformation, suggesting that while efforts to improve elections in Texas and the U.S. at large have had some success, significantly more work is needed.

The rest of this paper is structured as follows: in Section 2 we give a brief overview of elections in the United States, as well as the key policies that constrain them. In Section 3, we describe how and why we chose to interview election officials to shed light on the state of elections in the U.S.. We present case studies based on our interviews in Section 4, Section 5, and Section 6. Finally, we conclude in Section 7, synthesizing the knowledge our research has revealed and providing some recommendations for policy changes and future lines of research.

2 Elections in the United States

Elections in the United States are a complex system of multiple government entities with different degrees of jurisdiction, a small number of vendors, and a onerous regulatory regime. For brevity we choose to elide much of the complexity, and will try to provide just enough context to support the remainder of the paper.

2.1 Whose election is it anyway?

As the United States is a republic of states, its constitution delegates the vast majority of election responsibilities to the states, who in turn may delegate responsibilities down to local counties or townships. The exact degree of delegation varies widely between states; for example the states of Hawaii and Georgia largely run elections at a state level, handling most of the logistics and administration centrally while delegating the running of polling locations to counties. In contrast, states like Michigan and Wisconsin delegate almost all tasks to the local township level. Therefore, where responsibility for election security lies varies widely with each state.

In Texas, the state where our interviews were conducted, counties perform most of the election procedures. The state performs some regulatory tasks like deciding which type of voting equipment counties may purchase, distributes funding from the state and federal governments, and enforces policies about when and how elections may be conducted. However, county election officials ultimately have the power over what type of voting equipment to purchase (provided it is approved by the state) and therefore how voters are allowed to cast their ballots. Texas has a variety of voting technology, ranging from all-electronic DRE systems to hand-marked, hand-counted paper ballots. Most counties use a combination of voting technologies to accommodate absentee voting, voters with disabilities, and other considerations like the need for ballots in multiple languages.

County officials must also maintain significant information technology infrastructure, including: websites and social media accounts on which election information is distributed; email servers; and the technology required to maintain and program voting equipment, including voter information, ballot preparation and tabulation, and results reporting.

2.2 Help America Vote Act

Most of the requirements surrounding voting systems trace their lineage to the Help America Vote Act of 2002 (HAVA). HAVA was passed in response to the 2000 presidential election, in which a close contest ultimately came down to ballots cast on out-dated and poorly usable voting equipment in the state of Florida [16]. HAVA provided \$2 billion to states to upgrade their outdated voting equipment. HAVA also created several new regulations about voting equipment, including the Voluntary Voting System Guidelines (VVSG), requirements against which voting equipment can be certified. The VVSG includes performance and correctness requirements, as well as requirements that voting systems accommodate voters with disabilities.

HAVA is widely regarded as having brought direct-recording electronic voting machines (DREs) into popular use, as they were some of the only market-ready equipment at the time that could meet VVSG requirements for accessibility. In the years since HAVA's passing, DREs have fallen out of favor due to their insecurity, and been replaced with a wide array of technology, including hand-marked, optically scanned paper ballots and ballot marking devices [8]. Because

certification to VVSG standards is expensive and time-intensive, voting technology tends to lag behind modern technology standard. Until the passage of VVSG 2.0 this year (an update to the standard), the security of voting technology was an after thought as it was not required for certification.

HAVA is also the mechanism through which federal funding is made available for elections. In 2019 and 2020 two disbursements of HAVA money were made, with the first designed to improve election security [25] and the second designed to bolster states' responses to the COVID-19 pandemic as part of the CARES Act [14].

2.3 Elections in Texas

Texas is a state in the southern region of the United States, among one of the largest states with a population of approximately 29,145,505 based on 2020 estimates [34]. Texas has approximately 16,955,519 registered voters as of November 2020 and 8,745 precincts as of November 2018 [35].

The state does not have stringent requirements on type of voting equipment deployed to polling locations, however it does require that systems be certified by the U.S. Elections Assistance Commission. In practice, this means that systems in Texas are certified to the first version of the VVSG, which dates to 2005. This version of the VVSG contains little in terms of security requirements from voting systems. In addition to EAC certification, Texas also retains independent certification, such that a voting system may not be certified in Texas even if it is EAC certified.

The outdated standards and otherwise somewhat laissez faire environment means that Texas is one of the most diverse states in terms of voting equipment. Systems in Texas range from hand-marked, hand counted paper ballots to ballot-marking devices to DREs that have no paper record at all [35]. Texas also supports early voting, where voters can vote in a polling location several weeks before the official election day. Texas additionally supports absentee voting, however voters must qualify to vote absentee via a number of conditions, like disability, military status, or out-of-state residence at the time of the election.

2.4 The 2016 election and its aftermath

The 2016 election was one of the most contentious elections in recent memory. Misinformation and hacking campaigns were carried out by numerous foreign actors to attempt to sway the election, the most prominent being Russia [23]. Election officials around the country were often caught off guard and unable to respond to these attacks due to lack of resources, training, and communication (a fact which all three of our interviewees confirmed).

In response to the shortcomings of the 2016 election, the federal government designated elections as critical infrastructure, which provided additional support from the federal government to elections infrastructure, and established the Cybersecurity and Infrastructure Security Agency (CISA), tasked with aiding sectors like elections in improving their robustness to cyber attack. Additional

emphasis was placed on Information Sharing and Analysis Centers (ISACs) with the establishment of the EI-ISAC, channels through which information like critical security vulnerabilities and incidences could be disseminated to local elections officials [11].

In 2019 additional HAVA money was disbursed to allow jurisdictions to upgrade their aging voting equipment. All three of the election officials we spoke to had recently upgraded at least some of their voting equipment in part due to this additional funding.

In addition to the federal push to improve election security, the state of Texas passed two bills to improve the state's ability to respond to cyber attacks. Texas passed House Bill 8, the Texas Cybersecurity Act, in 2017, which "provides specific measures to protect sensitive and confidential data and maintain cyberattack readiness." The bill was passed partially in response to the state's being targeted in 2016 [21], but also due to the widespread increase in cyber attacks against the state in recent years [6]. Texas House Bill 9, the Texas Cybercrime Act, was passed as a companion bill that "updates the Texas Penal Code to recognize several new types of cybercrime and their punishments." These laws expand officials' roles to protect essential data for which they are responsible.

2.5 The COVID-19 pandemic

In response to the COVID-19 pandemic, most states rapidly pivoted their election infrastructure to de-emphasize in-person voting, believed to carry an increased risk of transmitting the disease. States greatly expanded voting by mail, extended voting hours, and procured extra equipment to be used in polling locations to limit the spread. An additional \$400 million was disbursed to states via HAVA [14]. In total, more voters voted through non-traditional means in 2020 than ever before [28].

However, while some states like Michigan, Arizona, and Wisconsin could pivot to sending out ballots by mail, some policies proved inflexible. The absentee ballot counting period in these states does not ordinarily start until election day, as historically most voters vote in-person and the proportion of vote-by-mail ballots is relatively small. Amidst a tsunami of mail in voting, however, these counting rules led to significant delays in results reporting on election night. These delays were fodder for misinformation campaigns, and helped feed into the narrative that the election results were illegitimate [12], a narrative that was picked up by elected officials in Texas and elsewhere [19].

3 Examining preparedness

In order to delve into the effectiveness of measures taken post-2016, as well as to assess how an election could be "the most secure in American history" while simultaneously resulting in political violence, we set out to interview election officials who were responsible for running the 2020 election. We solicited interviews with several, and conducted interviews with three election officials over seeing

County	Registered Voters	Type of in-person voting system	2020 Elections budget (USD)	Budget per registered voter
Harris	2,480,522	BMD	\$12,362,000	\$4.98
Bexar	1,189,373	BMD+DRE	\$4,278,082	\$3.60
Cameron	218,910	HMPB+BMD	\$1,498,560	\$6.85

Table 1. Case Study Counties in Texas with Details—Summary data for the three counties where we interviewed election officials. Harris is the largest county in Texas and second largest county in the United States. Bexar County comprises about half of Harris’s population, and Cameron County one tenth. All three counties spend similar amount of money per voter on elections.

medium- and large-sized counties in the state of Texas. Texas was chosen in part due to proximity to the researchers as well as it represents a good mix of election policies, voting equipment, and demographics in comparison to the United States as a whole [34,35]. Together, the election officials we interviewed oversee elections for 23% of voters in Texas [35].

We interviewed officials from Bexar, Cameron, and Harris counties (who chose to remain anonymous) with the hope of understanding how secure their voting systems are and whether they have the necessary resources to provide the best security for the voting machines. A summary of information about the counties we interviewed can be seen in Table 1.

The interviews we conducted were semi-structured, with a predetermined set of questions to prompt our interviewees, shown in Appendix A. We focused on four main areas with each official: their voting system, election preparation procedures, experiences in 2020, and an overall takeaway about the security of their election system. All interviewees signed a consent form and clarified whether they wished to remain anonymous.

4 Case Study: Bexar County

With San Antonio as the county seat, Bexar County is the fourth most populated county in Texas and the 16th most populated in the United States. The county’s election department is led by Elections Administrator Jacquelyn F. Callanen, a non-partisan candidate appointed by the election commission, including county judge, clerk, and more. Like other counties, “[t]he Bexar County Elections Department is responsible for voter registration activities and election operations throughout Bexar County.”

We interviewed a Bexar County official who answered questions regarding election security and voting systems used in the County. During the interview with the County official, we asked questions about the voting processes and procedures. Below we provide a summary of the information provided to us by the official, along with quotes where relevant.

Voting equipment Prior to the 2020 election, Bexar County used Direct-Recording Electronic (DRE) as its voting system for 17 years. In November 2019, Bexar County finally decided to upgrade its voting system to what they now call blended or hybrid system purchased from ES&S. While a “hybrid” system has colloquially become known as a ballot marking and tabulation all-in-one device [37], Bexar County’s system is a separate BMD and scanner for most voters. Texas also supports “curbside” voting, where voters who cannot enter a polling place due to a disability vote on a machine brought to the vehicle that transported them to the polling location. Bexar County’s new system includes paperless DREs for curbside voters [35].

Election preparation Bexar County’s election preparation process differs from the other counties because it services the three military bases. Voters who are in the military or overseas fall under a different set of regulations than most voters: the Uniformed and Overseas Citizens Absentee Voting Act (UOCAVA) [32]. Forty-five days before the election, UOCAVA requires that the County elections department to send all the military ballots to the bases, in addition to sending absentee ballots to UOCAVA voters overseas. Failure to do so on time results in the county covering the ballots’ delivery and returning costs.

Because of this early deadline, the interviewee indicated that preparations for the election begin six months before election day. Early preparation includes equipment testing and inspection of other systems like voter registration. During this period, the election administrator holds weekly meetings with staff to brief on all critical functions that must be completed before the election.

Experiences in 2020 Despite all of the changes incurred by rolling out a new voting system, dealing with the COVID-19 pandemic, and rampant misinformation campaigns led by foreign adversaries [22], the official we spoke with in Bexar County felt that the election overall went smoothly:

I am unbelievably proud of my team and how well we worked in November. . . CISA made us essential workers which means that we spent over 90 hours in the office together. . . [we witnessed] the most voters, mail-in ballots, the governor extended early voting which helped. The County paid for all election officials to test for COVID-19. . . spent about half-million dollars to ensure everyone is protected.

The official alleged that in 2016, the Texas Secretary of State’s (SOS) system was hacked. However, the then-Secretary of State denied that the state was a victim of the Russian hacks, asking the DHS to make corrections of the list they published [21]. Nevertheless, Bexar county opted to develop new protocols to provide better defense against cyberattacks.

We have written protocols to protect ourselves in case the SOS gets hacked. We now have a complete shadow election set up that is moved

offsite and double locked. Should something happen, we will still be able to get elections results. Prior to 2016, we never thought of this. The climate changed a little bit.

The interviewee also indicated that better communication was a boon to preparing for this election, citing both CISA’s role in the 2020 election as well as Texas’s emergency operations center, stating that they “[w]ork hand-in-hand with them and see if we can remediate anything from the back-end.”

Despite these changes, Bexar County experienced an unexpected problem during the 2020 election that left voters confused and contributed to misinformation spreading across the jurisdiction. In contrast with their old DRE system the new system sees voters handling paper ballots and placing them in the tabulator; however, no one was aware that those boxes could only hold 2,000 ballots. Election workers at the precincts were under the assumption that the tabulator was jammed when the tabulators filled up and they could not push through ballots. Voters began thinking that the tabulators were not working, and misinformation circulated rapidly within the county, to the extent that voters started leaving the polling locations without having voted. The official we interviewed explained that “everybody focused on hardware and software. No one informed us of the number of ballots held by the tabulator.”

As the administrator only had five election technology specialists working to cover 45 polling locations, they had to bring extra technicians from other departments to change the tabulators. Fortunately this issue was mitigated slightly by the changes to early voting rules to accommodate the pandemic, as during early voting voters can go to any polling location. This spread out the load on individual locations and resulted in tabulators filling up less frequently.

Takeaways Overall, the official we spoke with agreed that Bexar County is well-equipped to defend the voting systems and voter information from cyberattacks. The official cited their newly adopted policies, annual voting systems inspection by the vendor, the ability to tabulate absentee ballots earlier than election day, the use of electronic pollbooks; as well as in-house databases, VPNs, and password policies, and that “the voting system is encrypted” with an “encryption method signed off by [the] federal government and State of Texas.”¹

5 Case Study: Cameron County

Cameron County is located in the southernmost part of Texas, bordered to the south by Mexico and to the east by the Gulf of Mexico, with Brownsville as the county seat. Cameron County has approximately 218,910 registered voters in approximately 102 precincts.

¹ As ES&S has never submitted their systems for independent review, and the standards used to certify the system used in Bexar county only requires encryption for electronic transmission [31], we could not verify these claims about encryption.

Voting equipment Cameron County uses hand-marked paper ballots with ballot marking devices for accessibility, and an optical scanner for tabulation. They follow a precinct count voting system where the ballots are tabulated at the polling location [35]. Our interviewee indicated that the county uses two backup methods for voting counts and results to avoid data loss, including utilizing a USB drive and an internal record. The presiding judge brings the copy of the counts to the central location, and if an issue arises, an individual is sent to collect a copy of the USB drive and machines.

Election preparation Expounding on procedures around the voting equipment, the official explained that there is ongoing security surveillance of the County IT system. When setting up a machine for the next election, the machine's state is verified to ensure that nothing has changed while the machine was in storage. Then, machines are programmed with data for the upcoming election over an Internet-connected virtual private network, and security patches are installed. (This manner of connection adds risk of compromise due to Internet exposure.) A week before the start of early voting, the public is invited to view logic and accuracy testing, where machines are tested to ensure the election programming is correct.

For other election-related procedures, like communication, transferring voter data, and aggregating tabulation results, the interviewee reported the use of VPNs, FTP, and software access controls for all software that is used. Additionally, the official indicated that the elections warehouse was physically secured. While they reported not using encryption on their office systems, they encrypt data transferred to locations with no additional security layer.

Experiences in 2020 After 2016, the interviewee reported that there were significant concerns regarding the security of Cameron County elections and voting systems, including a past data breach. However, they noted significant improvements to their security posture. Funding from the state and grant money from two outside entities allowed for better security of their voting equipment. They also noted a precipitous increase in cybersecurity training and awareness, and established processes for handling phishing attacks and other suspicious incidences. The interviewee also cited better communication between federal, state, and local officials, “with the [cybersecurity] training, we all started speaking the same language and understand each other.”

Takeaways When asked how prepared Cameron County was to face cyber threats, the official responded by stating, “[e]very election offers its challenges. COVID-19 was a challenge for us, and the machines did not have any problems. Using the paper-based system, COVID did not have as much impact.” Overall, the interviewee expressed confidence that Cameron County did provide adequate security for its voting systems, but identified misinformation as a major threat:

[We will] try our best to make sure that information [that we share with the public] is legitimate. After 2020, misinformation is becoming a danger to our elections and we need to provide additional training [to the public and staff] to recognize it. The awareness of the threat has increased since 2016. The more information we have, the better we can protect ourselves and our democracy.

6 Case Study: Harris County

Harris County is the largest county in Texas, with an estimated population of over 4,713,325. The County’s election department is under the leadership of County Judge Lina Hidalgo and elections administrator Isabel Longoria. According to VerifiedVoting, Harris County had over 2,480,522 registered voters as of November 2020, with an estimate of 1,012 precincts [35].

We interviewed Michael Winn, the Harris County Chief Deputy Administrator, who has over 25 years of experience working in the government. Mr. Winn agreed to be identified in our paper. He has worked in Bexar, Travis, and Harris Counties in the elections department and is on the Election Assistance Commission Board of Advisors. He was also one of the election officials who contributed to the development of STAR-Vote, an end-to-end cryptographic voting system [5].

Voting equipment Harris County has used DREs as their primary voting system since 2001, including during the 2020 general election, although they are currently transitioning to an all-BMD system for in-person voters.

Election preparation When preparing for elections, Mr. Winn identified three preparation windows, which include 45, 60, and 90 days before elections. During these periods, they conduct hash code testing to verify whether the code matches a hash code that was previously taken. If they do not, they choose to replace the system due to the possibility of the system being tampered with. They also perform the logic and accuracy testing to ensure that machines are programmed correctly for the upcoming election.

Forty-five days before the elections is known as the “lockdown period”, where the system is air-gapped. During the preparation period, the county involves its partners to check that their information such as addresses, contact and more is correct, including schools, political parties, and polling places. Per Mr. Winn, this process, known as entity proofing, is a vital means of ensuring the correctness of public information.

During the interview, we asked Mr. Winn how the County deals with equipment security. He explained that “if there are updates [to the voting system software], we do communicate with vendors. . . we do get the updates and when completed, there is a file, a record that shows that there has been an update and it outlines the details of it including dates and times. Record is sent to the state [the Secretary of State], and they have a version of the last update.” He reported

that the county uses encryption to protect their systems and voter information. To backup votes or results, Harris County performs audits during tabulation; with the old DRE system, this involved printing periodic results tapes from the machines to compare with other data collected during the election.

According to Mr. Winn, Harris County has a team of information technology (IT) specialists who provide the election officials updated on their voting equipment status and any critical information they need to know daily. The county also ensures that education and training are provided for every employee and encourages them to become certified in election security. They also have a program such as the automatic shutdown of programs when the computers experience inactivity, mitigating insider threat.

Experiences in 2020 When we asked Mr. Winn how the 2020 presidential elections differed from 2016, he responded, saying that "there was more communication between CISA, DHS, FBI. Continuation efforts to make sure that county officials become a part of the decision-making of elections. The government was keeping information to themselves [in 2016]; part was to avoid vulnerability. In 2020, there was the inclusion of all officials."

Takeaways As Harris County is Texas's largest county, one might assume that they have all the necessary tools and are well equipped to secure elections and electoral infrastructure. However, when posed with the question of whether Harris County is well equipped to protect the voting systems and voter information from cyberattacks, Mr. Winn's response was no different from the other officials we interviewed: "nothing is guaranteed, our county does a good job of making sure that we stay current and make sure we have the best system in place. We just purchased a new voting system that will be used for the first time in May 2021."

7 Conclusion

We set out to understand two incongruous facts: the 2020 election overall appears to have run smoothly and largely without incident, and yet it has led to some of the most tumultuous political discourse in recent memory in the U.S. We provided some context for elections in the United States, and then performed interviews with three election officials in the state of Texas to get a better picture on the ground. All three of our interviewees echoed that the election went well, and all three also indicated that the elections could have gone better from a security stand point.

What went well All three interviewees noted that there was significantly better communication at all levels of government about cyberthreats. Even in the presence of active threats [22], officials were able to move quickly to shore up defenses and quash any issues that might affect voters. Significant policy changes

and an increase in resourcing at all levels of government post-2016 also enabled a much more nimble response to the COVID-19 pandemic. Voters in Texas voted in unprecedented numbers in 2020, including a nearly 10-point increase in turnout and record high numbers of absentee and early voting [10, 36].

Improved communication at all levels of government played a key role in making the 2020 elections some of the smoothest and most secure in history, according to our interviewees. Federal initiatives like the founding of CISA and the EI-ISAC opened channels of communication between elections officials and intelligence officials that was pinpointed as a major problem in 2016 [17].

What went poorly Despite the major improvements, the 2020 election was not without its flaws. Issues ranging from delays in results reporting [15] to rampant misinformation campaigns still hindered public confidence in the election outcome [24]. Misinformation about results reporting flourished even despite serious efforts to communicate about the expected delays [26]. One of our interviewees saw firsthand how a run-of-the-mill problem in an election, like their scanners filling up with ballots, could lead to misinformation that ultimately led people to walk away from voting. All three of our interviewees noted that misinformation is one of the tallest hurdles to U.S. elections moving forward.

Unfortunately, misinformation is already having a significant impact on elections in Texas and the U.S. at large. The Texas Senate recently proposed more restrictive voting laws that would dramatically change the voting landscape in Texas [33], largely in response to misinformation that has spread after the election [2]. Many other states are considering similar laws, as well as withstanding partisan efforts to attempt to overturn the 2020 election outcome [1].

Takeaways The U.S. has made significant strides to improve its election security post-2016. Improved training and resources to election officials, improved communication between government entities, and improved processes all made 2020 a much smoother election than 2016. Practices long heralded by the election security community, like risk-limiting audits [18] and paper ballots [27], are seeing widespread adoption [20].

Despite these successes, misinformation continue to run wild, spurring legislative action [33], partisan campaigns to undermine elections [1], and leading to violence [4]. Research in combating misinformation is in its relative infancy, but our key takeaway from our discussions with election officials is that it is the single most important election security issue right now. Prior election security efforts have focused on providing voters with evidence, ranging from end-to-end encrypted voting systems to plain paper ballots and transparent counting [7]. However, it appears that merely *providing* this evidence is not enough: we need to be proactive in *using* it to convince voters that the election outcomes are secure.

Acknowledgements We would like to thank the election officials who agreed to speak with us. Their courage and insights helped us paint a clearer picture of the landscape of elections in the United States.

References

1. Anglen, R., Randazzo, R.: Arizona Senate considers expanding audit of Maricopa County ballots to all races. <https://www.azcentral.com/story/news/local/arizona-investigations/2021/05/14/arizona-senate-considers-expanding-audit-maricopa-county-ballots-all-races/5100735001/> (2021)
2. Astor, M.: A Perpetual Motion Machine’: How Disinformation Drives Voting Laws. <https://www.nytimes.com/2021/05/13/us/politics/disinformation-voting-laws.html> (2021)
3. Barrett, T., Raju, M., Foran, C.: Top Republicans defend Trump on baseless voter fraud claims as concerns grow in the ranks. <https://www.cnn.com/2020/11/05/politics/election-2020-congressional-republicans-trump-election-fraud/index.html> (2020)
4. Barry, D., McIntire, M., Rosenberg, M.: ‘Our President Wants Us Here’: The Mob That Stormed the Capitol. *New York Times* (2021), accessed from <https://www.nytimes.com/2021/01/09/us/capitol-rioters.html>
5. Bell, S., Benaloh, J., Byrne, M.D., DeBeauvoir, D., Eakin, B., Kortum, P., McBurnett, N., Pereira, O., Stark, P.B., Wallach, D.S., Winn, M.: STAR-Vote: A secure, transparent, auditable, and reliable voting system. In: 2013 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 13) (2013)
6. Benton, Jackie: Cyberdefense for Texas State Government. <https://comptroller.texas.gov/economy/fiscal-notes/2019/mar/tx-cyberdefense.php> (2019)
7. Bernhard, M., Benaloh, J., Halderman, J.A., Rivest, R.L., Ryan, P.Y., Stark, P.B., Teague, V., Vora, P.L., Wallach, D.S.: Public evidence from secret ballots. In: International Joint Conference on Electronic Voting. pp. 84–109. Springer (2017)
8. Bernhard, M., McDonald, A., Meng, H., Hwa, J., Bajaj, N., Chang, K., Halderman, J.A.: Can voters detect malicious manipulation of ballot marking devices? In: 2020 IEEE Symposium on Security and Privacy (SP). pp. 679–694. IEEE (2020)
9. Bruer, W., Perez, E.: Officials: Hackers breach election systems in illinois, arizona. <https://www.cnn.com/2016/08/29/politics/hackers-breach-illinois-arizona-election-systems> (2016)
10. Cai, M.: At least 9.7 million Texans — 57% of registered voters — voted early. <https://apps.texastribune.org/features/2020/texas-early-voting-numbers/> (2020)
11. Center for Internet Security: Elections infrastructure information sharing and communication (ei-isac). <https://www.cisecurity.org/ei-isac/> (2021)
12. Chen, E., Chang, H., Rao, A., Lerman, K., Cowan, G., Ferrara, E.: COVID-19 misinformation and the 2020 US presidential election. *The Harvard Kennedy School Misinformation Review* (2021)
13. Cybersecurity and Infrastructure Security Agency: Joint Statement from Elections Infrastructure Government Coordinating Council & the Election Infrastructure Sector Coordinating Executive Committees. <https://www.cisa.gov/news/2020/11/12/joint-statement-elections-infrastructure-government-coordinating-council-election> (2020)
14. Election Assistance Commission: 2020 CARES Act Grants . <https://www.eac.gov/payments-and-grants/2020-cares-act-grants> (2020)
15. Henley, J., Sullivan, H., McCarthy, T.: When will we know the US election result – and why the delay? <https://www.theguardian.com/us-news/2020/nov/06/when-will-we-know-the-us-election-result-and-why-the-delay> (2020)
16. Jones, D., Simons, B.: Broken ballots: Will your vote count? CSLI Publications Stanford (2012)

17. Kamarck, E.: The federal-state disconnect in securing the 2016 election and how not to repeat it. <https://www.brookings.edu/blog/fixgov/2019/08/23/the-federal-state-disconnect-in-securing-the-2016-election-and-how-not-to-repeat-it/> (2019)
18. Lindeman, M., Stark, P.B.: A gentle introduction to risk-limiting audits. *IEEE Security & Privacy* **10**(5), 42–49 (2012)
19. Livingston, A., Mulcahy, S.: As states count votes, some of Texas' most prominent Republican politicians are spreading misinformation about the election. <https://www.texastribune.org/2020/11/06/texas-republicans-trump-results/> (2020)
20. McCadney, A., Howard, E., Norden, L.: Voting machine security: Where we stand six months before the new hampshire primary. Brennan Center for Justice. Retrieved from <https://www.brennancenter.org/our-work/analysis-opinion/voting-machine-security-where-we-stand-six-months-new-hampshire-primary> (2019)
21. Najmabadi, S.: Texas denies state was target of election-related hacking by Russia. <https://www.texastribune.org/2017/09/29/texas-denies-it-was-target-election-related-hacking/> (2017)
22. National Intelligence Council: Foreign Threats to the 2020 US Federal Elections. <https://www.dni.gov/files/ODNI/documents/assessments/ICA-declass-16MAR21.pdf> (2021)
23. Office of the Director of National Intelligence: Assessing Russian Activities and Intentions in Recent US Election. https://www.dni.gov/files/documents/ICA_2017_01.pdf (2017)
24. Ognyanova, K., Lazer, D., Robertson, R.E., Wilson, C.: Misinformation in action: Fake news exposure is linked to lower trust in media, higher trust in government when your side is in power. *Harvard Kennedy School Misinformation Review* (2020)
25. Parks, M.: Congress Allocates \$425 Million For Election Security In New Legislation (2019)
26. Riccardi, N.: AP Explains: The election result may be delayed. That's OK. <https://apnews.com/article/election-2020-biden-trump-delayed-result-d9208787554db4c4575579f6b75a7cde> (2020)
27. Rivest, R.: On the notion of 'software independence' in voting systems. *Phil. Trans. R. Soc. A* **366**(1881), 3759–3767 (October 2008)
28. Scherer, Z.: Majority of Voters Used Nontraditional Methods to Cast Ballots in 2020. <https://www.census.gov/library/stories/2021/04/what-methods-did-people-use-to-vote-in-2020-election.html> (2021)
29. Sprunt, B.: Here are the republicans who objected to the electoral college count. <https://www.npr.org/sections/insurrection-at-the-capitol/2021/01/07/954380156/here-are-the-republicans-who-objected-to-the-electoral-college-count> (2021)
30. of State, T.S.: Covid-19 resources for election officials. <https://www.sos.state.tx.us/elections/covid/index.shtml> (2020)
31. United States Election Assistance Commission: The Voluntary Voting System Guidelines 1.0. <https://www.eac.gov/voting-equipment/voluntary-voting-system-guidelines> (2005), accessed both volumes on 14 May 2021.
32. The Uniform and Overseas Citizens Absentee Voting Act, text found on fvap.gov on 14 May 2021.
33. Ura, A.: Here's how Texas elections would change, and become more restrictive, under the bill Texas Republicans are pushing. <https://www.texastribune.org/2021/04/21/texas-voting-restrictions-senate-bill-7/> (2021)
34. U.S. Census Bureau: QuickFacts Texas; United States. <https://www.census.gov/quickfacts/fact/table/TX,US/PST045219>, Accessed 13 May 2021

35. Verified Voting Foundation: The verifier. <https://verifiedvoting.org/verifier/>, accessed on 13 May 2021
36. Wallace, J.: Texas voter turnout was best in almost 30 years. <https://www.houstonchronicle.com/news/election2020/article/Texas-voter-turnout-was-best-in-almost-30-years-15705990.php> (2020)
37. Wilkie, J.: America's new voting machines bring new fears of election tampering. <https://www.theguardian.com/us-news/2019/apr/22/us-voting-machines-paper-ballots-2020-hacking> (2019)

A Interview Prompts

- Can you please tell me a little bit about yourself and your responsibilities at Bexar County?
- What are the sources for voting processes and procedures do you have for your region?
- What type of voting system is used in your precinct?
- When does your precinct start preparing the elections? Finding and taking care of vulnerabilities?
- What process does or did your county take to prepare for the 2020 elections? How was your county's process different from the 2016 election for the 2020 election?
- What steps do you follow to test your equipment prior to elections?
- What methods do you use to backup voting counts or results?
- What is your process to deal with equipment security?
- What procedures do you employ to protect voter information?
- In the 2020 elections, did you encounter any specific problems with your voting machines? If yes, what were the problems?
 - What was done to resolve the problems found in the systems? Is there a procedure in place for this?
 - After resolving the problems, what did you learn from this? What should have or needs to be done better?
- Do you have a special organization that deals with election security? Can you tell me about it?
- Overall, would you say that your county is well equipped to secure the voting systems and voter information from cyberattacks?
 - Please explain.
 - If not, what is needed in order to better safeguard voting systems or voter information?

Risk Limiting Audits

RiLACS: Risk Limiting Audits via Confidence Sequences

Ian Waudby-Smith¹[0000-0003-4471-1405], Philip B. Stark²[0000-0002-3771-9604],
and Aaditya Ramdas¹[0000-0003-0497-311X]

¹ Carnegie Mellon University, Pittsburgh, PA, USA {ianws, aramdas}@cmu.edu

² University of California, Berkeley, Berkeley, CA, USA stark@stat.berkeley.edu

Abstract. Accurately determining the outcome of an election is a complex task with many potential sources of error, ranging from software glitches in voting machines to procedural lapses to outright fraud. Risk-limiting audits (RLA) are statistically principled “incremental” hand counts that provide statistical assurance that reported outcomes accurately reflect the validly cast votes. We present a suite of tools for conducting RLAs using confidence sequences — sequences of confidence sets which uniformly capture an electoral parameter of interest from the start of an audit to the point of an exhaustive recount with high probability. Adopting the SHANGRLA [13] framework, we design nonnegative martingales which yield computationally and statistically efficient confidence sequences and RLAs for a wide variety of election types.

Keywords: Martingales · sequential hypothesis tests · SHANGRLA

1 Introduction

The reported outcome of an election may not match the validly cast votes for a variety of reasons, including software configuration errors, bugs, human error, and deliberate malfeasance. Trustworthy elections start with a trustworthy paper record of the validly cast votes. Given access to a trustworthy paper trail of votes, a risk-limiting audit (RLA) can provide a rigorous probabilistic guarantee:

1. If an initially announced assertion \mathcal{A} about an election is *false*, this will be corrected by the audit with high probability;
2. If the aforementioned assertion \mathcal{A} is *true*, then \mathcal{A} will be confirmed (with probability one).

Here, an electoral assertion \mathcal{A} is simply a claim about the aggregated votes cast (e.g. “Alice received more votes than Bob”). An auditor may wish to audit several claims: for example, whether the reported winner is correct or whether the margin of victory is as large as announced.

From a statistical point of view, efficient risk-limiting audits can be implemented as sequential hypothesis tests. Namely, one tests the null hypothesis H_0 :

Waudby-Smith et al.

“the assertion \mathcal{A} is false,” versus the alternative H_1 : “the assertion \mathcal{A} is true”. Imagine then observing a random sequence of voter-cast ballots X_1, X_2, \dots, X_N , where N is the total number of ballots. A sequential hypothesis test is represented by a sequence $(\phi_t)_{t=1}^N$ of binary-valued functions:

$$\phi_t := \phi(X_1, \dots, X_t) \mapsto \{0, 1\},$$

where $\phi_t = 1$ represents rejecting H_0 (typically in favor of H_1), and $\phi_t = 0$ means that H_0 has not yet been rejected. The sequential test (and thus the RLA) stops as soon as $\phi_t = 1$ or once all N ballots are observed, whichever comes first. The “risk-limiting” property of RLAs states that if the assertion is false (in other words, if H_0 holds), then

$$\mathbb{P}_{H_0}(\exists t \in \{1, \dots, N\} : \phi_t = 1) \leq \alpha,$$

which is equivalent to type-I error control of the sequential test. Another way of interpreting the above statement is as follows: if the assertion is incorrect, then with probability at least $(1 - \alpha)$, $\phi_t = 0$ for every $t \in \{1, \dots, N\}$ and hence all N ballots will eventually be inspected, at which point the “true” outcome (which is the result of the full hand count) will be known with certainty.

1.1 SHANGRLA Reduces Election Auditing to Sequential Testing

Designing the sequential hypothesis test $(\phi_t)_{t=1}^N$ depends on the type of vote, the aggregation method, or the social choice function for the election, and thus past works have constructed a variety of tests. Some works have designed $(\phi_t)_{t=1}^N$ in the context of a particular type of election [6,7,9]. On the other hand, the “SHANGRLA” (**S**ets of **H**alf-**A**verage **N**ulls **G**enerate **R**LA) framework unifies many common election types including plurality elections, approval voting, ranked-choice voting, and more by reducing each of these to a simple hypothesis test of whether a finite collection of finite lists of bounded numbers has mean μ^* at most $1/2$ [13,1]. Let us give an illustrative example to show how SHANGRLA can be used in practice.

Suppose we have an election with two candidates, Alice and Bob. A ballot may contain a vote for Alice or for Bob, or it may contain no valid vote, e.g., because there was no selection or an overvote. It is reported that Alice and Bob received N_A and N_B votes respectively with $N_A > N_B$ and that there were a total of N_I invalid ballots for a total of $N = N_A + N_B + N_I$ voters. We encode votes for Alice as “1”, votes for Bob as “0” and invalid votes as “1/2”, to obtain a set of numbers $\{x_1, x_2, \dots, x_N\}$. Crucially, Alice indeed received more votes than Bob if and only if $\mu^* := \frac{1}{N} \sum_{i=1}^N x_i > 1/2$. In other words, *the report that Alice beat Bob can be translated into the assertion that $\mu^* \in (1/2, 1]$.*

SHANGRLA proposes to audit an assertion by testing its complement: rejecting that “complementary null” is affirmative evidence that the assertion is indeed true. In other words, if one can ensure that X_1, X_2, \dots, X_N is a random permutation of $\{x_1, \dots, x_N\}$ by sampling ballots without replacement (each ballot

is chosen uniformly amongst remaining ballots), then we can concern ourselves with designing a hypothesis test $(\phi_t)_{t=1}^N$ to test the null $H_0 : \mu^* \leq 1/2$ against the alternative $H_1 : \mu^* > 1/2$.

One of the major benefits of SHANGRLA is the ability to reduce a wide range of election types to a testing problem of the above form. This permits the use of powerful statistical techniques which were designed specifically for such testing problems (but may not have been designed with RLAs in mind). Throughout this paper, we adopt the SHANGRLA framework, and while we return to the example of plurality elections for illustrative purposes, all of our methods can be applied to any election audit which has a SHANGRLA-like testing reduction [13].

1.2 Confidence Sequences

In the fixed-time (i.e. non-sequential) hypothesis testing regime, there is a well-known duality between hypothesis tests and confidence intervals for a parameter μ^* of interest. We describe this briefly for $\mu^* \in [0, 1]$ for simplicity. For each $\mu \in [0, 1]$, suppose that $\phi^\mu \equiv \phi^\mu(X_1, \dots, X_n) \mapsto \{0, 1\}$ is a level- α nonsequential, fixed-sample test for the hypothesis $H_0 : \mu^* = \mu$ versus $H_1 : \mu^* \neq \mu$. Then, a nonsequential, fixed-sample $(1 - \alpha)$ confidence interval for μ^* is given by the set of all $\mu \in [0, 1]$ for which ϕ^μ does not reject, that is $\{\mu \in [0, 1] : \phi^\mu = 0\}$.

As we discuss further in Section 2, an analogous duality holds for sequential hypothesis tests and time-uniform *confidence sequences* (here and throughout the paper, “time” is used to refer to the number of samples so far, and need not correspond to any particular units such as hours or seconds). We first give a brief preview of the results to come. Consider a family of sequential hypothesis tests $\{(\phi_t^\mu)_{t=1}^N\}_{\mu \in [0, 1]}$, meaning that for each μ , $(\phi_t^\mu)_{t=1}^N$ is a sequential test for μ . Then, the set of all μ for which $\phi_t^\mu = 0$,

$$C_t := \{\mu \in [0, 1] : \phi_t^\mu = 0\}$$

forms a $(1 - \alpha)$ *confidence sequence* for μ^* , meaning that

$$\mathbb{P}(\exists t \in [N] : \mu^* \notin C_t) \leq \alpha,$$

where $[N]$ is used to denote the set $\{1, 2, \dots, N\}$. In other words, C_t will cover μ^* at *every single* time t , except with some small probability $\leq \alpha$. Since C_t is typically an interval $[L_t, U_t]$, we call the lower endpoint $(L_t)_{t=1}^N$ as a lower confidence sequence (and similarly for upper).

In particular, given the sequential hypothesis testing problem that arises in SHANGRLA, we can cast the RLA as a sequential estimation problem that can be solved by developing confidence sequences (see Figure 1).³ As we will see in Section 2, our confidence sequences provide added flexibility and an intuitive visualizable interpretation for SHANGRLA-compatible election audits, without sacrificing any statistical efficiency.

³ Code to reproduce all plots can be found at github.com/wannabesmith/RiLACS.

Waudby-Smith et al.

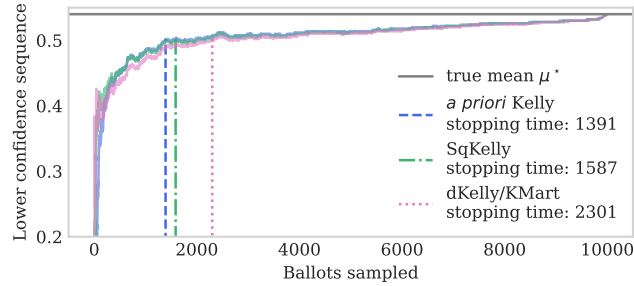


Fig. 1. 95% Lower confidence sequences for the margin of a plurality election between Alice and Bob for three different auditing methods. Votes for Alice are encoded by “1” and those for Bob are encoded by “0”. The parameter of interest is then the average of these votes, which in this particular example is 54% (given by the horizontal grey line). The outcome is verified once the lower confidence sequence exceeds $1/2$. The time at which this happens is given by the vertical blue, green, and pink lines.

1.3 Contributions and Outline

The contributions of this work are twofold. First, we introduce confidence sequences to the election auditing literature as intuitive and flexible ways of interpreting and visualizing risk-limiting audits. Second, we present algorithms for performing RLAs based on confidence sequences by deriving statistically and computationally efficient nonnegative martingales. At the risk of oversimplifying the issue, modern RLAs face a computational-statistical efficiency tradeoff. Methods such as BRAVO are easy to compute, but potentially less statistically efficient than the current state-of-the-art, KMart [13], but KMart can be prohibitively expensive to compute for large elections. The methods presented in this paper resolve this tradeoff: they typically match or outperform both BRAVO and KMart, while remaining practical to compute in large elections.

In Section 2, we show how confidence sequences generate risk-limiting audits, how they relate to more familiar RLAs based on sequentially valid p -values, and how they can be used to audit multiple contests. Section 3 derives novel confidence sequence-based RLAs and compares them to past RLA methods via simulation. Finally, Section 4 discusses how all of the aforementioned results apply to risk-limiting tallies for coercion-resistant voting schemes.

2 Confidence Sequences are Risk-Limiting

Consider an election consisting of N ballots. Following SHANGRLA [13], suppose that these can be transformed to a set of $[0, u]$ -bounded real numbers $x_1, \dots, x_N \in [0, u]$ with mean $\mu^* := \frac{1}{N} \sum_{i=1}^N x_i$ for some known $u > 0$. Suppose that electoral assertions can be made purely in terms of μ^* . A classical $(1 - \alpha)$ confidence interval CI_n for μ^* is an interval computed from data X_1, X_2, \dots, X_n with the

Risk Limiting Audits via Confidence Sequences

guarantee that

$$\forall n \in [N], \mathbb{P}(\mu^* \in \text{CI}_n) \geq 1 - \alpha.$$

In contrast, a $(1 - \alpha)$ *confidence sequence* for μ^* is a sequence of confidence sets, C_1, C_2, \dots, C_N which all simultaneously capture μ^* with probability at least $(1 - \alpha)$. That is,

$$\underbrace{\mathbb{P}(\forall t \in [N], \mu^* \in C_t) \geq 1 - \alpha}_{\text{simultaneous coverage probability}}, \quad \text{or equivalently} \quad \underbrace{\mathbb{P}(\exists t \in [N] : \mu^* \notin C_t) \leq \alpha}_{\text{error probability}}.$$

The two probabilistic statements above are equivalent, but provide a different way of interpreting α and the corresponding guarantee.

If we have access to a $(1 - \alpha)$ confidence sequence for μ^* , we can audit any assertion about the election outcome made in terms of μ^* with risk limit α . Here, we use $\mathcal{A} \subseteq [0, u]$ to denote an assertion. For example, SHANGRLA typically uses assertions of the form “ μ^* is greater than $1/2$ ”, in which case $\mathcal{A} = (1/2, u]$.

Algorithm 1.1: Risk limiting audits via confidence sequences (RiLACS)

```

Input: Assertion  $\mathcal{A} \subseteq [0, u]$ , risk limit  $\alpha \in (0, 1)$ .
for  $t \in [N]$  do
  Randomly sample and remove  $X_t$  from the remaining ballots.
  Compute  $C_t \equiv C(X_1, \dots, X_t)$  at level  $\alpha$ .
  if  $\mathcal{A} \subseteq C_t$  then
    Certify the assertion  $\mathcal{A}$  and stop if desired.
  end if
end for

```

If the goal is to finish the audit as soon as possible above all else, then one can ignore the “if desired” condition. However, continued sampling can provide added assurance in \mathcal{A} , and maintains the risk limit at α . The following theorem summarizes the risk-limiting guarantee of the above algorithm.

Theorem 1. *Let $(C_t)_{t=1}^N$ be a $(1 - \alpha)$ confidence sequence for μ^* . Let $\mathcal{A} \subseteq [0, u]$ be an assertion about the electoral outcome (in terms of μ^*). The audit mechanism that certifies \mathcal{A} as soon as $C_t \subseteq \mathcal{A}$ has risk limit α .*

Proof. We need to prove that if $\mu^* \notin \mathcal{A}$, then $\mathbb{P}(\exists t \in [N] : C_t \subseteq \mathcal{A}) \leq \alpha$. First, notice that if $C_t \subseteq \mathcal{A}$, then we must have that $\mu^* \notin C_t$ since $\mu^* \notin \mathcal{A}$. Then,

$$\begin{aligned} \mathbb{P}(\exists t \in [N] : C_t \subseteq \mathcal{A}) &\leq \mathbb{P}(\exists t \in [N] : \mu^* \notin C_t) \\ &\leq \alpha, \end{aligned}$$

where the second inequality follows from the definition of a confidence sequence. This completes the proof. \square

Let us see how this theorem can be used in an example. Consider an election with two candidates, Alice and Bob, and a total of N cast ballots. Let $\{x_1, \dots, x_N\}$

Waudby-Smith et al.

be the list of numbers that result from encoding votes for Alice as 1, votes for Bob as 0, and ballots that do not contain a valid vote as $1/2$. Let $(C_t)_{t=1}^N$ be a $(1 - \alpha)$ confidence sequence for $\mu^* := \frac{1}{N} \sum_{i=1}^N x_i$. If we wish to audit the assertion that “Alice beat Bob”, then $u = 1$ and $\mathcal{A} = (1/2, 1]$. We can sequentially sample X_1, X_2, \dots, X_N without replacement, certifying the assertion once $C_t \subseteq \mathcal{A}$. By Theorem 1, this limits the risk to level α .

2.1 Relationship to Sequential Hypothesis Testing

The earliest work on RLAs did not use anytime p -values [10,11], but since about 2009, most RLA methods have used anytime p -values to conduct sequential hypothesis tests [12,8,7,13,3]. An anytime p -value is a sequence of p -values $(p_t)_{t=1}^N$ with the property that under some null hypothesis H_0 ,

$$\mathbb{P}_{H_0}(\exists t \in [N] : p_t \leq \alpha) \leq \alpha. \quad (1)$$

The anytime p -values $p_t \equiv p_t(\mu)$ are typically defined implicitly for each null hypothesis $H_0 : \mu^* = \mu$ and yield a sequential hypothesis test $\phi_t^\mu := \mathbb{1}(p_t(\mu) \leq \alpha)$. As alluded to in Section 1.2, this immediately recovers a confidence sequence:

$$C_t := \{\mu \in [0, u] : \phi_t^\mu = 0\}.$$

Notice in Figure 2 that the times at which nulls are rejected (or “stopping times”) are the same for both confidence sequences and the associated p -values. Thus, nothing is lost by basing the RLA on confidence sequences rather than anytime p -values. Confidence sequences benefit from being visually intuitive and are arguably easier to interpret than anytime p -values.

For example, consider conducting an RLA for a simple two-candidate election between Alice and Bob with no invalid votes. Suppose that it is reported that Alice won, i.e., $\mu^* := \frac{1}{N} \sum_{i=1}^N x_i > 1/2$ where $x_i = 1$ if the i th ballot is for Alice, 0 if for Bob, and $1/2$ if the ballot does not contain a valid vote for either candidate. A sequential RLA in the SHANGRLA framework would posit a null hypothesis $H_0 : \mu^* \leq 1/2$ (the complement of the announced result: Bob actually won or the outcome is a tie), sample random ballots sequentially, and stop the audit (confirming the announced result) if and when H_0 is rejected at significance level α . If H_0 is not rejected before all ballots have been inspected, the true outcome is known.⁴

On the other hand, a ballot-polling RLA [6] based on confidence sequences proceeds by computing a lower $1 - \alpha$ confidence bound for the fraction μ^* of votes for Alice. The audit stops, confirming the outcome, if and when this lower bound is larger than $1/2$. If that does not occur before the last ballot has been examined, the true outcome is known. In this formulation, there is no need to

⁴ At any point during the sampling, an election official can choose to abort the sampling and perform a full hand count for any reason. This cannot increase the risk limit: the chance of failing to correct an incorrect reported outcome does not increase.

Risk Limiting Audits via Confidence Sequences

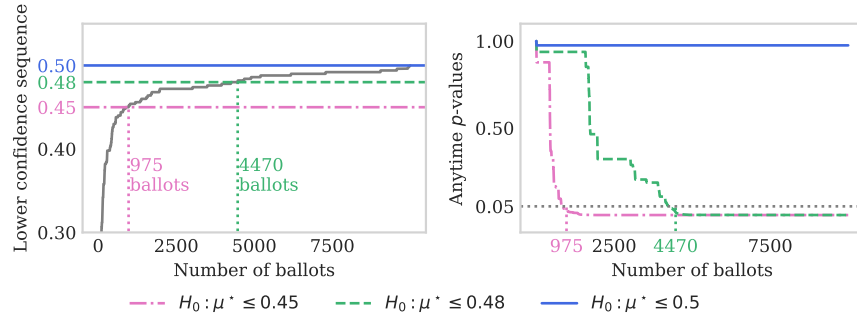


Fig. 2. The duality between anytime p -values and confidence sequences for three nulls: $H_0: \mu^* \leq \mu_0$ for $\mu_0 \in \{0.45, 0.48, 0.5\}$. The p -value for $H_0: \mu^* \leq 0.45$ (pink dash-dotted line) drops below 5% after 975 samples, exactly when the 95% lower confidence sequence exceeds 0.45. However, the p -value for $H_0: \mu^* \leq 0.5$ never reaches 0.05 and the 95% confidence sequence never excludes 0.5, the true value of μ^* .

define a null hypothesis as the complement of the announced result and interpret the resulting p -value, and so on. The approach also works for comparison audits using the “overstatement assorter” approach developed in [13], which transforms the problem into the same canonical form: testing whether the mean of any list in a collection of nonnegative, bounded lists is less than $1/2$.

2.2 Auditing Multiple Contests

It is known that RLAs of multi-candidate, multi-winner elections can be reduced to several pairwise contests without adjusting for multiplicity [6]. This is accomplished by testing whether every single reported winner beat every single reported loser, and stopping once each of these tests rejects their respective nulls at level $\alpha \in (0, 1)$. For example, suppose it is reported that a set of candidates \mathcal{W} beat a set of candidates \mathcal{L} in a k -winner plurality contest with K candidates in all (that is, $|\mathcal{W}| = k$ and $|\mathcal{L}| = K - k$). For each reported winner $w \in \mathcal{W}$ and each reported loser $\ell \in \mathcal{L}$, encode votes for candidate w as “1”, votes for ℓ as “0” and ballots with no valid vote in the contest or with a vote for any other candidate as “1/2” to obtain the population $\{x_1^{w,\ell}, \dots, x_N^{w,\ell}\}$. Then as before, candidate w beat candidate ℓ if and only if $\mu_{w,\ell}^* := \frac{1}{N} \sum_{i=1}^N x_i^{w,\ell} > 1/2$. In a two-candidate plurality election we would have proceeded by testing the null $H_0^{w,\ell}: \mu_{w,\ell}^* \leq 1/2$ against the alternative $H_1^{w,\ell}: \mu_{w,\ell}^* > 1/2$. To use the decomposition of a single winner or multi-winner plurality contest into a set of pairwise contests, we test each null $H_0^{w,\ell}: \mu_{w,\ell}^* \leq 1/2$ for $w \in \mathcal{W}$ and $\ell \in \mathcal{L}$. The audit stops if and when all $k(K - k)$ null hypotheses are rejected. Crucially, if candidate $w \in \mathcal{W}$ did not win (i.e. $\mu_{w,\ell}^* \leq 1/2$ for some $\ell \in \mathcal{L}$), then

$$\mathbb{P}(\text{reject all } H_{0,w,\ell}: w \in \mathcal{W}, \ell \in \mathcal{L}) \leq \min_{w \in \mathcal{W}, \ell \in \mathcal{L}} \mathbb{P}(\text{reject } H_{0,w,\ell}) \leq \alpha.$$

Waudby-Smith et al.

The same technique applies when auditing with confidence sequences. Let $\{(C_t^{w,\ell})_{t=1}^N\}$ be $(1 - \alpha)$ confidence sequences for $\{\mu_{w,\ell}^*\}$, $w \in \mathcal{W}$, $\ell \in \mathcal{L}$. We verify the electoral outcome of every contest once $C_t^{w,\ell} \subseteq (1/2, u]$ for all $w \in \mathcal{W}$, $\ell \in \mathcal{L}$. Again, if $\mu_{w,\ell}^* \leq 1/2$ for some $w \in \mathcal{W}$, and $\ell \in \mathcal{L}$, then

$$\mathbb{P}(\forall w \in \mathcal{W}, \forall \ell \in \mathcal{L}, C_t^{w,\ell} \subseteq (1/2, u]) \leq \min_{w \in \mathcal{W}, \ell \in \mathcal{L}} \mathbb{P}(C_t^{w,\ell} \subseteq (1/2, u]) \leq \alpha.$$

This technique can be generalized to handle audits of any number of contests from the same audit sample, as explained in [13]. For the sake of brevity, we omit the derivation, but it is a straightforward extension of the above.

3 Designing Powerful Confidence Sequences for RLAs

So far we have discussed how to conduct RLAs from confidence sequences for the parameter μ^* . In this section, we will discuss how to derive powerful confidence sequences for the purposes of conducting RLAs as efficiently as possible. For mathematical and notational convenience in the following derivations, we consider the case where $u = 1$. Note that nothing is lost in this setup since any population of $[0, u]$ -bounded numbers can be scaled to the unit interval $[0, 1]$ by dividing each element by u (thereby scaling the population’s mean as well).

As discussed in Section 2.1, we can construct confidence sequences by “inverting” sequential hypothesis tests. In particular, given a sequential hypothesis test $(\phi_t^\mu)_{t=1}^N$, the sequence of sets,

$$C_t := \{\mu \in [0, 1] : \phi_t^\mu = 0\}$$

forms a $(1 - \alpha)$ confidence sequence for μ^* . Consequently, in order to develop powerful RLAs via confidence sequences, we can simply focus on carefully designing sequential tests $(\phi_t^\mu)_{t=1}^N$.⁵

To design sequential hypothesis tests, we start by finding *martingales* that translate to powerful tests. To this end, define $M_0(\mu) := 1$ and consider the following process for $t \in [N]$:

$$M_t(\mu) := \prod_{i=1}^t (1 + \lambda_i(X_i - \mathcal{C}_i(\mu))), \quad (2)$$

where $\lambda_i \in \left[0, \frac{1}{\mathcal{C}_i(\mu)}\right]$ is a tuning parameter depending only on X_1, \dots, X_{i-1} , and

$$\mathcal{C}_i(\mu) := \frac{N\mu - \sum_{j=1}^{i-1} X_j}{N - i + 1}$$

⁵ Notice that it is not always feasible to compute the set of all $\mu \in [0, 1]$ such that $\phi_t^\mu = 0$ since $[0, 1]$ is uncountably infinite. However, all confidence sequences we will derive in this section are intervals (i.e. convex), and thus we can find the endpoints using a simple grid search or standard root-finding algorithms.

Risk Limiting Audits via Confidence Sequences

is the conditional mean of $X_i \mid X_1, \dots, X_{i-1}$ if the mean of $\{x_1, \dots, x_N\}$ were μ .

Following [15, Section 6], the process $(M_t(\mu^*))_{t=0}^N$ is a nonnegative martingale starting at one. Formally, this means that $M_0(\mu^*) = 1$, $M_t(\mu^*) \geq 0$, and

$$\mathbb{E}(M_t(\mu^*) \mid X_1, \dots, X_{t-1}) = M_{t-1}(\mu^*)$$

for each $t \in [N]$. Importantly for our purposes, nonnegative martingales are unlikely to ever become very large. This fact is known as *Ville's inequality* [14,2], which serves as a generalization of Markov's inequality to nonnegative (super)martingales, and can be stated formally as

$$\mathbb{P}(\exists t \in [N] : M_t(\mu^*) \geq 1/\alpha) \leq \alpha M_0(\mu^*) = \alpha, \quad (3)$$

where $\alpha \in (0, 1)$, and the equality follows from the fact that $M_0(\mu^*) = 1$. As alluded to in Section 2, $(M_t(\mu^*))_{t=0}^N$ can be interpreted as the reciprocal of an anytime p -value:

$$\mathbb{P}\left(\exists t \in [N] : \frac{1}{M_t(\mu^*)} \leq \alpha\right) \leq \alpha,$$

which matches the probabilistic guarantee in (1). As a direct consequence of Ville's inequality, if we define the test $\phi_t^\mu := \mathbb{1}(M_t(\mu) \geq 1/\alpha)$, then

$$\mathbb{P}(\exists t \in [N] : \phi_t^{\mu^*} = 1) \leq \alpha,$$

and thus $(\phi_t^\mu)_{t=1}^N$ is a level- α sequential hypothesis test. We can then invert $(\phi_t^\mu)_{t=1}^N$ and apply Theorem 1 to obtain confidence sequence-based RLAs with risk limit α .

3.1 Designing Martingales and Tests from Reported Vote Totals

So far, we have found a process $(M_t(\mu))_{t=0}^N$ that is a nonnegative martingale when $\mu = \mu^*$, but what happens when $\mu \neq \mu^*$? This is where the tuning parameters $(\lambda_t)_{t=1}^N$ come into the picture. Recall that an electoral assertion \mathcal{A} is certified once $C_t \subseteq \mathcal{A}$. Therefore, to audit assertions quickly, we want C_t to be as tight as possible. Since C_t is defined as the set of $\mu \in [0, 1]$ such that $M_t(\mu) < 1/\alpha$, we can make C_t tight by making $M_t(\mu)$ as *large* as possible. To do so, we must carefully choose $(\lambda_t)_{t=1}^N$. This choice will depend on the type of election as well as the amount of information provided prior to the audit. First consider the case where reported vote totals are given (in addition to the announced winner).

For example, recall the election between Alice and Bob of Section 2, and suppose that $\{x_1, \dots, x_N\}$ is the list of numbers encoding votes for Alice as 1, votes for Bob as 0, and ballots with no valid vote for either candidate as $1/2$. Recall that Alice beat Bob if and only if $\mu^* := \frac{1}{N} \sum_{i=1}^N x_i > 1/2$, so we are interested in testing the null hypothesis $H_0 : \mu^* \leq 1/2$ against the alternative $H_1 : \mu^* > 1/2$. Suppose it is reported that Alice beat Bob with N'_A votes for Alice, N'_B for Bob,

Waudby-Smith et al.

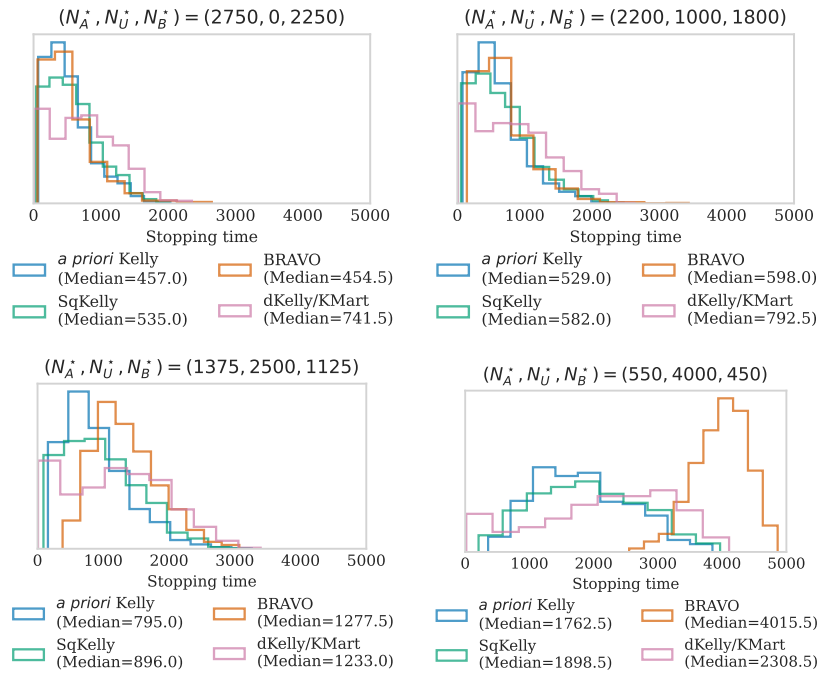


Fig. 3. Ballot-polling audit workload distributions under four possible outcomes of a two-candidate plurality election. Workload is defined as the number of distinct ballots examined before completing the audit. The first example considers an outcome where Alice and Bob received 2750 and 2250 votes respectively, and no ballots were invalid, for a margin of 0.1. The second, third, and fourth examples have the same margin, but with increasing numbers of invalid or “nuisance” ballots represented by N_U^* . Notice that in the case with no nuisance ballots, *a priori* Kelly and BRAVO have an edge, while in the setting with many nuisance ballots, *a priori* Kelly vastly outperforms BRAVO. On the other hand, neither SqKelly nor dKelly require tuning based on the reported outcomes, but SqKelly outperforms dKelly in all four scenarios.

and N_U' nuisance votes (i.e. either invalid or for another party). If the reported outcome is *correct*, then for any fixed λ , we know the exact value of

$$\prod_{i=1}^N (1 + \lambda(x_i - 1/2)), \quad (4)$$

which is an inexact but reasonable proxy for $M_N(1/2)$, the final value of the process $(M_t(1/2))_{t=0}^N$. We can then choose the value of λ' that maximizes (4). Some algebra reveals that the maximizer of (4) is given by

$$\lambda' := 2 \frac{N_A' - N_B'}{N_A' + N_B'}. \quad (5)$$

We then truncate λ' to obtain

$$\lambda_t^{\text{apK}} := \min \left\{ \lambda', \frac{1}{\mathcal{C}_t(\mu^*)} \right\}, \quad (6)$$

ensuring that it lies in the allowable range $[0, 1/\mathcal{C}_t(\mu)]$. We call this choice of λ_t^{apK} ***a priori Kelly*** due to its connections to Kelly's criterion [5,15] for maximizing products of the form (4). This choice of λ_t^{apK} also has the desirable property of yielding convex confidence sequences, which we summarize below.

Proposition 1. *Let X_1, \dots, X_N be a sequential random sample from $\{x_1, \dots, x_N\}$ with $\mu^* := \frac{1}{N} \sum_{i=1}^N x_i$. Consider $(\lambda_t^{\text{apK}})_{t=1}^N$ from (6) and define the process $M_t(\mu) := \prod_{i=1}^t (1 + \lambda_i^{\text{apK}}(X_i - \mathcal{C}_i(\mu)))$ for any $\mu \in [0, 1]$. Then the confidence set*

$$\mathcal{C}_t^{\text{apK}} := \{\mu \in [0, 1] : M_t(\mu) < 1/\alpha\}$$

is an interval with probability one.

Proof. Notice that since $\lambda' \geq 0$, $\mathcal{C}_t(\mu) \geq 0$, and $X_i \geq 0$, we have that

$$\lambda_t^{\text{apK}}(X_i - \mathcal{C}_i(\mu)) = \min\{\lambda' X_i, X_i/\mathcal{C}_i(\mu)\} - \min\{\lambda' \mathcal{C}_i(\mu), 1\}$$

is a nonincreasing function of μ for each $t \in [N]$. Consequently, $M_t(\mu)$ is a nonincreasing and quasiconvex function of μ , so its sublevel sets are convex. \square

Note that *any* sequence $(\lambda_t)_{t=1}^N$ such that $\lambda_t \in [0, 1/\mathcal{C}_t(\mu)]$ would have yielded a valid nonnegative martingale, but we chose that which maximizes (4) so that the resulting hypothesis test $\phi_t := \mathbf{1}(M_t(1/2) > 1/\alpha)$ is powerful. In situations more complex than two-candidate plurality contests, the maximizer of (4) can still be found efficiently via standard root-finding algorithms. All of these methods are implemented in our Python package.⁶

While audits based on *a priori Kelly* display excellent empirical performance (see Figure 3), their efficiency may be hurt when vote totals are erroneously reported. Small errors in reported vote totals seem to have minor adverse effects on stopping times (and in some cases can be slightly beneficial), but larger errors can significantly affect stopping time distributions (see Figure 4). If we wish to audit the reported winner of an election but prefer not to rely on (or do not have access to) exact reported vote totals, we need an alternative to *a priori Kelly*. In the following section, we describe a family of such alternatives.

3.2 Designing Martingales and Tests without Vote Totals

If the exact vote totals are not known, but we still wish to audit an assertion (e.g. that Alice beat Bob), we need to design a slightly different martingale that does not depend on maximizing (4) directly. Instead of finding an optimal

⁶ github.com/wannabesmith/RiLACS

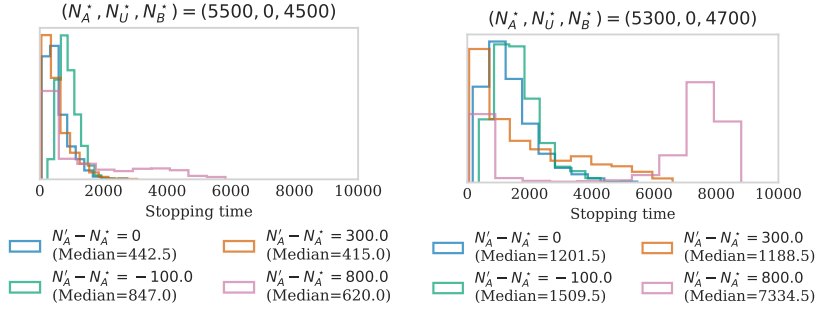


Fig. 4. Stopping times for *a priori* Kelly under various degrees of error in reported outcomes. In the above legends, N_A^* refers to the *true* number of votes for Alice, while N_A' refers to the incorrectly reported number of votes. Notice that empirical performance is relatively strong for $N_A' - N_A^* \in \{0, 300\}$ but is adversely affected when $N_A' - N_A^* \in \{-100, 800\}$, especially in the right-hand side plot with a narrower margin.

λ' , we will take $D \geq 2$ points evenly-spaced on the allowable range $[0, 1/\mathcal{C}_t(\mu)]$ and “hedge our bets” among all of these. Making this more precise, note that a convex combination of martingales (with respect to the same filtration) is itself a martingale [15], and thus for any $(\theta_1, \dots, \theta_D)$ such that $\theta_d \geq 0$ and $\sum_{d=1}^D \theta_d = 1$, we have that

$$M_t^D(\mu^*) := \sum_{d=1}^D \theta_d \prod_{i=1}^t \left(1 + \frac{d}{(D+1)\mathcal{C}_i(\mu^*)} (X_i - \mathcal{C}_i(\mu^*)) \right) \quad (7)$$

forms a nonnegative martingale starting at one. Notice that we no longer have to depend on the reported vote totals to begin an audit. Furthermore, confidence sequences generated using sublevel sets of $M_t^D(\mu)$ are intervals with probability one [15, Proposition 4]. Nevertheless, choosing $(\theta_1, \dots, \theta_D)$ is a nontrivial task. A natural — but as we will see, suboptimal — choice is to set $\theta_d = 1/D$ for each $d \in [D]$. Previous works [15] call this **dKelly** (for “diversified Kelly”), a name we adopt here. In fact, this choice of $(\theta_1, \dots, \theta_D)$ gives an arbitrarily close and computationally efficient approximation to the *Kaplan martingale* (**KMart**) [13] which can otherwise be prohibitively expensive to compute for large N .

Better choices of $(\theta_d)_{d=1}^D$ exist for the types of elections one might encounter in practice. Recall that near-optimal values of λ are given by (5). However, setting $\theta_d = 1/D$ for each $d \in [D]$ implicitly treats all $d/((D+1)\mathcal{C}_i(\mu^*))$ as equally reasonable values of λ . Elections with large values of μ^* (e.g. closer to 1) are “easier” to audit, and the interesting or “difficult” regime is when μ^* is close to (but strictly larger than) $1/2$. Therefore, we recommend designing $(\theta_1, \dots, \theta_D)$ so that $(M_t^D(1/2))_{t=0}^N$ upweights optimal values of λ for margins close to 0, and downweights those for margins close to 1. Consider the following concrete

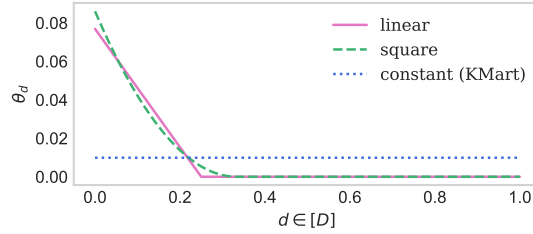


Fig. 5. Various values of the convex weights $(\theta_1, \dots, \theta_D)$, which can be used in the construction of the diversified martingale (7). Notice that the linear and square weights are largest for d near 0, and decrease as d approaches $1/4$, finally remaining at 0 for all large d . Smaller values of d are upweighted since they correspond to those values of λ in $M_t^D(\mu^*)$ that are optimal for smaller (i.e. interesting) electoral margins. This is in contrast to the constant weight function, which sets $\theta_d = 1/D$ for each $d \in [D]$. We find that square weights perform well in practice (see Figure 3) but these can be tuned and tailored based on prior knowledge and the particular problem at hand.

examples. First, we have the truncated-square weights,

$$\theta_d^{\text{square}} := \frac{\gamma_d^{\text{square}}}{\sum_{d=1}^D \gamma_d^{\text{square}}}, \quad \text{where } \gamma_d^{\text{square}} := (1/3 - x)^2 \mathbf{1}_{d \leq 1/3}.$$

and we normalize by $\sum_d \gamma_d^{\text{linear}}$ to ensure that $\sum_d \theta_d = 1$. Another sensible choice is given by the truncated-linear weights, where we simply replace γ_d^{square} by $\gamma_d^{\text{linear}} := \max\{0, 1 - 2d\}$. These values of θ_d^{linear} and θ_d^{square} are large for $d \approx 0$ and small for $d \gg 0$, and hence the summands in the martingale given by (7) are upweighted for implicit values of λ which are optimal for “interesting” margins close to 0, and downweighted for simple margins much larger than 0 (see Figure 5).

When M_t^D is combined with θ_d^{square} , we refer to the resulting martingales and confidence sequences as **SqKelly**. We compare their empirical workload against that of *a priori* Kelly, dKelly, and BRAVO in Figure 3. A hybrid approach is also possible: suppose we want to use reported outcomes or prior knowledge alongside these convex-weighted martingales. We can simply choose $(\theta_1, \dots, \theta_D)$ so that M_t^D upweights values in a neighborhood of λ' (or some other value chosen based on prior knowledge⁷).

4 Risk-Limiting Tallies via Confidence Sequences

Rather than audit an already-announced electoral outcome, it may be of interest to determine (for the purposes of making a first announcement) the election

⁷ The use of the word “prior” here should not be interpreted in a Bayesian sense. No matter what values of $(\theta_1, \dots, \theta_D)$ are chosen, the resulting tests and confidence sequences have *frequentist* risk-limiting guarantees.

winner with high probability, without counting all N ballots. Such procedures are known as risk-limiting tallies (RLTs), which were developed for coercion-resistant, end-to-end verifiable voting schemes [4]. For example, suppose a voter is being coerced to vote for Bob. If the final vote tally reveals that Bob received few or no votes, then the coercer will suspect that the voter did not comply with instructions. RLTs provide a way to mitigate this issue by providing high-probability guarantees that the reported winner truly won, leaving a large proportion of votes shrouded. In such cases, the voter is guaranteed plausible deniability, as they can claim to the coercer that their ballot is simply among the unrevealed ones.

While the motivations for RLTs are quite different from those for RLAs, the underlying techniques are similar. The same is true for confidence sequence-based RLTs. All methods introduced in this paper can be applied to RLTs (with the exception of “*a priori* Kelly” since it depends on the reported outcome) but with two-sided power. Consider the martingales we discussed in Section 3.2,

$$M_t^D(\mu^*) := \sum_{d=1}^D \theta_d \prod_{i=1}^t \left(1 + \frac{d}{(D+1)\mathcal{C}_i(\mu^*)} (X_i - \mathcal{C}_i(\mu^*)) \right), \quad (8)$$

where $(\theta_1, \dots, \theta_D)$ are convex weights. Recall that our confidence sequences at a given time t were defined as those $\mu \in [0, 1]$ for which $M_t^D(\mu) < 1/\alpha$. In other words, a given value μ is only excluded from the confidence set if $M_t^D(\mu)$ is large. However, notice that $M_t^D(\mu)$ will become large if the conditional mean $\mathcal{C}_t(\mu^*) \equiv \mathbb{E}(X_t | X_1, \dots, X_{t-1})$ is larger than the null conditional mean $\mathcal{C}_t(\mu)$, but the same cannot be said if $\mathcal{C}_t(\mu^*) < \mathcal{C}_t(\mu)$. As a consequence, the resulting confidence sequences are all one-sided *lower* confidence sequences. To ensure that our bounds have non-trivial two-sided power, we can simply combine (8) with a martingale that also grows when $\mathcal{C}_t(\mu^*) < \mathcal{C}_t(\mu)$.

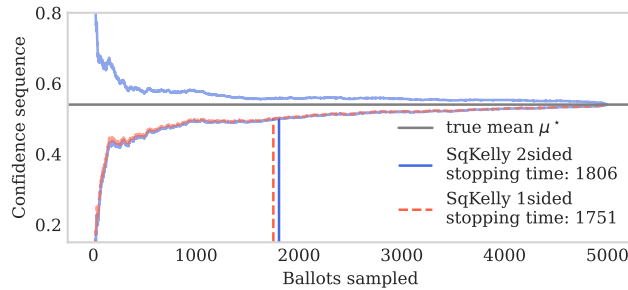


Fig. 6. Confidence sequence-based risk-limiting tally for a two-candidate election. Unlike RLAs, RLTs require two-sided confidence sequences so that the true winner can be determined (with high probability) without access to an announced result. Notice that testing the same null $H_0 : \mu^* \leq 0.5$ is less efficient in an RLT than in an RLA. This is a necessary sacrifice for having nontrivial power against other alternatives.

Proposition 2. For nonnegative vectors $(\theta_1^+, \dots, \theta_D^+)$ and $(\theta_1^-, \dots, \theta_D^-)$ that each sum to one, define the processes

$$M_t^{D^+}(\mu) := \sum_{d=1}^D \theta_d^+ \prod_{i=1}^t \left(1 + \frac{d}{(D+1)\mathcal{C}_i(\mu^*)} (X_i - \mathcal{C}_i(\mu^*)) \right),$$

$$M_t^{D^-}(\mu) := \sum_{d=1}^D \theta_d^- \prod_{i=1}^t \left(1 - \frac{d}{(D+1)(1 - \mathcal{C}_i(\mu^*))} (X_i - \mathcal{C}_i(\mu^*)) \right).$$

Next, for $\beta \in [0, 1]$, define their mixture

$$M_t^{D^\pm}(\mu) := \beta M_t^{D^+}(\mu) + (1 - \beta) M_t^{D^-}(\mu).$$

Then, $M_t^{D^\pm}(\mu^*)$ is a nonnegative martingale starting at one. Consequently,

$$C_t^\pm := \{\mu \in [0, 1] : M_t^{D^\pm}(\mu) < 1/\alpha\}$$

forms a $(1 - \alpha)$ confidence sequence for μ^* .

Proof. This follows immediately from the fact that both $M_t^{D^+}(\mu^*)$ and $M_t^{D^-}(\mu^*)$ are martingales with respect to the same filtration, and that convex combinations of such martingales are also martingales. \square

With this setup and notation in mind, M_t^D as defined in Section 3.2 is a special case of $M_t^{D^\pm}$ with $\beta = 1$. As noted by [4], RLTs involving multiple assertions *do* require correction for multiple testing, unlike RLAs. The same is true for confidence sequence-based RLTs (and hence the tricks of Section 2.2 do not apply). It suffices to perform a simple Bonferroni correction by constructing $(1 - \alpha/K)$ confidence sequences to establish K simultaneous assertions.

5 Summary

This paper presented a general framework for conducting risk-limiting audits based on confidence sequences, and derived computationally and statistically efficient martingales for computing them. We showed how *a priori* Kelly takes advantage of the reported vote totals (if available) to stop ballot-polling audits significantly earlier than extant ballot-polling methods, and how alternative martingales such as SqKelly also provide strong empirical performance in the absence of reported outcomes. Finally, we demonstrated how a simple tweak to the aforementioned algorithms provides two-sided confidence sequences, which can be used to perform risk-limiting tallies. Confidence sequences and these martingales can be applied to ballot-level comparison audits and batch-level comparison audits as well, using “overstatement assorters” [13], which reduce comparison audits to the same canonical statistical problem: testing whether the mean of any list in a collection of non-negative bounded lists is at most $1/2$. We hope that this new perspective on RLAs and its associated software will aid in making election audits simpler, faster, and more transparent.

References

1. Blom, M., Budurushi, J., Rivest, R.L., Stark, P.B., Stuckey, P.J., Teague, V., Vukcevic, D.: Assertion-based approaches to auditing complex elections, with application to party-list proportional elections. In: International Joint Conference on Electronic Voting. Springer (2021)
2. Howard, S.R., Ramdas, A., McAuliffe, J., Sekhon, J.: Time-uniform Chernoff bounds via nonnegative supermartingales. *Probability Surveys* **17**, 257–317 (2020)
3. Huang, Z., Rivest, R.L., Stark, P.B., Teague, V.J., Vukcevic, D.: A unified evaluation of two-candidate ballot-polling election auditing methods. In: International Joint Conference on Electronic Voting. pp. 112–128. Springer (2020)
4. Jamroga, W., Roenne, P.B., Ryan, P.Y., Stark, P.B.: Risk-limiting tallies. In: International Joint Conference on Electronic Voting. pp. 183–199. Springer (2019)
5. Kelly Jr, J.: A new interpretation of information rate. *Bell System Technical Journal* **35**(4), 917–926 (1956)
6. Lindeman, M., Stark, P.B., Yates, V.S.: BRAVO: Ballot-polling risk-limiting audits to verify outcomes. In: 2012 Electronic Voting Technology Workshop/-Workshop on Trustworthy Elections (EVT/WOTE 12). USENIX Association, Bellevue, WA (Aug 2012), <https://www.usenix.org/conference/evtwote12/workshop-program/presentation/lindeman>
7. Ottoboni, K., Bernhard, M., Halderman, A., Rivest, R., Stark, P.: Bernoulli ballot polling: a manifest improvement for risk-limiting audits. In: International Conference on Financial Cryptography and Data Security. pp. 226–241. Springer (2019)
8. Ottoboni, K., Stark, P., Lindeman, M., McBurnett, N.: Risk-limiting audits by stratified union-intersection tests of elections (SUITE). In: International Joint Conference on Electronic Voting. pp. 174–188. Springer (2018)
9. Rivest, R.L.: ClipAudit: A simple risk-limiting post-election audit. arXiv preprint arXiv:1701.08312 (2017)
10. Stark, P.B.: Conservative statistical post-election audits. *The Annals of Applied Statistics* **2**(2), 550–581 (2008)
11. Stark, P.B.: CAST: Canvass audits by sampling and testing. *IEEE Transactions on Information Forensics and Security* **4**(4), 708–717 (2009)
12. Stark, P.B.: Risk-limiting postelection audits: Conservative p -values from common probability inequalities. *IEEE Transactions on Information Forensics and Security* **4**(4), 1005–1014 (2009)
13. Stark, P.B.: Sets of half-average nulls generate risk-limiting audits: SHANGRLA. In: International Conference on Financial Cryptography and Data Security. pp. 319–336. Springer (2020)
14. Ville, J.: Etude critique de la notion de collectif. *Bull. Amer. Math. Soc* **45**(11), 824 (1939)
15. Waudby-Smith, I., Ramdas, A.: Estimating means of bounded random variables by betting. arXiv preprint arXiv:2010.09686 (2021)

Assertion-based Approaches to Auditing Complex Elections, with application to party-list proportional elections

Michelle Blom¹[0000–0002–0459–9917], Jurlind Budurushi²[0000–0002–6732–4400],
Ronald L. Rivest³[0000–0002–7105–3690], Philip B. Stark⁴[0000–0002–3771–9604],
Peter J. Stuckey⁵[0000–0003–2186–0459], Vanessa Teague⁶[0000–0003–2648–2565],
and Damjan Vukcevic^{7,8}[0000–0001–7780–9586]

¹ School of Computing and Information Systems, University of Melbourne, Parkville,
Australia

`michelle.blom@unimelb.edu.au`

² Cloudical Deutschland GmbH

`jurlind.budurushi@cloudical.io`

³ Computer Science & Artificial Intelligence Laboratory, Massachusetts Institute of
Technology, USA

⁴ Department of Statistics, University of California, Berkeley, CA, USA

⁵ Department of Data Science and AI, Monash University, Clayton, Australia

⁶ Thinking Cybersecurity Pty. Ltd.

⁷ School of Mathematics and Statistics, University of Melbourne, Parkville, Australia

⁸ Melbourne Integrative Genomics, University of Melbourne, Parkville, Australia
`damjan.vukcevic@unimelb.edu.au`

Abstract. Risk-limiting audits (RLAs), an ingredient in evidence-based elections, are increasingly common. They are a rigorous statistical means of ensuring that electoral results are correct, usually without having to perform an expensive full recount—at the cost of some controlled probability of error. A recently developed approach for conducting RLAs, SHANGRLA, provides a flexible framework that can encompass a wide variety of social choice functions and audit strategies. Its flexibility comes from reducing sufficient conditions for outcomes to be correct to canonical ‘assertions’ that have a simple mathematical form.

Assertions have been developed for auditing various social choice functions including plurality, multi-winner plurality, super-majority, Hamiltonian methods, and instant runoff voting. However, there is no systematic approach to building assertions. Here, we show that assertions with *linear* dependence on transformations of the votes can easily be transformed to canonical form for SHANGRLA. We illustrate the approach by constructing assertions for party-list elections such as Hamiltonian free list elections and elections using the D’Hondt method, expanding the set of social choice functions to which SHANGRLA applies directly.

Keywords: Risk-limiting audits · Party-list proportional elections · Hamiltonian methods · D’Hondt method

1 Introduction

Risk-limiting audits (RLAs) test reported election outcomes statistically by manually inspecting random samples of paper ballots. An RLA terminates either by endorsing the reported outcome or by proceeding to a full manual count if the evidence is inconclusive. The outcome according to the full count corrects the reported outcome if they differ. The *risk limit* is an upper bound on the probability that a wrong election outcome will not be corrected—this is set in advance, typically between 1% and 10%.

SHANGRLA [4] is a general framework for conducting RLAs of a wide variety of social choice functions.⁹ SHANGRLA involves reducing the correctness of a reported outcome to the truth of a set \mathcal{A} of quantitative *assertions* about the set of validly cast ballots, which can then be tested using statistical methods. The assertions are either true or false depending on the votes on the ballots. If all the assertions are true, the reported outcome is correct.

This paper shows how to use the SHANGRLA RLA method to audit some complex social choice functions not addressed in the SHANGRLA paper. We give a recipe for translating sufficient conditions for a reported outcome to be correct into canonical form for SHANGRLA, when those conditions are the intersection of a set of linear inequalities involving transformations of the votes on each ballot. We focus on European-style party-list proportional representation elections, with the German state of Hesse as a case study.

1.1 Assertion-based auditing: Properties and challenges

For some social choice functions, the reduction to assertions is obvious. For instance, in plurality (first-past-the-post) elections, common in the United States, Alice won the election if and only if Alice’s tally was higher than that of each of the other $n - 1$ candidates (where n is the total number of candidates). That set of $n - 1$ assertions is clearly a set of linear inequalities among the vote totals for the n candidates.

In general, assertions involve not only the votes but also the reported results—the reported outcome and possibly the voting system’s interpretation of individual ballots (CVRs) or tallies of groups of ballots.

SHANGRLA [4, Sec 2.5] shows how to make asserters for any ‘scoring rule’ (e.g. Borda, STAR-voting, and any weighted scheme). For more complex social choice functions, constructing sufficient sets of assertions may be much less obvious. Blom *et al.* [2] use a heuristic method, RAIRE, to derive assertions for Instant Runoff Voting (IRV) from the CVRs. RAIRE allows the RLA to test an IRV outcome—the claim that Alice won—without checking the entire IRV elimination. RAIRE’s assertions are *sufficient*: if all of the assertions in \mathcal{A} are true,

⁹ Any social choice function that is a *scoring rule*—that assigns ‘points’ to candidates on each ballot, sums the points across ballots, and declares the winner(s) to be the candidate(s) with the most ‘points’—can be audited using SHANGRLA, as can some social choice functions that are not scoring rules, such as super-majority and IRV.

then the announced election outcome is correct. However, the set of assertions might not be necessary—even if one of the assertions in \mathcal{A} is false, Alice may still have won, but for reasons not checked by the audit.

A social choice function might be expensive to audit for two different reasons: it might require a very large sample for reasonable confidence, even when there are no errors (for instance, if it tends to produce small margins in practice); alternatively, it might be so complex that it is difficult to generate assertions that are sufficient to prove the reported election outcome is correct. Pilots and simulations suggest that IRV elections do not have small margins any more often than first-past-the-post elections. Hence IRV is feasible to audit in both senses.

Below, the sets of assertions we consider are *conjunctive*: the election outcome is correct if all the assertions in \mathcal{A} are true. Although it is possible to imagine an audit method that tests more complex logical structures (for example, the announced outcome is correct if either all the assertions in \mathcal{A}_1 or all the assertions in \mathcal{A}_2 are true), this is not currently part of the SHANGRLA framework.

Summary: An audit designer must devise a set \mathcal{A} of assertions.

- \mathcal{A} generally depends on the social choice function and the reported electoral outcome, and may also depend on the CVRs, vote subtotals, or other data generated by the voting system.
- If every assertion in \mathcal{A} is true, then the announced electoral result is correct.
- The announced electoral result may be correct even if not every assertion in \mathcal{A} is true.

SHANGRLA relies on expressing assertions in terms of *assorters*.

1.2 Assorters

The statistical part of SHANGRLA is agnostic about the social choice function. It simply takes a collection of sets of numbers that are zero or greater (with a known upper bound), and decides whether to reject the hypothesis that the mean of each set is less than or equal to $1/2$ —this is the *assorter null hypothesis*.

An *assorter* for some assertion $A \in \mathcal{A}$ assigns a nonnegative value to each ballot, depending on the selections the voter made on the ballot and possibly other information (e.g. reported vote totals or CVRs). The assertion is true iff the mean of the assorter (over all ballots) is greater than $1/2$. Generally, ballots that support the assertion score higher than $1/2$, ballots that cast doubt on it score less than $1/2$, and neutral ballots score exactly $1/2$. For example, in a simple first-past-the-post contest, A might assert that Alice’s tally is higher than Bob’s. The corresponding assorter would assign 1 to a ballot if it has a vote for Alice, 0 if it has a vote for Bob, and $1/2$ if it has a no valid vote or a vote for some other candidate.

The audit designer’s first job is to generate a set \mathcal{A} of assertions which, if all true, imply that the announced electoral outcome (the winner or winners) is correct. Then they need to express each $A \in \mathcal{A}$ as an assorter. Finally, they need

to test the hypothesis that any assorter mean is less than or equal to $1/2$. If all those hypotheses are rejected, the audit concludes that the reported outcome is correct. The chance this conclusion is erroneous is at most the risk limit.

[Section 2](#) gives a more precise definition of an assorter and a general technique for transforming linear assertions into assorters.

1.3 Risk-limiting audits using SHANGRLA: Pulling it all together

An overview of the workflow for a sequential SHANGRLA RLA is:

1. Generate a set of assertions.
2. Express the assertions as assorters.
3. Test every assertion in \mathcal{A} , in parallel:
 - (a) Retrieve a ballot or set of ballots selected at random.
 - (b) Apply each assorter to every retrieved ballot.
 - (c) For each assertion in \mathcal{A} , test its corresponding assorter null hypothesis (i.e. that the assorter mean is $\leq 1/2$) using a sequentially valid test.¹⁰
 - (d) If the assorter null is rejected for $A \in \mathcal{A}$, remove A from \mathcal{A} .
 - (e) If \mathcal{A} is empty (i.e. all of the null hypotheses have been rejected), stop the audit and certify the electoral outcome.
 - (f) Otherwise, continue to sample more ballots.
 - (g) At any time, the auditor can decide to ‘cut to the chase’ and conduct a full hand count: anything that increases the chance of conducting a full hand count cannot increase the risk.

As with any RLA, the audit may not confirm the reported result (for example, that Alice’s tally is the highest) even if all assertions are true (Alice’s tally may actually be higher than Bob’s, but the audit may not gather enough evidence to conclude so). This may happen because there are many tabulation errors or because one or more margins are small. When the audit proceeds to a full hand count, its result replaces the reported outcome if the two differ.

Conversely, the audit may mistakenly confirm the result even if the announced result is wrong. The probability of this kind of failure is not more than the *risk limit*. This is a parameter to SHANGRLA; setting it to a smaller value generally entails examining more ballots.

1.4 Party-list proportional representation contests

Party-list proportional representation contests allocate seats in a parliament (or delegates to an assembly) in proportion to the entities’ popularity within the electorate. The first step is (usually) rounding the party’s fraction down to the nearest integer number of seats. Complexity arises from rounding, when the fractions determined by voters do not exactly match integer numbers of seats. *Largest Remainder Methods*, also called Hamiltonian methods, successively

¹⁰ It can be more efficient to sample ballots in ‘rounds’ rather than singly; SHANGRLA can accommodate any valid test of the assorter nulls.

allocate leftover seats to the entities with the largest fractional parts until all seats are allocated. *Highest Averages Methods*, such as the D’Hondt method (also called Jefferson’s method), weight this extra allocation by divisors involving a fraction of the seats already allocated to that party—they are hence more likely to allocate the leftover seats to small parties. The Sainte-Laguë method (also called Webster’s method) is mathematically similar but its divisors penalise large parties even more.¹¹

1.5 Related work and our contribution

Blom *et al.* [1] showed how to construct a SHANGRLA RLA for preferential Hamiltonian elections with a viability threshold, applicable to many US primaries. Stark and Teague [5] showed how to construct an RLA for highest averages party-list proportional representation elections. Their method was not directly based on assertions and assorters, but it reduces the correctness of the reported seat allocation to a collection of two-entity plurality contests, for which it is straightforward to construct assorters, as we show below.

This paper shows how to extend SHANGRLA to additional social choice functions. We use party-list proportional representation elections as an example, showing how the assorter from [1] can be derived as a special case of the solution for more general Hamiltonian elections. We have simulated the audit on election data from the German state of Hesse; results are shown in Section 4. Auditing the allocation of integer portions of seats involves inspecting a reasonable number of ballots, but the correctness of the allocations based on the fractional remainders and the correctness of the particular candidates who receive seats within each party generally involve very small margins, which in turn require large audit sample sizes. We also show how to apply the construction to highest averages methods such as D’Hondt and Sainte-Laguë. Our contributions are:

- A guide to developing assertions and their corresponding SHANGRLA assorters, so that audits for contest types that are not already supplied can be derived, when correctness can be expressed as the intersection of a set of linear inequalities (Section 3).
- New SHANGRLA-based methods for auditing largest remainder methods that allow individual candidate selection (no audit method was previously known for this variant of largest remainder method) (Section 3.1).
- Simulations to estimate the average sample sizes of these new methods in the German state of Hesse (Section 4).
- SHANGRLA assorters for highest averages methods (RLAs for these methods were already known, but had not been expressed as assorters). (Section 5).

¹¹ Another source of complexity is the opportunity for voters to select, exclude, or prioritise individual candidates within the party.

2 Preliminaries

2.1 Nomenclature and notation for assertion-based election audits

An election contest is decided by a set of ‘ground truth’ ballots \mathcal{L} (of cardinality $|\mathcal{L}|$). Many social choice functions are used in political elections. Some yield a single winner; others multiple winners. Some only allow voters to express a single preference; others allow voters to select or rank multiple candidates or parties.

Here, we focus on elections that allow voters to select (but not rank) one or more ‘entities,’ which could be candidates or parties.¹²

Let S be the number of ‘seats’ (positions) to be filled in the contest, of which a_e were awarded to entity e . Each ballot might represent a single vote for an entity, or multiple votes for multiple entities. Important quantities for individual ballots $b \in \mathcal{L}$ include:

- m , the maximum permitted number of votes for any entity.
- $m_{\mathcal{L}}$, the maximum permitted number of votes in total (across all entities).
- b_e , the total number of (valid) votes for entity e on the ballot.
- $b_T := \sum_{e \in E} b_e$, the total number of (valid) votes on the ballot.

Any of these may be greater than one, depending on the social choice function. Validity requires $b_e \leq m$ and $b_T \leq m_{\mathcal{L}}$. If ballot b does not contain the contest in question or is deemed invalid, $b_e := 0$ for all entities E , and $b_T := 0$.

Important quantities for the set \mathcal{L} of ballots include:

- $T_e = \sum_{b \in \mathcal{L}} b_e$, the *tally* of votes for entity e .
- $T_{\mathcal{L}} = \sum_{e \in E} T_e$, the total number of valid votes in the contest.
- $p_e = T_e/T_{\mathcal{L}}$, the *proportion* of votes for entity e .

2.2 Assertion-based auditing: Definitions

Here we formalize assertion-based auditing sketched in [Section 1](#) and introduce the relevant mathematical notation. An *assorter* h is a function that assigns a non-negative number to each ballot depending on the votes reflected on the ballot and other election data (e.g. the reported outcome, the set of CVRs, or the CVR for that ballot). Each assertion in the audit is equivalent to ‘the average value of the assorter for all the cast ballots is greater than $1/2$.’ In turn, each assertion is checked by testing the complementary null hypothesis that the average is less than or equal to $1/2$. If all the complementary null hypotheses are false, the reported outcome of every contest under audit is correct.

Definition 1. *An assertion is a statement A about the set of paper ballots \mathcal{L} of the contest. An assorter for assertion A is a function h_A that maps selections on a ballot b to $[0, M]$ for some known constant $M > 0$, such that assertion A holds for \mathcal{L} iff $\bar{h}_A > 1/2$ where \bar{h}_A is the average value of h_A over all $b \in \mathcal{L}$.*

A set \mathcal{A} of assertions is *sufficient* if their conjunction implies that the reported electoral outcome is correct.

¹² Below, in discussing assorters, we use the term ‘entity’ more abstractly. For instance, when voters may rank a subset of entities, the assorters may translate ranks into scoring functions in a nonlinear manner, as in [\[2\]](#)—we do not detail that case here.

2.3 Example assertions and assorters

Example 1. First-past-the-post voting. Consider a simple first-past-the-post contest, where the winner w is the candidate with the most votes and each valid ballot records a vote for a single candidate. The result is correct if the assertions $p_w > p_\ell$ for each losing candidate ℓ all hold.

We can build an assorter h for the assertion $p_w > p_\ell$ as follows:

$$h(b) = \begin{cases} 1 & b_w = 1 \text{ and } b_\ell = 0, \\ 0 & b_w = 0 \text{ and } b_\ell = 1, \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

Example 2. Majority contests. Consider a simple majority contest, where the winner is the candidate w achieving over 50% of the votes, assuming again each valid ballot holds a single vote (if there is no winner, a runoff election is held). The result can be verified by the assertion $p_w > 1/2$.

We can build an assorter h for the more general assertion $p_w > t$ as follows:

$$h(b) = \begin{cases} \frac{1}{2t} & b_w = 1 \text{ and } b_\ell = 0, \forall \ell \neq w, \\ 0 & b_w = 0 \text{ and } b_\ell = 1 \text{ for exactly one } \ell \neq w, \\ \frac{1}{2} & \text{invalid ballot.} \end{cases}$$

3 Creating assorters from assertions

In this section we show how to transform generic linear assertions, i.e. inequalities of the form $\sum_{b \in \mathcal{L}} \sum_{e \in E} a_e b_e > c$, into canonical assertions using assorters as required by SHANGRLA. There are three steps:

1. Construct a set of linear assertions that imply the correctness of the outcome.¹³
2. Determine a ‘proto-assorter’ based on this assertion.
3. Construct an assorter from the proto-assorter via an affine transformation.

We work with social choice functions where each valid ballot can contribute a non-negative (zero or more) number of ‘votes’ or ‘points’ to various tallies (we refer to these as *votes* henceforth). For example, in plurality voting we have a tally for each candidate and each ballot contributes a vote of 1 to the tally of a single candidate and a vote of 0 to all other candidates’ tallies. The tallies can represent candidates, groups of candidates, political parties, or possibly some more abstract groupings of candidates as might be necessary to describe an assertion (see below); we refer to them generically as *entities*.

Let the various tallies of interest be T_1, T_2, \dots, T_m for m different entities. These represent the total count of the votes across all valid ballots.

¹³ Constructing such a set is outside the scope of this paper; we suspect there is no general method. Moreover, there may be social choice functions for which there is no such set.

A *linear assertion* is a statement of the form

$$a_1T_1 + a_2T_2 + \cdots + a_mT_m > 0$$

for some constants a_1, \dots, a_m .

Each assertion makes a claim about the ballots, to be tested by the audit. For most social choice functions, the assertions are about proportions rather than tallies. Typically these proportions are of the total number of valid votes, $T_{\mathcal{L}}$, in which case we can restate the assertion in terms of tallies by multiplying through by $T_{\mathcal{L}}$.

For example, a pairwise majority assertion is usually written as $p_A > p_B$, stating that candidate A got a larger proportion of the valid votes than candidate B . We can write this in linear form as follows. Let T_A and T_B be the tallies of votes in favour of candidates A and B respectively. Then:

$$\begin{aligned} p_A &> p_B \\ \frac{T_A}{T_{\mathcal{L}}} &> \frac{T_B}{T_{\mathcal{L}}} \\ T_A &> T_B \\ T_A - T_B &> 0. \end{aligned}$$

Another example is a super/sub-majority assertion, $p_A > t$, for some threshold t . We can write this in linear form similar to above, as follows:

$$\begin{aligned} p_A &> t \\ \frac{T_A}{T_{\mathcal{L}}} &> t \\ T_A &> tT_{\mathcal{L}} \\ T_A - tT_{\mathcal{L}} &> 0. \end{aligned}$$

For a given linear assertion, we define the following function on ballots, which we call a *proto-assorter*:

$$g(b) = a_1b_1 + a_2b_2 + \cdots + a_mb_m,$$

where b is a given ballot, and b_1, b_2, \dots, b_m are the votes contributed by that ballot to the tallies T_1, T_2, \dots, T_m respectively.¹⁴

Summing this function across all ballots, $\sum_b g(b)$, gives the left-hand side of the linear assertion. Thus, the linear assertion is true iff $\sum_b g(b) > 0$. The same property holds for the average across ballots, $\bar{g} = 1/|\mathcal{L}| \sum_b g(b)$; the linear assertion is true iff $\bar{g} > 0$.

To obtain an assorter in canonical form, we apply an affine transformation to g such that it never takes negative values and also so that comparing its average value to $1/2$ determines the truth of the assertion. One such transformation is

$$h(b) = c \cdot g(b) + 1/2 \tag{1}$$

¹⁴ Note that $g(b) = 0$ for any invalid ballot b , based on previous definitions.

for some constant c .¹⁵ There are many ways to choose c . We present two here. First, we determine a lower bound for the proto-assorter, a value a such that $g(b) \geq a$ for all b .¹⁶ Note that $a < 0$ in all interesting cases: if not, the assertion would be trivially true ($\bar{g} > 0$) or trivially false ($\bar{g} \equiv 0$, with $a_j = 0$ for all j). If $a \geq -1/2$, simply setting $c = 1$ produces an assorter: we have $h \geq 0$, and $\bar{h} > 1/2$ iff $\bar{g} > 0$. Otherwise, we can choose $c = -1/(2a)$, giving

$$h(b) = \frac{g(b) - a}{-2a}. \quad (2)$$

(See [4, Sec. 2.5].) To see that $h(b)$ is an assorter, first note that $h(b) \geq 0$ since the numerator is always non-negative and the denominator is positive. Also, the sum and mean across all ballots are, respectively:

$$\begin{aligned} \sum_b h(b) &= -\frac{1}{2a} \sum_b g(b) + \frac{|\mathcal{L}|}{2} \\ \bar{h} &= -\frac{1}{2a} \bar{g} + \frac{1}{2}. \end{aligned}$$

Therefore, $\bar{h} > 1/2$ iff $\bar{g} > 0$.

3.1 Example: Pairwise difference assorter

To illustrate the approach, we will now create an assorter for a fairly complex assertion for quite complicated ballots. We consider a contest where each ballot can have multiple votes for multiple entities; the votes are simple—not ranks or scores. Let $m_{\mathcal{L}}$ be the maximum number of votes a single ballot can contain for that contest. We can use the above general technique to derive an assorter for the assertion $p_A > p_B + d$. In Section 4 we will use this for auditing Hamiltonian free list contests, where A and B will be parties. This assertion checks that the proportion of votes A has is greater than that of B plus a constant, d . This constant may be negative.

We start with the assertion $p_A > p_B + d$. We can rewrite this in terms of tallies as we did in the previous examples, giving the following linear form:

$$\begin{aligned} p_A &> p_B + d \\ \frac{T_A}{T_{\mathcal{L}}} &> \frac{T_B}{T_{\mathcal{L}}} + d \\ T_A &> T_B + dT_{\mathcal{L}} \\ T_A - T_B - dT_{\mathcal{L}} &> 0. \end{aligned}$$

¹⁵ Note that $h(b) = 1/2$ if ballot b has no valid vote in the contest.

¹⁶ If the votes b_j are bounded above by s and below by zero, then a bound (not necessarily the sharpest) on g is given by taking just the votes that contribute negative values to g , setting all of those votes to s , and setting the other votes to 0:

$$a = \sum_{j:a_j < 0} a_j s.$$

The corresponding proto-assorter is

$$g(b) = b_A - b_B - d \cdot b_T.$$

If the votes are bounded above by $m_{\mathcal{L}}$ then this has lower bound given by

$$g(b) \geq -m_{\mathcal{L}} - dm_{\mathcal{L}} = -m_{\mathcal{L}} \cdot (1 + d).$$

Therefore, an assorter is given by

$$h(b) = \frac{b_A - b_B - d \cdot b_T + m_{\mathcal{L}} \cdot (1 + d)}{2m_{\mathcal{L}} \cdot (1 + d)}.$$

When $m_{\mathcal{L}} = 1$ this reduces to the pairwise difference assorter for ‘simple’ Hamiltonian contests, where each ballot can only cast a single vote [1]. When $d = 0$ this reduces to the pairwise majority assorter in the more general context where we can have multiple votes per ballot.

4 Case study: 2016 Hesse local elections

In the local elections in Hesse, Germany, each ballot allows the voter to cast S direct votes, where S is the number of seats in the region. Each party can have at most S candidates on the ballot. Voters can assign up to three votes to individual candidates; they can spread these votes amongst candidates from different parties as they like. Voters can cross out candidates, meaning none of their votes will flow to such candidates. Finally a voter can select a single party. The effect of this selection is that remaining votes not assigned to individual candidates are given to the party. At the low level these votes are then spread amongst the candidates of the party (that have not been crossed out) by assigning one vote to the next (uncrossed out) candidate in the selected party, starting from the top, and wrapping around to the top once we hit the bottom, until all the remaining votes are assigned. Budurushi [3] provides a detailed description of the vote casting and vote tallying rules.¹⁷

Example 3. Consider a contest in a region with 12 seats, and a ballot with 4 parties. The Greens have five candidates appearing in the order Arnold, Beatrix, Charles, Debra, and Emma. Consider a ballot that has 3 votes assigned directly to Beatrix, Charles crossed out, three votes assigned directly to Fox (a candidate for another party), and the Greens party selected.

Since 6 votes are directly assigned, the Greens receive the remaining 6 votes. We start by assigning one vote of the 6 to the top candidate, Arnold, then one to Beatrix, none to Charles, one to Debra, one to Emma, another to Arnold, and another to Beatrix. In total, the ballot assigns 2 votes to Arnold, 5 to Beatrix, 1 to Debra, 1 to Emma, and 3 to Fox. \square

¹⁷ The description is based on the (German only) official information from Hesse, see <https://wahlen.hessen.de/kommunen/kommunalwahlen-2021/wahlsystem>, last accessed 24.07.2021.

The social choice function involves two stages. In the first stage, the entities we consider are the parties. This stage determines how many seats are awarded to each party. Each party is awarded the total votes assigned on a ballot to that party via individual candidates votes and the party selection remainder. There is a Hamiltonian election to determine the number of seats awarded to each party. Given S seats in the region, we award $s_e = \lfloor Sp_e \rfloor$ to each party $e \in E$. The remaining $k = S - \sum_{e \in E} s_e$ seats are awarded to the k parties with greatest remainders $r_e = Sp_e - s_e$. Let a_e be the total number of seats awarded to party e (which is either s_e or $s_e + 1$).

In the second stage, seats are awarded to individual candidates. For each party e awarded a_e seats, those a_e candidates in the party receiving the most votes are awarded a seat.

Performing a risk-limiting audit on a Hesse local election involves a number of assertions. The first stage is a Hamiltonian election. The assertions required to verify the result are described by Blom *et al.* [1]. For each pair of parties $m \neq n$ we need to test the assertion

$$p_m > p_n + \frac{a_m - a_n - 1}{S}, \quad n, m \in E, n \neq m. \quad (3)$$

While Blom *et al.* [1] define an assorter for this assertion, it is made under the assumption that each ballot contains a vote for at most one entity. The assorter defined in Section 3.1—with $A = m$, $B = n$ and $d = (a_m - a_n - 1)/S$ —is more general and allows for multiple votes per ballot.

These (All-Seats) assertions may require large samples to verify. We can verify a simpler assertion—that each party e deserved to obtain at least s_e seats—using the assertion $p_e > s_e/S$. We check this with an ‘All-But-Remainder’ audit.

The second stage of the election is a multi-winner first-past-the-post contest within each party: party e ’s a_e seats are allocated to the a_e individual candidates with highest tallies. An audit would require comparing each winner’s tally to each loser’s. The margins are often very small—the example data includes margins of only one vote—so these allocations are likely to require a full recount, and we have not included them in our simulations.

For experiments we consider a collection of 21 local district-based elections held in Hesse, Germany, on March 6, 2016. An ‘All-But-Remainder’ audit checks that each party e deserved the seats awarded to it in the first phase of distribution (s_e), excluding those assigned to parties on the basis of their ‘remainder’. An ‘All-Seats’ audit checks a_e , i.e. all of the seats awarded to party e , including their last seat awarded on the basis of their remainder (if applicable).

Across the 21 district contests in our case study, the number of seats available varied from 51 to 87, the number of parties from 6 to 11, and the number of voters from 39,839 to 157,100. For each assertion, we estimate the number of ballot checks required to audit it, assuming no errors are present between each paper ballot and its electronic record. Table 1 shows the number of ballot checks required to audit the most difficult assertion in each of these contests as the contest’s ASN (average sample number) for the two levels of auditing (All-But-Remainder and All-Seats). An ASN of ∞ indicates that a full manual recount

would be required. We record the ASN for risk limits, of 5% and 10%. The Kaplan–Kolmogorov risk function (with $g = 0.1$) was used to compute ASNs, given the margin for an assertion, following the process outlined in [Section 4.1](#).

[Table 1](#) shows that an All-Seats audit can be challenging in terms of the sample size required, but that an All-But-Remainder audit is usually quite practical. The estimated sample size required in an audit depends on the margin of each assertion being checked. Where these margins are small—for example, where two parties receive a similar remainder—the average sample size is likely to be large. This is an inherent property of the auditing, not a failure of our method. For example, the All-Seats audit for Limburg-Weilburg has an infinite ASN. The vote data shows why: the lowest remainder to earn an extra seat is the CDU Party’s, with a remainder of 24,267 votes; the highest remainder *not* to earn an extra seat is the FW Party’s, with 24,205 votes. An audit would need to check that the FW did not, in fact, gain a higher remainder than the CDU. However, a single ballot can contain up to 71 votes, so this comparison (and hence the electoral outcome) could be altered by a single misrecorded ballot. An electoral outcome that can be altered by the votes on one ballot requires a full manual count in any election system, regardless of the auditing method.

Even the All-Seats audit is quite practical when the margins represent a relatively large fraction of ballots. This is consistent with prior work ([\[1\]](#)) on US primaries, showing that an All-Seats audit is quite practical in that context.

4.1 Estimating an initial sample size using a risk function

We use the margin of the assorter for each assertion to estimate the number of ballot checks required to confirm that an assertion holds in an audit. As defined in [\[4\]](#), the margin for assertion A is 2 times its assorter mean, \bar{h}_A , minus 1.

Let V the total number of valid ballots and I be the total number of invalid ballots cast in the contest. Note that the sum $V + I$ may differ from the total number of votes, $T_{\mathcal{L}}$, since there may be multiple votes expressed on each ballot.

For an All-But-Remainder assertion indicating that party e received more than proportion t of the total vote, $T_{\mathcal{L}}$, the assorter mean is

$$\bar{h} = \frac{1}{V + I} \left(\frac{1}{2t} T_e - \frac{1}{2} T_{\mathcal{L}} + \frac{1}{2} (V + I) \right),$$

where T_e is the total number of votes for all candidates in party e . We compute t for a given assertion as follows:

$$q = \frac{T_{\mathcal{L}}}{S}, \quad \delta = \left\lfloor \frac{T_e}{q} \right\rfloor, \quad t = \frac{q\delta}{T_{\mathcal{L}}}.$$

For an All-Seats comparative difference assertion between two parties, A and B , we need to test a pairwise difference assertion where the difference is given by

$$d = \frac{(a_A - a_B - 1)}{S}.$$

The assorter mean for testing this assertion is given by

$$\bar{h} = \frac{1}{V + I} \left(\frac{T_A - T_B - T_{\mathcal{L}}d + VS \cdot (1 + d)}{2S \cdot (1 + d)} + \frac{I}{2} \right).$$

Once we have computed the assorter mean for an assertion, we use functionality from the SHANGRLA software implementation,¹⁸ using the Kaplan–Kolmogorov risk function with $g = 0.1$, and an error rate of 0.

Table 1. Estimates of audit sample sizes for each local district election held in Hesse on March 6th, 2016. We record the number of assertions to be checked in an All-But-Remainder and All-Seats audit, alongside the estimated number of ballot checks required to complete these audits for risk limits of 5% and 10%, assuming no discrepancies are found between paper ballots and their electronic records. S is the number of seats, $|\mathcal{L}|$ is the total number of ballots cast, $|E|$ is the total number of parties, and V is the total number of valid ballots. $|\mathcal{L}|$ and V are recorded to the nearest thousand.

District	S	$ \mathcal{L} $	$ E $	V	All-But-Rem.			All-Seats		
					RL 5%	RL 10%		RL 5%	RL 10%	
					$ \mathcal{A} $	ASN	ASN	$ \mathcal{A} $	ASN	ASN
Marburg-Biedenkopf	81	92k	8	88k	8	128	99	56	2,004	1,544
Fulder	81	95k	8	91k	8	27	20	56	34,769	28,142
Wetterau	81	122k	11	115k	11	26	20	110	12,570	9,790
Groß Gerau	71	85k	11	80k	11	291	224	110	7,844	6,101
Limburg-Weilburg	71	67k	7	64k	7	879	677	42	∞	∞
Kassel	81	100k	7	95k	7	1,180	909	42	4,580	3,540
Darmstadt-Dieburg	71	113k	8	107k	8	39	30	56	86,480	76,879
Bergstrasse	71	101k	9	96k	9	19	14	72	5,329	4,123
Werra-Meißner	61	45k	6	42k	6	8	6	30	3,252	2,522
Hersfeld-Rotenburg	61	52k	7	50k	7	29	23	42	5,173	4,026
Offenbach	87	119k	9	113k	9	35	27	72	25,691	20,323
Rheingau Taunus	81	78k	7	74k	7	27	21	42	4,382	3,392
Lahn-Dill	81	88k	8	83k	8	50	38	56	2,752	2,124
Waldeck-Frankenberg	71	65k	8	62k	8	234	180	56	1,508	1,162
Main-Taunus	81	95k	8	91k	8	66	51	56	23,669	18,808
Schwalm-Eder	71	82k	8	78k	8	24	18	56	35,724	29,301
Odenwald	51	40k	7	38k	7	74	57	42	933	719
Main-Kinzig	87	157k	10	148k	10	15	12	90	4,105	3,165
Landkreis Gießen	81	103k	8	98k	8	41	24	56	8,324	6,464
Hochtaunus	71	94k	8	90k	8	83	64	56	36,978	30,069
Vogelsberg	61	50k	7	47k	7	10	8	42	9,668	7,624

¹⁸ `TestNonnegMean.initial_sample_size()` from https://github.com/pbstark/SHANGRLA/blob/main/Code/assertion_audit_utils.py, last accessed 24.07.2021.

5 Example: Assorters for D’Hondt and related methods

Risk-limiting audits for D’Hondt and other highest averages methods were developed by Stark and Teague [5]. In this section we show how to express those audits in the form of assertions, and develop the appropriate assorters.

5.1 Background on highest averages methods

Highest averages methods are used by many parliamentary democracies in Europe, as well as elections for the European Parliament (which uses D’Hondt).¹⁹

Highest averages methods are similar to Hamiltonian methods in that they allocate seats to parties in approximate proportion to the fraction of the overall vote they won. They differ in how they allocate the last few seats when the voting fractions do not match an integer number of seats.

A highest averages method is parameterized by a set of divisors $d(1), d(2), \dots, d(S)$ where S is the number of seats. The seats are allocated by forming a table in which each party’s votes are divided by each of the divisors, then choosing the S largest numbers in the whole table—the number of selected entries in a party’s row is the number of seats that party wins. The divisors for D’Hondt are $d(i) = i, i = 1, 2, \dots, S$. Sainte-Laguë has divisors $d(i) = 2i - 1, \text{ for } i = 1, 2, \dots, S$.

Let $f_{e,s} = T_e/d(s)$ for entity e and seat s . The *Winning Set* \mathcal{W} is

$$\mathcal{W} = \{(e, s) : f_{e,s} \text{ is one of the } S \text{ largest}\}.$$

This can be visualised in a table by writing out, for each entity e , the sequence of numbers $T_e/d(1), T_e/d(2), T_e/d(3), \dots$, and then selecting the S largest numbers in the table. Each party receives a number of seats equal to the number of selected values in its row.

Like Hamiltonian methods, highest averages methods can be used in a simple form in which voters choose only their favourite party, or in a variety of more complex forms in which voters can express approval or disapproval of individual candidates. We deal with the simple case first.

5.2 Simple D’Hondt: Party-only voting

In the simplest form of highest averages methods, seats are allocated to each entity (party) based on individual entity tallies. Let W_e be the number of seats won and L_e the number of the first seat lost by entity e . That is:

$$W_e = \max\{s : (e, s) \in \mathcal{W}\}; \perp \text{ if } e \text{ has no winners.}$$

$$L_e = \min\{s : (e, s) \notin \mathcal{W}\}; \perp \text{ if } e \text{ won all the seats.}$$

If e won some, but not all, seats, then $L_e = W_e + 1$.

¹⁹ [https://www.europarl.europa.eu/RegData/etudes/BRIE/2019/637966/EPRS_BRI\(2019\)637966_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/BRIE/2019/637966/EPRS_BRI(2019)637966_EN.pdf), last accessed 24.07.2021.

The inequalities that define the winners are, for all parties A with at least one winner, for all parties B (different from A) with at least one loser, as follows:

$$f_{A,W_A} > f_{B,L_B}. \quad (4)$$

Converting this into the notation of [Section 3](#), expressing [Equation 4](#) as an linear assertion gives us, $\forall A \text{ s.t. } W_A \neq \perp, \forall B \neq A \text{ s.t. } L_B \neq \perp,$

$$T_A/d(W_A) - T_B/d(L_B) > 0.$$

From this, we define the proto-assorter for any ballot b as

$$g_{A,B}(b) = \begin{cases} 1/d(W_A) & \text{if } b \text{ is a vote for party } A, \\ -1/d(L_B) & \text{if } b \text{ is a vote for party } B, \\ 0 & \text{otherwise,} \end{cases}$$

$$\text{or equivalently } g_{A,B}(b) = b_A/d(W_A) - b_B/d(L_B)$$

where b_A (resp. b_B) is 1 if there is a vote for party A (resp. B), 0 otherwise. The lower bound is clearly $a = -1/d(L_B)$. Substituting into [Equation 2](#) gives

$$h_{A,B}(b) = \begin{cases} 1/2 [d(L_B)/d(W_A) + 1] & \text{if } b \text{ is a vote for party } A, \\ 0 & \text{if } b \text{ is a vote for party } B, \\ 1/2 & \text{otherwise.} \end{cases}$$

Note that order matters: in general, both $h_{A,B}$ and $h_{B,A}$ are necessary—the first checks that party A 's lowest winner beat party B 's highest loser; the second checks that party B 's lowest winner beat party A 's highest loser.

5.3 More complex methods: Multi-candidate voting

Like some Hamiltonian elections, many highest averages elections also allow voters to select individual candidates. A party's tally is the total of its candidates' votes. Then, within each party, the won seats are allocated to the candidates with the highest individual tallies. The main entities are still parties, allocated seats according to [Equation 4](#), but the assorter must be generalised to allow one ballot to contain multiple votes for various candidates.

The proto-assorter for entities (parties) $A \neq B \text{ s.t. } W_A \neq \perp,$ and $L_B \neq \perp,$ is very similar to the single-party case, but votes for each party (b_A and b_B) count the total, over all that entity's candidates, and may be larger than one.

$$g_{A,B}(b) = b_A/d(W_A) - b_B/d(L_B).$$

The lower bound is $-m/d(L_B)$, again substituting in to [Equation 2](#) gives

$$h_{A,B}(b) = \frac{b_A d(L_B)/d(W_A) - b_B + m}{2m}.$$

Note this reduces to the single-vote assorter when $m = 1$ ($b_A, b_B \in \{0, 1\}$).

6 Conclusion & future work

SHANGRLA reduces RLAs for many social choice functions to a canonical form involving ‘assorters.’ This paper shows how to translate general linear assertions into canonical assorter form for SHANGRLA, illustrated by developing the first RLA method for Hamiltonian free list elections and the first assertion-based approach for D’Hondt style elections.

We show that party-list proportional representation systems can be audited using simple assertions that are both necessary and sufficient for the reported outcome to be correct. In some settings, including in Hesse, elections are inherently expensive to audit because margins are frequently small, both between parties vying for the seats allocated by remainder, and between candidates in the same party.

There are social choice functions for which no set of linear assertions guarantees the reported winner really won, for instance, social choice functions in which the order of in which the votes are tabulated matters or that involve a random element. Some variants of Single Transferable Vote (STV) have one or the other of those properties.

Other variants of STV might be amenable to RLAs and to SHANGRLA in particular: the question is open. We conjecture that STV is inherently hard to audit. Although a sufficient set of conditions is easy to generate—simply check every step of the elimination and seat-allocation sequence—this is highly likely to have very small margins and hence to require impractical sample sizes. We conjecture that it is hard to find a set of conditions that imply an STV outcome is correct and that requires reasonable sample sizes to audit. Of course, this was also conjectured for IRV and turns out to be false.

References

1. Blom, M., Stark, P.B., Stuckey, P.J., Teague, V., Vukcevic, D.: Auditing Hamiltonian elections. arXiv **2102.08510** (2021), <https://arxiv.org/abs/2102.08510>
2. Blom, M., Stuckey, P.J., Teague, V.: RAIRE: Risk-limiting audits for IRV elections. arXiv **1903.08804** (2019), <https://arxiv.org/abs/1903.08804>
3. Budurushi, J.: Usable Security Evaluation of EasyVote in the Context of Complex Elections. Ph.D. thesis, Technische Universität Darmstadt, Darmstadt (February 2016), <https://tuprints.ulb.tu-darmstadt.de/5418/>
4. Stark, P.B.: Sets of half-average nulls generate risk-limiting audits: SHANGRLA. In: Bernhard, M., Bracciali, A., Camp, L.J., Matsuo, S., Maurushat, A., Rønne, P.B., Sala, M. (eds.) *Financial Cryptography and Data Security*. pp. 319–336. Springer International Publishing, Cham (2020)
5. Stark, P.B., Teague, V., Essex, A.: Verifiable European elections: Risk-limiting audits for D’Hondt and its relatives. *USENIX Journal of Election Technology and Systems (JETS)* **3**(1), 18–39 (Dec 2014)

Risk-limiting Audits: A Practical Systematization of Knowledge

Matthew Bernhard

VotingWorks

Abstract. Risk-limiting audits (RLAs) are broadly accepted as the gold standard for tabulation audits: when I count ballots using software, an RLA provides software-independent evidence that tabulation declared the correct winner. While there have been many advances in RLAs over the last 14 years, many of the underlying assumptions and practical applications of RLAs have gone unexamined. In this paper, I present a review of existing RLA techniques, providing a concise definition of an RLA, examining its underlying assumptions, and discussing how RLAs work in practice, all from the perspective of the maintainer of a popular RLA tool. I present several attacks which can cause RLAs to fail to provide evidence of the correctness of an election outcome. Finally, I provide discussion on the RLA's place in the landscape of election security and observations about the value proposition RLAs present.

1 Introduction

Post-election audits have long been a tool to evaluate election processes. They serve as an opportunity for election officials to check their work and provide transparency into election processes for voters. Additionally, post-election audits have long been a point of emphasis for election security, as they often provide robust means of achieving important properties like software independence [38]. In the wake of the 2016 presidential election in the United States, post-election audits, and risk-limiting audits in particular, have become widely adopted as a means of securing and improving the election process [13, 20, 27, 31, 35].

Much work has been conducted on post-election audits as a means of building evidence-based elections [52]. Stark produced the concept of a risk-limiting audit (RLA), a post-election tabulation audit which can provide a desired level of confidence in the election outcome with an amount of work that depends on the desired level of confidence [46].

Interwoven in much of the work around post-election audits are subtle assumptions and an implicit threat model. Most early efforts focused largely on the threat model of the election itself, with audits as a defense. While efficiency is a common theme, it is usually framed as pragmatism, not a feature of the audit to be safeguarded. Stark and Wagner contemplate ways in which the audit can be compromised, if, e.g., the paper trail is not properly maintained [52]; however they do not formally consider how a compliance audit can factor into the critical functions of an RLA.

In this paper I explore the assumptions underlying risk-limiting audits, discuss how they are conducted in practice, and what these things mean for the threat model for RLAs. I start by defining what an RLA is and its capabilities in the next section. I then examine how RLAs are performed in practice in Section 3. In Section 4 I describe attacks against RLAs that defeat their primary functions before concluding in Section 5.

2 What is a risk-limiting audit?

Broadly speaking, the goal of an RLA is to provide some degree of confidence in the outcome of an election while doing some amount of work. The trade-off is centrally between the level of confidence (framed as the “risk limit”) and the amount of work required to reach it (usually in terms of the number of ballots that need to be counted by hand). However, there are numerous assumptions baked into RLAs that are often not explicitly stated, though works like [8, 19, 52] have attempted to make these assumptions more concrete.

The core of an RLA lies with the generation and preservation of evidence for an election outcome [19, 28, 52]. If an election outcome is produced by the automatic tabulation of ballots, the tabulators may be compromised in such a way that is not obvious (e.g. they may flip a few votes in a close race to change the outcome). Absent efforts to check the tallies, this process does not provide sufficiently verifiable evidence that the election’s outcome is correct. This is the problem that RLAs attempt to address.

RLAs assume that there exists a “true” election outcome. Ideally the true election outcome would be produced by a social choice function based on the preferences of the voters. For instance, if first-past-the-post is the social choice function, and most voters prefer Alice over Bob, then the “true” outcome is that Alice wins. Unfortunately, political preferences are often inconsistent even within individuals [3], so elections must force people to commit to their preferences. This can be done in a variety of ways, but the most common form is via a secret ballot onto which voters imprint their preferences, typically by selecting from a provided list of choices (though occasionally something more complex). Since the voter’s choices on the ballot are assumed to align with their preferences, the ballot is the record of the voter’s intent, considered the “true” expression of each voter’s preferences. While some voting systems allow voters to mark ballots in a purely mechanical or electronic way, as on a direct-recording electronic voting machine (DRE), these methods do not as of yet produce a durable record of the vote that can be independently verified [10] (i.e. they aren’t software-independent [38]). Therefore, RLAs rely on *paper* ballots as the source of true preferences for an election outcome.

Assumption 1. Paper ballots reflect voters’ true preferences.

This assumption means that RLAs do not attempt to characterize mistakes voters may make when marking their ballots, even though mistakes may be prevalent enough to change the outcome [12, 41]. Nor do they account for any

other mechanism that can cause the paper record of the vote to not reflect the voters' preferences (or the choices voters make to reflect those preferences), like malicious software on ballot marking devices [1, 11, 24].

Another assumption that RLAs make is that the set of ballots being examined is complete. If the ballots examined in the RLA do not comprise of all of the valid ballots used to tabulate the election results, then an RLA's output is largely meaningless. This also means that RLAs require stringent chain-of-custody records to provide verifiable evidence that the paper trail is well maintained [52].¹

Assumption 2. The paper trail examined by the RLA is complete and correct.

To make an election process evidence-based and independently verifiable, RLAs rely on a means of tabulation that does not rely on software: hand counting. While research has found that humans do make mistakes when counting ballots [21], RLAs implicitly trust that hand counts are accurate. This assumption may be reasonable, as audits have a much greater degree of transparency and public visibility than software running on scanners, but has largely been unexamined in the literature. In summary, RLAs treat the result of hand counts as ground truth for the “real” election outcome. Since hand counts do not rely on software (or do not rely on software in a way that violates software independence; see Section 3), they are a software-independent mechanism for evidence evaluation. In other words, RLAs assume that a hand count of all the ballots in an election always produces the correct outcome [46].

Assumption 3. Full hand counts always produce “true” election outcomes.

Even though we now have a definition of what a true evidence-based election process would look like, RLAs rely on one more key assumption. If hand counts were truly the best method of conducting elections, then surely they would be the most widely adopted means of tabulating votes. However, this is obviously not the case in countries all over the world [2, 42, 57]. Therefore there must be benefits to software tabulation that outweigh its costs.² RLAs thus seeks to gain confidence that an election outcome is correct while not doing a full hand count, if possible.³

Assumption 4. Hand counting is undesirable and should be minimized.

We now have a clearly defined problem statement for RLAs:

Definition 1. Risk-limiting audits are an evidence-based process to provide a specified level of confidence in an election outcome while minimizing hand counting.

¹ This is not a given, as Enguehard et al. have documented, among others [17].

² Namely, cost, accuracy, speed, and repeatability of tabulation, though see, e.g., [53].

³ Though RLAs may also be performed on hand-tabulated elections as well [40].

3 Logistics of RLAs

Now that I have established what an RLA is attempting to accomplish, I can set about describing how it does so. There are several dimensions to performing an RLA that each introduce unique challenges to the threat model.

3.1 Hypothesis testing

RLAs draw a sample of ballots (typically uniformly at random [29] though occasionally not [5, 46]), hand count the sample, and then use the data produced by the hand count to evaluate the election result produced by software tabulation. There are numerous details here that I will discuss later on in this section, but this description suffices for now.

Given software-produced tabulation and our hand count data, an RLA seeks to assert or reject the hypothesis that the election outcome is correct. Specifically, RLAs usually take the software-produced outcome as an alternative hypothesis (though other alternatives are possible; see [23, 58]), and that the outcome is wrong as the null hypothesis. In this case, “the outcome is wrong” is formalized as the true outcome either preferring a candidate that did not win or a tie. These two hypotheses are mutually exclusive (though it may take some massaging depending on the social choice function) and exhaustive (by definition), so if the null hypothesis is rejected, then one can conclude to a degree of confidence that the alternative is true.

As this is a hypothesis test, the test can have four outputs depending on the underlying “true” hypothesis and the efficacy of the test. A **true positive** results when the null hypothesis is correctly rejected, i.e. the reported outcome is correct and the audit found evidence to support that to the specified risk limit. A **true negative** results when the reported election outcome disagrees with the outcome that would be produced by a full hand-tally. In this case, RLAs are designed to escalate to a full hand-tally before concluding that the reported election result is wrong. This feature makes a **false positive** effectively impossible: a correct reported outcome can never be overturned by an RLA [46], as outcomes can only be overturned by a full recount, at which point the true outcome will be known. A **false negative** results when the election outcome *should* be overturned, but the audit incorrectly terminates after fewer than all of the ballots have been audited. False negatives can occur by bad luck: if the sample data just happens to strongly favor the reported winner, the audit may conclude that the reported winner really did win even if that is not the case. The largest chance of this occurring is the **risk limit**.

The output of a hypothesis test is a **p-value** (occasionally called the risk measurement), an estimation of how likely the data would have occurred by random chance. If this value is below the risk-limit, the audit stops. If not, it draws more ballots and generates another p-value until all of the ballots have been counted or the p-value is below the risk-limit.

3.2 Evaluating hypotheses

In order to test the hypothesis that the reported election outcome is correct, RLAs require a means of evaluating evidence. There are two primary ways paper ballots are used to evaluate the reported outcome: **polling** audits and **comparison** audits [29]. There are also two **units** that audits can operate over: individual ballots or batches of ballots.

In a polling audit, a sample of ballots is drawn, and the social choice function is computed over the sample. The resulting tally is then used to evaluate the null and alternative hypotheses [26, 33, 51, 58]. Polling audits typically require dramatically larger sample sizes than comparison audits, as the evaluation being done tends to be less sensitive. However, polling audits require significantly less infrastructure to perform than comparison audits [29, 58].

Comparison audits, rather than relying on the social choice function to evaluate hypotheses, directly evaluates the evidence generated by the election. Each ballot (or batch) is audited, and the result of the audit is compared to the record of how that ballot (or batch) was tabulated. If discrepancies are found between the audited ballot and the reported tabulation, this indicates that an outcome-changing event may have occurred [22, 46, 48–51].

Finally, there are also *hybrid* auditing methods, most notably SUITE [34] and SHANGRLA [51]. These audits break the set of ballots into separate *strata*, where different types of evaluation (polling and comparison), different units of auditing (ballot or batch), or both, are used in each stratum.

3.3 Collecting evidence

In order to collect evidence to support or refute a hypothesis, RLAs must draw a sample from a population of ballots. We may not trust a reported election result, but at a minimum we must know how many paper ballots there are (i.e. how large the population is) and where they are (i.e. how can a specific ballot be found) in a software independent way.

The first piece of data, the population size, can be determined several ways, for example examining the voter check-in data to see how many voters checked in to vote (though this data must be verified against the chain of custody of paper ballots). The latter requires a bit more work depending on the storage procedures for the ballots. We can imagine a warehouse containing all of the ballots in a contest, where ballots are stored in boxes. In order to audit ballot n , then, we need to know which box it is contained in. This data is called a **ballot manifest**, and includes a software-independent accounting of ballot storage [29, 52]. As we shall see in Section 4, errors in the ballot manifest can cause the RLA to fail to achieve its goals.

Now that we have established what our population is, we can sample from it. There are a variety of sampling techniques available. The simplest and most common is sampling uniformly at random after the results have been fully tabulated and reported. However, Ottoboni et al. examined sampling uniformly before the results have been fully tabulated [33]. Sampling especially in batch

audits may be done with respect to the amount of possible “error” that can be found in a unit [5, 47].

To generate a sample, the manifest is used to create a “master list” of ballots (or batches), where each ballot gets a sequential number (e.g. if the last ballot in the first box is ballot n , then the first ballot in the next box is $n + 1$ and so on). In principle, ballots can be sampled at random by merely rolling enough dice to cover the range of ballot indices, modulo the population size [15]. In practice, this is far too laborious, so pseudo-random number generating programs (PRNGs) are often used to generate a list of ballots to sample [15, 33].

Once a ballot is sampled, the manifest is used to figure out which container that ballot is stored in, and a variety of methods can be used to find the specific ballot in question (counting down in the stack of ballots or using a counting scale, for example [36]). Sridhar et al. proposed an alternative method for sampling an individual ballot from a batch, called k-cut: cutting the stack of ballots like a deck of cards. They demonstrated that cutting the stack of ballots six times in randomly generated places was sufficient to get a uniformly sampled ballot [43].⁴ Batch-level audits do not require these techniques, as once a batch is sampled all of the ballots it contains are part of the sample and are hand counted.

3.4 Examining the evidence

Once the ballots to be examined have been drawn, RLAs require a means of evaluating the evidence. The nature of this evaluation depends heavily on the specific hypothesis being tested. BRAVO ballot polling audits examine hypotheses about how the margin is distributed *proportionally* to each candidate [26, 33, 34, 58], while the type of ballot polling described in [34, 51] looks at the margin in terms of integer votes. In both cases, ballot polling audits test the hypothesis that the winner actually won by examining the margin in the sample against the null hypothesis (the winner didn’t win; in most cases the margin under the null is 0, i.e. a tie) and an alternative hypothesis (typically the reported margin, but occasionally not [23]). Most extant ballot polling audits examine the likelihood ratios of the null and the alternative given the tally of a sample [26, 29, 32, 37, 55, 58], though methods like [39, 51] use other test statistics. In any case, these audits require very little information to evaluate their hypotheses: the null margin, alternative margin, and a hand tally of the sampled ballots. If their test-statistic outputs a p-value below the risk limit, the audit stops. Otherwise, more ballots are sampled.

Comparison style audits also examine the margin in votes, but rather than examine a tally of the votes in the sample, comparison audits directly compare the sampled units against their reported counterparts. If a ballot’s audited record differs from its reported record, this is deemed an error. Errors can work in multiple

⁴ K-cut may not be suitable for ballot comparison audits, since they need the ability to look up a sampled ballot’s cast vote record (see below). If the ballots do not have an identifier (imprinted on the ballot at the time of scanning), the ballots must be kept in the order they were scanned.

ways: **overstatements** are errors which cause the margin to be overstated, i.e. the winner didn't win by as much as was reported. **Understatements** are errors which indicate that the winner actually won by more than originally reported. If the number of overstatements is larger than the margin of victory, then the reported outcome is wrong.

In a ballot comparison audit, if a ballot is sampled, it is examined and the votes it contains are directly compared to how it was originally tabulated [46, 50, 51]. These audits need an additional input: a record of how each individual ballot was tabulated (called a **cast vote record** or CVR) that can be compared against an examination of the corresponding paper ballot.

For batch comparison audits, the hand-tallied batch totals are compared to the tabulator results for each batch. Rather than a CVR, batch comparison audits require a batch totals file that contains how each batch was tabulated by software, which is then compared to hand tallies. It should be noted that ballot comparison audits are just special cases of batch comparison audits where each batch contains only one ballot [22, 48].

3.5 Tools for performing an RLA

Much of the logistical support and calculation in RLAs can in principle be done without reliance on any software. However, this is practically infeasible in most contexts (e.g. if an RLA needs to draw 5,000 ballots, rolling enough dice to draw a sample would take an inordinate amount of time). Moreover, election officials (or other auditors) may not have the expertise in statistics required to perform the calculations necessary to evaluate evidence. Finally, coordinating an audit in a large jurisdiction, like a state, becomes intractable when dozens or hundreds of ballot manifests, CVRs, and other election information must be collated.

To address these problems, numerous software tools have been developed to support the logistics of an audit. In order to preserve software-independence, the inputs and outputs of the software need to be made available. Publishing this data enables anyone to independently verify that the software conducted the audit correctly (either by working it through by hand or by using different, trusted software). To my knowledge, all existing RLA software is open source, which aids in the verifiability of the software's correctness and also supports independent evaluation of audits.

Tools for performing ballot polling audits have been made available by Stark [44] and VotingWorks [56]. Additional ballot polling software has been made available by Ottoboni et al. [33], Stark [51], and Zagorski et al. [58]. Tools for performing comparison audits have been made available by Stark [45, 51], VotingWorks [56], and Free and Fair [18]. McBurnett also has a tool for older batch comparison audits including SAFE [30], NEGEXP [5], and PPEB [47]. Finally, tools for performing hybrid audits have been provided by Ottoboni et al. [34] and VotingWorks [56].

As we shall see in the next section, these tools encompass a variety of communications channels between participants in the audit, and can present risk.

3.6 How does an RLA actually work?

We have covered much of the inputs and outputs of RLAs so far, but I have not described the process in full nor identified the stakeholders.

Stakeholders RLAs are fundamentally transactions between evidence havers and evidence seekers. Election officials are in charge of the software that does the tabulation as well as the paper trail. For large audits, there are often two types of election official: **audit administrators**, in charge of coordinating the data collection and entry procedures, and **local election officials** who have custody of the paper trail and do the work of pulling and examining ballots. The election officials rely on **audit software** to conduct the audit. The **public** observes the audit and seeks to be convinced that the paper ballots match the reported election outcome. Finally, **attackers** seek to disrupt this process.

Process The RLA process is sequential, and is described as follows (and depicted in Figure 1):

1. The election ends, a final tally is produced, and an outcome is declared.
2. Audit administrators initialize the audit with information about the contest to be audited (the contest name, number of winners, candidates, tally of votes for each candidates, and outcome), as well as a list of authorized participants if the audit is happening in multiple places and involves distributed access to the tool (local election officials who possess the paper ballots).
3. Local election officials collect and submit all necessary inputs, including a ballot manifest and, if performing a comparison audit, batch totals or CVRs.
4. A ceremony for initializing a PRNG is conducted, where some physical randomness (e.g. dice) is used to seed the PRNG.
5. Based on the contest information (the margin of victory) and previous samples (if not in the first round of the audit), a sample size is estimated (i.e. the number of ballots or batches to audit is determined).
6. The ballots to be sampled are identified by the PRNG, and information about which ballots or batches are to be examined (called a retrieval list) is distributed to local election officials.
7. Local election officials retrieve the ballots in their jurisdiction, examine them, and submit the information to the audit administrators (either out of band or through the audit software).
8. Once all information about the sample is entered, the audit administrator computes a p-value for sample based on the evidence. If the p-value is greater than the risk limit, go back to step 5, generate a new sample and collect more evidence. If all ballots have been sampled, stop and announce the outcome of the sampled data as the new outcome. If the p-value is less than or equal to the risk limit, halt the audit and announce that the outcome is confirmed.
9. At the conclusion of the audit, all information about the audit is published, including the manifests, CVRs, the random seed and PRNG algorithm, and adjudication of the sampled ballots so that anyone may verify the results.

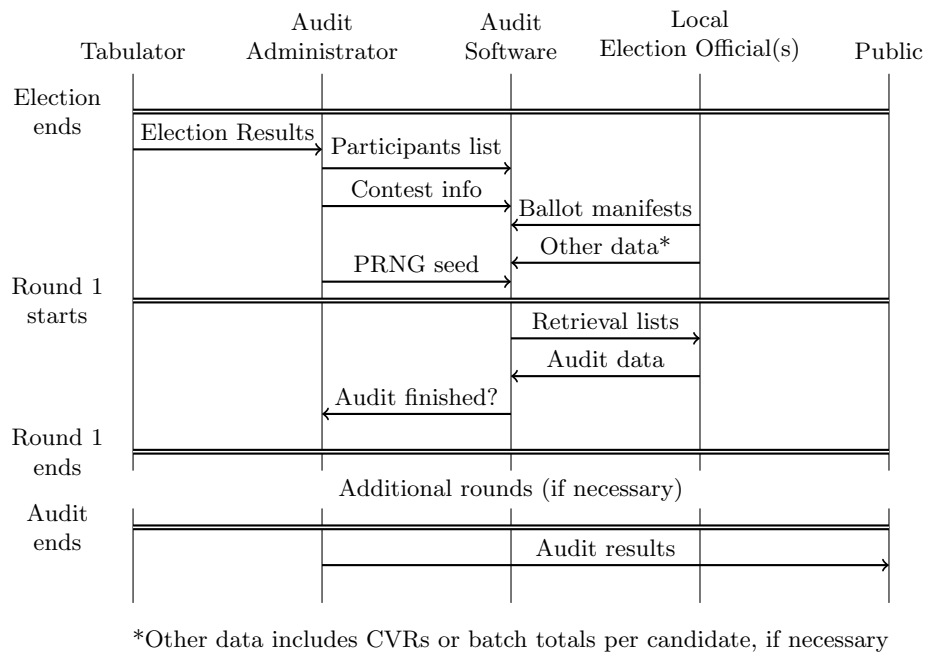


Fig. 1. A process diagram of an RLA—Each column represents a stakeholder. Arrows between stakeholders indicate communication of some sort between two stakeholders. At the start of each round, the audit tool computes a sample size and draws a sample of ballots, which are distributed to local election officials. Having received their retrieval lists, LEOs sample ballots as indicated by the retrieval list and enter the audit data into the audit software. After all ballots in round have been entered, the software computes a p-value for the audit so far, and notifies the audit administrator whether the audit has met the risk limit, or if more ballots need to be sampled. Note that the depicted protocol is an insecure strawman, and the communication channels must be secured and the data must be made public for the audit to be successfully verifiable (see Section 4.5).

4 Attacks and Defenses

An overview of the audit process and relevant communication channels can be seen in Figure 1. As we shall see shortly, the way the process is laid out presents several opportunities for an attacker to compromise the audit. Revisiting our assumptions about RLAs from Section 2, we can define a major and a minor goal for our attacker:

- **Major goal:** Cause an RLA to stop early even though a full hand count would indicate a different result more frequently than the risk limit allows.
- **Minor goal:** Force the RLA to a full hand count.

Recall from Section 3.1 that the hypothesis testing at the core of an RLA can indicate four distinct outcomes. The “true” outcomes are where an RLA does what it is supposed to: either correctly stop after seeing sufficient evidence or correctly escalating to a full recount and overturning an election. An attacker seeks to cause a “false” outcome: either incorrectly concluding that the reported outcome was correct or overturning a correct result. As we discussed earlier, the latter outcome (a false positive) is impossible if the paper trail is preserved, so an attacker can at best cause a false negative (accepting an incorrect result).

This last point is our attacker’s major goal: cause a false negative with probability greater than the risk limit. This might occur if the attacker had also compromised the tabulator and wished to remain undetected.

Additionally, recall that RLAs also assume that hand counting all of the ballots are undesirable. Therefore a minor goal for an attacker is to cause a full recount every time an RLA is performed (intended to waste resources).

In general, an attacker who can accomplish the major goal can also accomplish the minor goal, but not the other way around. Additional goals like sowing public distrust are considered out of scope for now. With these goals in mind, we present several attack scenarios.

4.1 An active network attacker

In this scenario, the attacker is able to intercept, alter, and transmit messages to any stakeholders. This attacker cannot alter the paper trail, audit tool, or PRNG generation ceremony. However, they can impersonate all parties and use the audit tool to generate expected output.

Forged sample sizes This attacker can forge the expected sample sizes and cause them to be larger or smaller than required. This achieves the minor goal, as they could indicate that the audit needs to examine all of the ballots.

Forged retrieval lists This attacker can issue forged retrieval lists directly to local elections officials (LEOs). In doing so, they can bias the sample by selecting ballots from jurisdictions that lean more towards the declared winner. If the audit samples these ballots, it can falsely conclude that the declared result was correct.

Forged audit results This attacker can similarly forge messages from the LEOs to the audit tool and input fake audit data that causes the software to convince that the audit can terminate.

Forged messages from the audit administrator This attacker can announce audit results to the public early or otherwise in a way that sows doubt about the correctness of the election process.

This attacker’s power largely comes from the lack of out-of-band communication between stakeholders. To mitigate this attack, an authenticated channel should be established between all parties to ensure that only authenticated parties are submitting and generating audit data. Standard authentication techniques, like TLS, and the use of restricted domains and authenticated email accounts [54] apply here. Furthermore, independent verification of the audit would catch many

of these attacks, as the sample sizes, retrieval lists, and results are deterministic and can be checked.

4.2 Compromised audit software

An attacker can still gain full control over the audit if they compromise the audit software, even without actively attacking the RLA’s communication channels. All of the forgery possibilities above apply, except announcing the result. However, presuming an honest audit administrator, if the tool provides an incorrect audit result the audit administrator may just announce the audit result provided by the tool without verification.

In order to defend against a compromised audit tool, software-independent bookkeeping is critical. The random seed, PRNG algorithm, ballot manifests, and audit results must be published through a channel that doesn’t rely on the tabulator or the audit software (e.g. by scans of paper documents uploaded to the audit administrator’s website) [19,25,52]. With this data (and CVRs or batch totals files), in principle anyone in the public can verify the audit results by hand or by using different, trusted software.

4.3 Compromised audit administrator

In this case, the audit administrator is compromised, either because they themselves wish to confirm an incorrect outcome or because malware resident on the device they’re using to interact with the audit software does. The defenses from Section 4.2 protect against this except for one attack on comparison audits.

If the audit administrator announces wrong election results and tampers with the data provided to the auditing software, it is possible for the audit to reject the null hypothesis when it shouldn’t. By entering incorrect contest totals but providing the correct tabulation data needed for comparison audits, the audit will confirm the outcome even though the paper ballots, if counted by hand, would result in a different outcome. This is because comparison audits assume that the contest data is correctly entered, and that any discrepancies between the audited ballots and the tabulation data will catch incorrect contest data. By declaring the wrong result and entering the wrong vote totals, but using uncompromised CVRs or batch totals files, the comparison audit will terminate without seeing discrepancies, “confirming” that the announced outcome indeed corresponds to the paper ballots. However, this will not be the case.

In order to prevent this attack, the contest data and the CVRs or batch totals must be made public so that the contest data can be verified against the CVRs or batch totals. However, in ballot comparison audits, published CVRs can enable coercion (the so-called Sicilian attack, for example [10]). For this reason, states like Colorado explicitly forbid publishing CVRs in plain text [14]. Therefore, CVR values need to be hidden (except for audited ballots). To achieve this, methods like [8] split up CVRs per-contest and produced commitments, while [9] homomorphically encrypts CVRs and publishes the encrypted CVRs and a decryption key for the CVR totals before the audit starts. Anyone can

then sum the encrypted CVRs and decrypt the result to verify that the contest data entered by the audit administrator matches the files to be compared. The ballots which are audited also have their plaintext CVR published, which allows the public to verify that the audit performed as expected.⁵

4.4 Compromised local election official

A compromised local election official represents the greatest threat to an RLA. LEOs are the custodians of the paper trail, so if they desire to subvert the audit, they can do so trivially by tampering with paper ballots. However, RLAs assume that sufficient compliance auditing [52] is performed (see Section 2). Still, LEOs (or malware on their behalf, below LEO refers to both cases) may subvert the audit in a few other ways.

Forged manifests If an LEO or collection of LEOs misrepresents how many ballots they have in their custody, they can influence the result of the RLA. For instance, an LEO in a jurisdiction that votes more for the winner can claim more ballots than they actually have. In a comparison audit, this would cause the audit to recount all of the ballots, as ballots claimed to exist but not found are counted as the worst kind of error [6]. In a ballot polling audit using k-cut, things are even worse, as unless the number of ballots is counted ahead of time and found to be too small, samples can be drawn without noticing anything is wrong. This is why it is critical for LEOs to produce software independent ballot manifests that can be published out of band and independently verified.

Forged audit data LEOs may also enter incorrect data about the audited ballots (as in the network attacker above). This can cause the audit to finish early. Ideally, public observation would prevent this, and paper records about the audit can be made available for verification.

4.5 Summary of defenses

I have detailed a wide variety of attacks, and mentioned some defenses in passing:

- Authenticated communication channels between all parties, including authenticated access to the audit tool
- Publication of all audit data through an authenticated channel, like the audit administrator’s website
- Commitment to audit data before use (e.g. the contest information should be known before the audit starts and verifiably not changed afterwards)
- Use of CVR encryption, where applicable [9]
- Software independent record keeping of audit data [19]
- Public observation of all audit processes
- Compliance audits that enforce chain of custody for the paper trail [52]
- External verification of all audit inputs and outputs using a trusted audit tool

⁵ An implementation of [9] can be found here: [4].

5 Conclusion

In this paper I have discussed the theoretical and practical facets of risk-limiting audits. I have examined the underlying assumptions made by RLAs and identified how those play out in practice. I have shown that numerous attacks against RLAs exist which can cause an RLA to fail to accomplish its primary goals, and also presented several defenses which mitigate these attacks.

Despite my analysis, I have largely deferred a critical consideration. RLAs exist to provide trust in election outcomes. They rely on a durable evidence trail, large-scale transparency processes, and at times complex statistics to provide proof that a reported outcome really does match all available evidence. However, it might be worth considering whether they succeed. Absent any attack, even if a risk-limiting audit is carried out faithfully and correctly, does it make voters more confident in election outcomes? Future research is needed on this point, as after the 2020 election, more U.S. states performed risk-limiting audits than ever before, all of which found significant evidence that the election outcome was correct. Yet, a significant fraction of the population still maintains that the reported outcome was wrong [7]. If RLAs don't provide confidence, is there value in them at all?

Recent events in the U.S. state of Georgia may provide some clarity on the issue. Georgia is a historically Republican state, with all executive offices held by Republicans, including the chief elections officer Brad Raffensperger. In an environment where much of the Republican party refused to accept the election result, going so far as to vote against certifying it in the U.S. federal legislative bodies, the Republican executives in Georgia stood by their election results, in which a Democratic presidential candidate won the state of Georgia for the first time in 28 years. Secretary Raffensperger repeatedly cited Georgia's risk-limiting audit as a major reason for his confidence [20]. Even if RLAs may not be intelligible or convincing to the public at large, there is significant value in the fact that they provide election officials (and interested members of the public) confidence in the election results. In a time where misinformation is wreaking havoc the world over [16], RLAs bolster the bastion of democracy.

Acknowledgments Thanks to Mark Lindeman, Monica Childers, Ben Adida, Kevin Skoglund, Dan Wallach and the anonymous reviewers for their insightful feedback.

References

1. Appel, A.W., DeMillo, R.A., Stark, P.B.: Ballot-marking devices cannot ensure the will of the voters. *Election Law Journal: Rules, Politics, and Policy* **19**(3), 432–450 (2020)
2. Aranha, D.F., van de Graaf, J.: The good, the bad, and the ugly: two decades of e-voting in Brazil. *IEEE Security & Privacy* **16**(6), 22–30 (2018)
3. Arcuri, L., Castelli, L., Galdi, S., Zogmaister, C., Amadori, A.: Predicting the vote: Implicit attitudes as predictors of the future behavior of decided and undecided voters. *Political Psychology* **29**(3), 369–387 (2008)

4. Arlo CVR Encryption. <https://github.com/votingworks/arlo-cvr-encryption>
5. Aslam, J., Popa, R., Rivest, R.: On Auditing Elections When Precincts Have Different Sizes. In: 2008 USENIX/ACCURATE Electronic Voting Technology Workshop, San Jose, CA, 28–29 July (2008), http://www.usenix.org/event/evt08/tech/full_papers/aslam/aslam.pdf, retrieved 30 May 2011
6. Banuelos, J.H., Stark, P.B.: Limiting risk by turning manifest phantoms into evil zombies. arXiv preprint arXiv:1207.3413 (2012)
7. Barry, D., McIntire, M., Rosenberg, M.: ‘Our President Wants Us Here’: The Mob That Stormed the Capitol. *New York Times* (2021), accessed from <https://www.nytimes.com/2021/01/09/us/capitol-rioters.html>
8. Benaloh, J., Jones, D., Lazarus, E., Lindeman, M., Stark, P.: SOBA: Secrecy-preserving Observable Ballot-level Audits. In: Proceedings of the 2011 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (EVT/WOTE ’11). USENIX (2011), <http://statistics.berkeley.edu/~stark/Preprints/soba11.pdf>
9. Benaloh, J., Stark, P.B., Teague, V.: VAULT: Verifiable Audits Using Limited Transparency. *E-Vote-ID 2019* p. 69 (2019)
10. Bernhard, M., Benaloh, J., Halderman, J.A., Rivest, R.L., Ryan, P.Y., Stark, P.B., Teague, V., Vora, P.L., Wallach, D.S.: Public evidence from secret ballots. In: International Joint Conference on Electronic Voting. pp. 84–109. Springer (2017)
11. Bernhard, M., McDonald, A., Meng, H., Hwa, J., Bajaj, N., Chang, K., Halderman, J.A.: Can Voters Detect Malicious Manipulation of Ballot Marking Devices? In: 2020 IEEE Symposium on Security and Privacy (SP). pp. 679–694. IEEE (2020)
12. Byrne, M.D., Greene, K.K., Everett, S.P.: Usability of voting systems: Baseline data for paper, punch cards, and lever machines. In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 171–180 (2007)
13. Colorado Secretary of State: Colorado Secretary of State Jena Griswold Certifies the State’s 2020 General Election. <https://www.sos.state.co.us/pubs/newsRoom/pressReleases/2020/PR20201208CertifyElectionResults.html>
14. Colorado Secretary of State: Rule 25. Post-election audit. In: Election Rules, chap. 25.2.4. Colorado Secretary of State (2021), accessed from https://www.sos.state.co.us/pubs/rule_making/CurrentRules/8CCR1505-1/Rule25.pdf
15. Cordero, A., Wagner, D., Dill, D.: The role of dice in election audits—extended abstract. In: IAVoSS Workshop on Trustworthy Elections. Citeseer (2006)
16. Del Vicario, M., Bessi, A., Zollo, F., Petroni, F., Scala, A., Caldarelli, G., Stanley, H.E., Quattrociocchi, W.: The spreading of misinformation online. *Proceedings of the National Academy of Sciences* **113**(3), 554–559 (2016)
17. Enguehard, C., Graton, J.D.: Machines à voter et élections politiques en France: étude quantitative de la précision des bureaux de vote. *Cahiers Droit, Sciences & Technologies* (4), 159–198 (2014), original in French. Translated via Google.
18. Free, Fair: ColoradoRLA. <https://github.com/FreeAndFair/ColoradoRLA>
19. Garland, L., Lindeman, M., McBurnett, N., Morrell, J., Schneider, M.K., Singer, S.: Principles and best practices for post-election audits. <https://verifiedvoting.org/publication/principles-and-best-practices-for-post-election-tabulation-audits/> (2018)
20. Georgia Secretary of State: Historic First Statewide Audit of Paper Ballots Upholds Result of Presidential Race. https://sos.ga.gov/index.php/elections/historic_first_statewide_audit_of_paper_ballots_upholds_result_of_presidential_race
21. Goggin, S.N., Byrne, M.D., Gilbert, J.E.: Post-election auditing: effects of procedure and ballot type on manual counting accuracy, efficiency, and auditor satisfaction and confidence. *Election Law Journal: Rules, Politics, and Policy* **11**(1), 36–51 (2012)

22. Hall, J.L., Miratrix, L.W., Stark, P.B., Briones, M., Ginnold, E., Oakley, F., Peaden, M., Pellerin, G., Stanionis, T., Webber, T.: Implementing risk-limiting post-election audits in California. In: EVT/WOTE 2009. USENIX Association (2009)
23. Huang, Z., Rivest, R.L., Stark, P.B., Teague, V.J., Vukcevic, D.: A unified evaluation of two-candidate ballot-polling election auditing methods. In: International Joint Conference on Electronic Voting. pp. 112–128. Springer (2020)
24. Kortum, P., Byrne, M.D., Whitmore, J.: Voter Verification of BMD Ballots Is a Two-Part Question: Can They? Mostly, They Can. Do They? Mostly, They Don't. *Election Law Journal: Rules, Politics, and Policy* (2020)
25. Lindeman, M., Halvorson, M., Smith, P., Garland, L., Addona, V., McCrea, D.: Principles and best practices for post-election audits (2008)
26. Lindeman, M., Stark, P., Yates, V.: BRAVO: Ballot-polling risk-limiting audits to verify outcomes. In: Proceedings of the 2011 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (EVT/WOTE '11). USENIX (2012)
27. Lindeman, M.: Rhode Island presidential risk-limiting audit, November 19-24, 2020 (brief report). https://elections.ri.gov/publications/Election_Publications/RLA/Rhode%20Island%20presidential%20RLA%20brief%20report.pdf
28. Lindeman, M., Halvorson, M., Smith, P., Garland, L., Addona, V., McCrea, D.: Principles and Best Practices for Post-Election Audits (Sep 2008), <http://electionaudits.org/files/bestpracticesfinal.0.pdf>
29. Lindeman, M., Stark, P.B.: A gentle introduction to risk-limiting audits. *IEEE Security & Privacy* **10**(5), 42–49 (2012)
30. McCarthy, J., Stanislevic, H., Lindeman, M., Ash, A., Addona, V., Batcher, M.: Percentage-based versus SAFE vote tabulation auditing: a graphic comparison. *The American Statistician* **62**(1), 11–16 (2008)
31. Michigan Secretary of State: Statewide risk-limiting election audit process to begin at 11 a.m. <https://www.michigan.gov/som/0,4669,7-192-47796-549191--,00.html>
32. Morin, S., McClearn, G., McBurnett, N., Vora, P.L., Zagórski, F.: A Note on Risk-Limiting Bayesian Polling Audits for Two-Candidate Elections (2020)
33. Ottoboni, K., Bernhard, M., Halderman, J.A., Rivest, R.L., Stark, P.B.: Bernoulli ballot polling: a manifest improvement for risk-limiting audits. In: Financial Cryptography and Data Security: FC 2019 International Workshops, VOTING and WTSC, St. Kitts, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers. pp. 226–241. Springer (2019)
34. Ottoboni, K., Stark, P.B., Lindeman, M., McBurnett, N.: Risk-Limiting Audits by Stratified Union-Intersection Tests of Elections (SUITE). In: International Joint Conference on Electronic Voting. pp. 174–188. Springer (2018)
35. Pennsylvania Secretary of State: Risk-Limiting Audit Pilot of November 2020 Presidential Election Finds Strong Evidence of Accurate Count. <https://www.media.pa.gov/pages/State-details.aspx?newsid=453>
36. Rhode Island Risk Limiting Audit Working Group: Pilot Implementation Study of Risk-Limiting Audit Methods in the State of Rhode Island, <https://www.verifiedvoting.org/report-on-rhode-island-risk-limiting-audit-pilot-implementation-study-released/>
37. Rivest, R.L., Shen, E.: A Bayesian method for auditing elections. In: USENIX Electronic Voting Technology Workshop / Workshop on Trustworthy Elections. EVT/WOTE '12 (Aug 2012), https://www.usenix.org/system/files/conference/evtvote12/rivest_bayes_rev_073112.pdf
38. Rivest, R.: On the notion of 'software independence' in voting systems. *Phil. Trans. R. Soc. A* **366**(1881), 3759–3767 (October 2008)

39. Rivest, R.: ClipAudit: A Simple Risk-Limiting Post-Election Audit (2017), <https://arxiv.org/abs/1701.08312>
40. Schürmann, C.: A risk-limiting audit in Denmark: A pilot. In: International Joint Conference on Electronic Voting. pp. 192–202. Springer (2016)
41. Sled, S.M.: Vertical proximity effects in the California Recall Election (2003)
42. Springall, D., Finkenauer, T., Durumeric, Z., Kitcat, J., Hursti, H., MacAlpine, M., Halderman, J.A.: Security analysis of the Estonian internet voting system. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. pp. 703–715 (2014)
43. Sridhar, M., Rivest, R.L.: k-Cut: A Simple Approximately-Uniform Method for Sampling Ballots in Post-election Audits. In: International Conference on Financial Cryptography and Data Security. pp. 242–256. Springer (2019)
44. Stark, P.B.: Tools for Ballot-Polling Risk-Limiting Election Audits. <https://www.stat.berkeley.edu/~stark/Vote/ballotPollTools.htm>
45. Stark, P.B.: Tools for Comparison Risk-Limiting Election Audits . <https://www.stat.berkeley.edu/~stark/Vote/auditTools.htm>
46. Stark, P.B.: Conservative statistical post-election audits. *Ann. Appl. Stat.* **2**(2), 550–581 (2008)
47. Stark, P.B.: Election audits by sampling with probability proportional to an error bound: dealing with discrepancies (2008)
48. Stark, P.B.: CAST: Canvass audits by sampling and testing. *IEEE Transactions on Information Forensics and Security* **4**(4), 708–717 (2009)
49. Stark, P.B.: Efficient post-election audits of multiple contests: 2009 California tests. In: CELS 2009 4th annual conference on empirical legal studies paper (2009)
50. Stark, P.B.: Super-simple simultaneous single-ballot risk-limiting audits. In: Proceedings of the 2010 international conference on Electronic voting technology/workshop on trustworthy elections. pp. 1–16 (2010)
51. Stark, P.B.: Sets of half-average nulls generate risk-limiting audits: SHANGRLA. In: International Conference on Financial Cryptography and Data Security. pp. 319–336. Springer (2020)
52. Stark, P.B., Wagner, D.: Evidence-based elections. *IEEE Security & Privacy* **10**(5), 33–41 (2012)
53. Stein, R.M., Mann, C., Stewart III, C., Birenbaum, Z., Fung, A., Greenberg, J., Kawsar, F., Alberda, G., Alvarez, R.M., Atkeson, L., et al.: Waiting to vote in the 2016 presidential election: Evidence from a multi-county study. *Political Research Quarterly* **73**(2), 439–453 (2020)
54. The Cybersecurity and Infrastructure Security Agency (CISA): Leveraging the .gov Top-level Domain. <https://www.cisa.gov/sites/default/files/publications/cisa-leveraging-the-gov-top-level-domain.pdf>
55. Vora, P.L.: Risk-Limiting Bayesian Polling Audits for Two Candidate Elections. arXiv preprint arXiv:1902.00999 (2019)
56. VotingWorks: Arlo: Open-source risk-limiting audit software by VotingWorks. <https://github.com/votingworks/arlo>
57. Wolchok, S., Wustrow, E., Halderman, J.A., Prasad, H.K., Kankipati, A., Sakhamuri, S.K., Yagati, V., Gonggrijp, R.: Security analysis of India’s electronic voting machines. In: Proceedings of the 17th ACM conference on Computer and communications security. pp. 1–14 (2010)
58. Zagórski, F., McClearn, G., Morin, S., McBurnett, N., Vora, P.L.: The Athena Class of Risk-Limiting Ballot Polling Audits. arXiv preprint arXiv:2008.02315 (2020)

Managing Election Integrity

Retaining Election Officials is Imperative to Secure Future Elections

Recommendations for State and Local Governments and Their Election Officials

David Levine¹[0000-1111-2222-3333]

¹ The German Marshall Fund of the United States, 1744 R St NW, Washington, DC 20009

Abstract. This paper summarizes the countless ways many election administrators have been attacked during the 2020 U.S. presidential election and its aftermath and lays out ten specific recommendations for how state and local governments can better protect and, consequently, retain these administrators to ensure the security of future elections.

Keywords: election integrity, election security, human resources, critical infrastructure, employee retention, state government, local government

1 Introduction

The 2020 presidential election was called the most secure in U.S. history, largely due to efforts to protect the nation's different physical and cyber infrastructure.¹ This was a triumph considering the physical, cyber, and even human assets that make up the election infrastructure have been and continue to be susceptible to threats.² In 2020, many states adopted measures to mitigate threats to physical and cyber election assets like voting equipment, ballots, and facilities, as well as computer services and databases that store voter information. For example, states with close results in the 2020 presidential race had paper records of each vote, which gave them the ability to go back and count each ballot if necessary.³ Other measures such as pre-election testing, state and federal certification of voting equipment, and increased collaboration between election

¹ Joint Statement from Elections Infrastructure Government Coordinating Council & the Election Infrastructure Sector Coordinating Executive Committees. Cybersecurity and Infrastructure Security Agency, (November 12 2020). <https://www.cisa.gov/news/2020/11/12/joint-statement-elections-infrastructure-government-coordinating-council-election>. Last accessed 8 July 2021.

² DHS Plans Are Urgently Needed to Address Identified Challenges Before the 2020 Elections. Government Accountability Office, (Feb 2020). <https://www.gao.gov/assets/gao-20-267.pdf>. Last accessed 8 July 2021.

³ Joint Statement from Elections Infrastructure Government Coordinating Council & the Election Infrastructure Sector Coordinating Executive Committees.

officials and their security partners helped provide additional assurance that the 2020 election results were legitimate.⁴

Unfortunately, analogous measures were not in place to protect many of the people—from state and local election administrators to poll workers and vendor staff—who administered the elections. Due in large part to the incendiary rhetoric from former President Donald Trump and his followers, many election officials were threatened and harassed, and such behavior shows no signs of abating.^{5,6,7} Instead of taking steps to address this problem, many states have passed or are considering passing laws that could make this problem worse.⁸ Georgia’s new law removed the Georgia’s secretary of state—who rejected Donald Trump’s effort’s to overturn the state’s 2020 results—from decision-making power on the state election board and turned control of the board over to a politicized, less knowledgeable body: the state legislature.^{9,10} Iowa’s new law threatens election officials with criminal prosecution for failing to follow new voting rules.¹¹ And Florida recently adopted a rule that subjects its local election officials to a civil penalty of \$25,000 if a drop box at an early voting site is left accessible for the return of ballots outside of early voting hours.¹²

⁴ Ropek, L.: The ‘Most Secure’ U.S. Election Was Not Without Problems. *Government Technology*, (November 16, 2020). <https://www.govtech.com/elections/the-most-secure-us-election-was-not-without-problems.html>. Last accessed 8 July 2021.

⁵ Wines, M.: Here Are the Threats Terrorizing Election Workers. *New York Times*, (December 3, 2020). <https://www.nytimes.com/2020/12/03/us/election-officials-threats-trump.html>. Last accessed 8 July 2021.

⁶ Douglas, J.: Republicans aren’t just making it harder to vote. They’re going after election officials, too. *Washington Post*, (May 9, 2021). <https://www.washingtonpost.com/opinions/2021/05/09/republicans-arent-just-making-it-harder-to-vote-theyre-going-after-election-officials-too/>. Last accessed 8 July 2021.

⁷ Resnik, B. [@brahmresnik]. NOW Arizona Gov. Doug Ducey’s office has assigned state troopers to provide round-the-clock protection to @SecretaryHobbs after report of death threats. Second time in 6 months that AZ’s top elections officer has received law-enforcement protection. [Tweet]. *Twitter*, (May 7, 2021). <https://twitter.com/brahmresnik/status/1390764118051745794?s=20>. Last accessed 8 July 2021.

⁸ Douglas, J.

⁹ Hasen, R.: Republicans Aren’t Done Messing with Elections. *New York Times*, (April 23, 2021). <https://www.nytimes.com/2021/04/23/opinion/republicans-voting-us-elections.html>. Last accessed 8 July 2021.

¹⁰ Gardner, A.: ‘I just want to find 11,780 votes’: In extraordinary hour-long call, Trump pressures Georgia secretary of state to recalculate the vote in his favor. *Washington Post*, (January 3, 2021). https://www.washingtonpost.com/politics/trump-raffensperger-call-georgia-vote/2021/01/03/d45acb92-4dc4-11eb-bda4-615aefd0555_story.html. Last accessed 8 July 2021.

¹¹ Norden, L.: Protecting American Democracy Is No Crime. *Foreign Affairs*, (April 7, 2021). <https://www.foreignaffairs.com/articles/united-states/2021-04-07/protecting-american-democracy-no-crime>. Last accessed 8 July 2021.

¹² Douglas, J.

The unrelenting targeting of administrators—along with unfunded election mandates, threats, burn out, and fatigue—has pushed many election officials to leave or consider leaving their positions.^{13,14,15,16} According to a recent Democracy Fund/Reed College Survey of Local Election Officials, approximately one-third of chief local election officials will be eligible to retire before the 2024 election, including more than half of those in the largest jurisdictions, each of which serve more than 250,000 registered voters.¹⁷ Considering what many of these election officials and their staffs endured during the 2020 election cycle, and continue to experience, it would be unwise to simply wait and hope that they decide to hang on. Instead, every effort should be made to create an environment that is as conducive as possible for retaining these unsung heroes of the 2020 elections. The ability of the United States to conduct future secure elections may well depend on it.

This paper focuses on measures state and local governments can take in concert with their election offices to help retain as many of their high-performing election officials as possible. State and local governments should have support from the federal government and Congress to help address this national security issue, but if the fight over funding the 2020 election is any indication, they should begin taking steps to address this problem now. Below are ten suggestions for how state and local governments can retain their high-performing election officials.¹⁸ None is a “silver bullet,” not all of them will be relevant for each state and local government unit, and some governments have already taken many of these steps. But in addressing election official retention solutions, these suggestions provide remedies for some of the thorniest problems that election officials encounter.

¹³ Myers, J.: Political Road Map: Here’s why California counties can ignore a half-dozen election laws. *Los Angeles Times*, (April 2, 2017). <https://www.latimes.com/politics/la-pol-ca-road-map-election-funding-california-20170402-story.html>. Last accessed 8 July 2021.

¹⁴ Wines, M.

¹⁵ Albiges, M., Lisi, T.: Pa. election officials are burnt out and leaving their jobs after 2020 ‘nightmare’. *WHYY*, (December 22, 2020). <https://whyy.org/articles/pa-election-officials-are-burnt-out-and-leaving-their-jobs-after-2020-nightmare/>. Last accessed 8 July 2021.

¹⁶ Democracy Fund and Early Voting Information Center: Local Election Official Interview Report. Fors Marsh Group, (December 15, 2020). https://evic.reed.edu/wp-content/uploads/2021/04/leo2020_idi_report.pdf. Last accessed 8 July 2021.

¹⁷ Gronke, P., Manson, P., Lee, J., Creek, H.: Amplifying the Perspectives of Officials at the Front Lines of Elections. Democracy Fund, (April 19, 2021). <https://democracyfund.org/idea/amplifying-the-perspectives-of-officials-at-the-front-lines-of-elections/>. Last accessed 8 July 2021.

¹⁸ Scheck, T., Hing, G., Robinson, S., Stockton, G.: How Private Money From Facebook’s CEO Saved The 2020 Election. *All Things Considered*. National Public Radio, (December 8, 2020). <https://www.npr.org/2020/12/08/943242106/how-private-money-from-facebooks-ceo-saved-the-2020-election>. Last accessed 8 July 2021.

2 Recommendations

2.1 Ask your election officials what can be done to support them further.

During 2020, election officials had to administer elections while there was an ongoing threat from foreign adversaries coupled with a pandemic, civil unrest, and widespread, unjustified suspicion about their work. Just as election officials sought to understand what their voters needed, governments should seek to understand what their election officials need most to succeed. Asking officials questions is a good first step.

As part of this process, it could also be useful to ask election officials what skills they're most comfortable with and which they would like to develop.¹⁹ Inquire about areas that feel especially challenging. This can be done by discussing the parts of the jobs they feel are the most interesting and rewarding, the areas that are the most challenging, what they are doing to reach short- and long-term career goals, and what other projects, committees, or additional issues they would like to explore. Asking such questions not only shows empathy and understanding, but could lead to information that creates more effective training and development programs, which are critical to increasing employee retention.²⁰

2.2 If there was turnover among your election officials, ask why.

Examine turnover rates from the 2020 election cycle to the present for both election management and front-line staff and see how they compare to previous similar election cycles. Often, the reasons management and front-line staff provide for leaving a job, including elections, can be very different.²¹ This information should help inform the degree to which the unique circumstances surrounding the 2020 elections have affected the retention of your own election officials. These rates, combined with feedback from election officials, should give governments a better sense of how retention could be affected by future elections that are as contentious as 2020.

2.3 Consider the cost of election official employee turnover, or the “total cost” of losing an employee.

If governments and their election officials can help show the costs that employee turnover from accumulated strain caused by contentious elections—like the 2020 presidential election—is having on their workforces, that could make it easier to seek additional funding for retaining high performing election administrators and addressing other

¹⁹ Rogers, M.: A Better Way to Develop and Retain Top Talent. Harvard Business Review, (January 20, 2020). <https://hbr.org/2020/01/a-better-way-to-develop-and-retain-top-talent>. Last accessed 8 July 2021.

²⁰ Ibid.

²¹ Katzenbach, J., Santamaria, J.: Firing Up the Front Line. Harvard Business Review, (May–June 1999). <https://hbr.org/1999/05/firing-up-the-front-line>. Last accessed 8 July 2021.

priorities.²² Since the exact costs of employee turnover vary, it is something all employers, including election offices, need to monitor. One study found that the average costs to replace those earning under \$30,000 a year was 16 percent of their annual salary; the average costs to replace those earning \$30,000–\$50,000 a year was 20 percent of their annual salary; and the cost to replace executive positions could cost up to 213 percent.²³ If officials do not already have systems in place to track such costs, they should reach out to colleagues in relevant departments, such as human resources, finance, and operations to develop tools to measure these costs and reporting mechanisms to track them.

The costs of election turnover should include: 1) computing the cost of hiring a new employee including the advertising, interviewing, screening and hiring;²⁴ 2) determining the cost of onboarding a new person, including training and management time; 3) accounting for lost productivity – due to the different kinds of elections that occur in each state over a four year period and the wide array of skills needed to perform the job well, it can often take a new election official up to four years to reach the productivity of an existing person;²⁵ 4) factoring in lost engagement—other employees who see high turnover tend to disengage and lose productivity; 5) adjusting for customer service and errors—new employees generally take longer and are often less adept at solving problems; and 6) accounting for training—during a new election official’s first few years, the office is often likely to invest more of the employee’s salary in training.

2.4 Compare the compensation of your election administrators to other government employees with similar responsibilities and adjust appropriately.

According to a recent survey, the top reason workers quit their jobs for new ones is so they can make more money.²⁶ Currently, the typical local election official makes about \$50,000 annually, which appears to be more on par with administrative support positions than positions with commensurate skills.²⁷ In short, election administrators are underpaid.

²² Bersin, J.: Employee Retention Now a Big Issue: Why the Tide has Turned. LinkedIn, (August 16, 2013). <https://www.linkedin.com/pulse/20130816200159-131079-employee-retention-now-a-big-issue-why-the-tide-has-turned/>. Last accessed 8 July 2021.

²³ Boushey, H., Glynn, S.: There Are Significant Business Costs to Replacing Employees. Center for American Progress, (November 16, 2012). <https://www.americanprogress.org/wp-content/uploads/2012/11/CostofTurnover.pdf>. Last accessed 8 July 2021.

²⁴ Bersin, J.

²⁵ Gronke, P., Manson, P., Lee, J., Creek, H.

²⁶ Why They ‘Quit You.’ Payscale, (2019). <https://www.payscale.com/data/why-people-quit-their-jobs>. Last accessed 8 July 2021.

²⁷ Adona, N., Gronke, P., Manson, P., Cole, S.: Stewards of Democracy: The Views of American Local Election Officials. Democracy Fund, (2018). https://democracyfund.org/wp-content/uploads/2020/06/2019_DemocracyFund_StewardsOfDemocracy.pdf. Last accessed 8 July 2021.

Election officials' jobs were once clerical in nature and more akin to record-keeping, but those days are long gone.^{28,29} Today, they're often expected to be in experts in numerous disciplines, including cybersecurity, communications, logistics, finance, election law, public administration, public health, and human resources.³⁰ State election administrators often rely on information from many of these disciplines to make decisions about the rules of elections.³¹ Local elections officials must flawlessly fulfill many of these roles to successfully administer an election.³² This includes finding polling places, recruiting workers, and running the day-to-day operations of voter registration and voting, as well as preserving the integrity of elections by protecting against intrusions into voter rolls and local election official websites, and working closely with federal and state officials to ensure the security of their voting systems.³³ If key election administrators are not paid comparably to other government employees with similar responsibilities, retaining them is likely to be a greater challenge.

2.5 Establish a Government Resource Group.

One strategy used for helping keep talent in many industries, including elections, is to establish a resource group.³⁴ An employee resource group is a network within an employer where employees get together based on shared characteristics, experiences, or goals. Such a group offers a chance to network and socialize, work on professional development, and raise awareness of relevant issues. While many election officials have groups with other election officials elsewhere, the governments that they work for should strongly consider forming such groups if they have not already.

²⁸ Report of the Joint Legislative Audit and Review Commission on Compensation Of General Registrars to the Governor and the General Assembly of Virginia. Senate Document 5. Commonwealth of Virginia, (1992). <http://jlarc.virginia.gov/pdfs/reports/Rpt130.pdf>. Last accessed 8 July 2021.

²⁹ Levine, D.: The Election Official's Handbook: Six steps local officials can take to safeguard America's election system. Alliance for Securing Democracy. German Marshall Fund, (February 13, 2020). <https://securingdemocracy.gmfus.org/wp-content/uploads/2020/02/The-Election-Officials-Handbook.pdf>. Last accessed 8 July 2021.

³⁰ Howard, E.: 5 Things You May Not Know About Local Election Officials. Brennan Center for Justice, (October 26, 2020). <https://www.brennancenter.org/our-work/research-reports/5-things-you-may-not-know-about-local-election-officials>. Last accessed 8 July 2021.

³¹ The State and Local Role in Election Administration: Duties and Structures. Congressional Research Service, (March 4, 2019). <https://fas.org/sgp/crs/misc/R45549.pdf>. Last accessed 8 July 2021.

³² Election Administration at State and Local Levels. National Conference of State Legislatures, (February 3, 2020). <https://www.ncsl.org/research/elections-and-campaigns/election-administration-at-state-and-local-levels.aspx>. Last accessed 8 July 2021.

³³ Chase, M.: Letter on FY 2020 Elections Appropriations. National Association of Counties, (December 10, 2019). <https://naco.sharefile.com/share/view/s8aabb5333a74ce8b>. Last accessed 8 July 2021.

³⁴ 3 Simple Ways to Recruit and Retain Top Military Talent. Yello. <https://yello.co/blog/3-simple-ways-to-recruit-and-retain-top-military-talent/>. Last accessed 8 July 2021.

Many of the challenges election officials faced in the 2020 election cycle are an unfortunate outgrowth of decreasing trust in government and increasing amounts of mis- and disinformation, challenges that employees throughout government are confronting, not just in elections.³⁵ Election officials often depend on large numbers of fellow government employees, particularly around election time, to serve as poll workers, process voter registration applications, and help conduct their elections. A government resource group, particularly one geared towards how to address abusive members of the public, could go a long way towards strengthening this collaboration by enabling election and non-election employees to get to know one another better. Such groups can help employees form friendships across departments, which can supercharge an employee's engagement. They also enable employees to act together to address common issues and spread awareness, and contribute to higher retention rates.³⁶

2.6 Offer election administrators flexible working conditions when possible.

Flexible working conditions are not always possible in elections, particularly as election day gets closer and the demands on election officials from candidates, voters, observers, and others increase.³⁷ However, the pandemic provided an opportunity for many election officials to learn how to conduct more of their work remotely and with flexible work hours, and thousands of election officials responded by taking security training that made it easier for them to work remotely yet securely during the first Covid-19 peak in the spring and summer of 2020.^{38,39} For example, many election offices can input and validate voter registration and mail ballot application requests remotely after the hardcopy originals are scanned and securely shared with necessary employees. And nearly all elections officials can do a good deal of their phone-based work, such as the recruitment and confirmation of poll workers and polling places, outside the workplace as well.⁴⁰

³⁵ Aguilera, J.: 'An Epidemic of Misinformation.' New Report Finds Trust in Social Institutions Diminished Further in 2020. *Time Magazine*, (January 13, 2021). <https://time.com/5929252/edelman-trust-barometer-2021/>. Last accessed 8 July 2021.

³⁶ Asare, J.: How to Retain Diverse Talent. *Forbes*, (September 26, 2018). <https://www.forbes.com/sites/janicegassam/2018/09/26/how-to-retain-diverse-talent/?sh=261deb322d33>. Last accessed 8 July 2021.

³⁷ Election Administration at State and Local Levels. National Conference of State Legislatures.

³⁸ Freed, B.: Annual election security tabletop drill put officials through 'Armageddon-like' test. *StateScoop*, (July 31, 2020). <https://statescoop.com/dhs-election-tabletop-exercise-2020/>. Last accessed 8 July 2021.

³⁹ Norden, L., Ramachandran, G.: Election Officials Spent Four Years Beefing Up Voting Security. *It Paid Off. Slate*, (November 12, 2020). <https://slate.com/technology/2020/11/election-security-2020-pandemic.html>. Last accessed 8 July 2021.

⁴⁰ Safeguarding Staff and Work Environment from Covid-19. Election Infrastructure Government Coordinating Council, (May 28, 2020). https://www.eac.gov/sites/default/files/election-officials/inpersonvoting/Safeguarding_Staff_and_Work_Environment.pdf. Last accessed 8 July 2021.

While many election processes—such as voting equipment preparation, mail ballot signature validation, and provisional ballot adjudication—will continue to occur at the workplace, the successes achieved from working remotely should be built upon because they can be a win-win situation for both election officials and their governments.⁴¹ Election administrators can save time and money to commute, have improved work-life balance and fewer distractions and, be more productive.⁴² Governments may be able to save on some infrastructure costs and overhead costs, and reduce absenteeism.⁴³ Organizations that provide the option for remote work have 25 percent lower employee turnover; state and local governments would be wise to heed this warning if they hope to retain their best election officials.⁴⁴

2.7 Advocate for consolidating elections to no more than three per year.

Election officials know firsthand the cost, administrative burden, and job stress caused by the near constancy of elections in some states and localities, which can increase the likelihood of significant mistakes being made and make retaining election officials harder. Others in state and local governments may not feel the administrative burden and job stress as acutely, but they are certainly aware of the cost. Consolidating elections to no more than three per year would enable election officials to keep their skills sharp while also lessening costs and reducing election officials' stress. It could also help increase turnout in both local and national elections.⁴⁵

Consolidated elections admittedly are not perfect, and they require a good deal of change. For example, consolidated elections often lead to longer ballots, which can increase wait times at polling places.⁴⁶ However, other measures such as expanded pre-election day voting and a more efficient allocation of staff and voting machines at

⁴¹ Ibid.

⁴² Choudhury, P.: Our Work-from-Anywhere Future. Harvard Business Review, (November–December 2020). <https://hbr.org/2020/11/our-work-from-anywhere-future>. Last accessed 8 July 2021.

⁴³ Edwards, K., Zaber, M., Girven, R.: Should the Federal Workforce Stay Remote? Planning for After the Crisis. The RAND Blog, (April 3, 2020). <https://www.rand.org/blog/2020/04/should-the-federal-workforce-stay-remote-planning-for.html>. Last accessed 8 July 2021.

⁴⁴ Top 26 Employee Retention Strategies for the “New” Work World! The Vantage Circle, (July 7, 2021). <https://blog.vantagecircle.com/employee-retention-strategies/>. Last accessed 8 July 2021.

⁴⁵ Phillips, C.: The Effect of Election Consolidation on Turnout: Evidence from California. Presentation to the Election Sciences, Reform, and Administration Conference in Philadelphia, Pennsylvania, (July 2, 2019). <https://cpb-us-w2.wpmucdn.com/web.sas.upenn.edu/dist/7/538/files/2019/07/Connor-Phillips-ESRA-Paper.pdf>. Last accessed 8 July 2021.

⁴⁶ The American Voting Experience: Report and Recommendations of the Presidential Commission on Election Administration. Presidential Commission on Election Administration, (January 2014). <http://web.mit.edu/supportthevoter/www/files/2014/01/Amer-Voting-Exper-final-draft-01-09-14-508.pdf>. Last accessed 8 July 2021.

polling places can help offset such an increase, and most local election officials support consolidating local, state, and federal elections so that they occur at the same time.⁴⁷

2.8 Make sure election officials are aware of all current security protections in place.

This includes any arrangements with law enforcement, office and worktime security measures, and tools available to help ensure their personal physical security. A meeting between election officials and local law enforcement can help confirm existing protections and identify any potential gaps. Such efforts can also be aided by your local Cybersecurity & Infrastructure Security Agency (CISA) protective security advisor (PSA).⁴⁸ PSAs are trained critical infrastructure protection and vulnerability mitigation subject matter experts whose expertise protecting cyber, physical, and human components of critical infrastructure, like elections, makes them both an advisor and a natural go-between for election officials and law enforcement.

2.9 Develop a plan to provide more enhanced protection for election officials going forward.

During the 2020 election cycle, state and local election administrators across the country received violent threats at work, at home, and elsewhere.^{49,50,51} Some of these officials feared simply administering an election in which a defeated candidate's most ardent followers could refuse to accept the results. Retaining many of them will be difficult unless more can be done to ensure they feel safe. One place to look for developing a more robust security plan for election officials is federal judges, who are often subject to threats well beyond the courtroom due to the availability of personal information online.⁵² In the same way that election administrators can represent or personify the election system, judges and other judicial officials can represent or personify the justice

⁴⁷ Gronke, P., Manson, P., Lee, J., Creek, H.

⁴⁸ Running Elections Without Fear: Ensuring Physical Safety for Election Personnel. The Elections Group, (2020). <https://electionsgroup.com/assets/Running%20Elections%20Without%20Fear.pdf>. Last accessed 8 July 2021.

⁴⁹ Marks, J.: The Cybersecurity 202: Violent threats only make elections more vulnerable, experts fear. Washington Post, (December 3, 2020). <https://www.washingtonpost.com/politics/2020/12/03/cybersecurity-202-violent-threats-only-make-elections-more-vulnerable/>.

⁵⁰ Huseman, J.: For Election Administrators, Death Threats Have Become Part of the Job. ProPublica, (August 21, 2020). <https://www.propublica.org/article/for-election-administrators-death-threats-have-become-part-of-the-job>. Last accessed 8 July 2021.

⁵¹ Running Elections Without Fear: Ensuring Physical Safety for Election Personnel. The Elections Group.

⁵² Judicial Security: Safeguarding Courts and Protecting Judges. *Judicature* 104(3), (Fall–Winter 2020–21). <https://judicature.duke.edu/articles/judicial-security-safeguarding-courts-and-protecting-judges/>. Last accessed 8 July 2021.

system; in both cases, the motive for an attack can arise out of anger at the respective systems or simply a desire for revenge.⁵³

Like federal judges, all election officials should have access to safety education programs that offer trainings on a wide range of threats and how to reduce exposure to them.⁵⁴ And like federal judges, state and local governments could also consider more expensive actions as their budgets allow, such as installing and/or updating security systems in their election officials' offices; installing security systems at election officials' homes on an as needed basis; hiring additional security in response to a potential threat; and/or monitoring the public availability of election officials' personally identifiable information and referring suspicious posts to the appropriate law enforcement authorities.⁵⁵ Governments should also consult with their legal counsel to see if they can devise a legal strategy for helping deter untoward behavior by election officials. This could include conducting a public messaging campaign that describes how best to interact with election administrators for optimal results, while also noting the consequences for abusive behavior and harassment. It could also include designating a legal liaison to the elections office to help ensure that any potential illicit conduct towards elections officials is quickly reported and addressed.

2.10 Form a task force made up of government employees to advocate for legislative measures that support your government's work, including your election officials.

Over the past several months, many states have either adopted or introduced legislation that penalizes election administrators and workers.⁵⁶ Georgia's new law gives the legislature the power to pick an official who could vote on the state election board for a temporary takeover of up to four county election boards during the time when an election is being administered.⁵⁷ Iowa makes it a crime if election workers violate its new rules.⁵⁸ And Florida has enacted a law that says, "If any drop box at an early voting site is left accessible for the return of ballots outside of early voting hours, the supervisor is subject to a civil penalty of \$25,000."⁵⁹

Election administration often requires making decisions in the heat of closely contested elections. As a former election administrator myself, I regularly made many such decisions to both protect an individual's right to vote and ensure the integrity of the

⁵³ Reynolds, M.: An attack on a judge's family is putting judicial security center stage. ABA Journal, (October 1, 2020). <https://www.abajournal.com/web/article/attack-on-judges-family-puts-judicial-security-center-stage>. Last accessed 8 July 2021.

⁵⁴ Judicial Security: Safeguarding Courts and Protecting Judges. Judicature.

⁵⁵ Reynolds, M.

⁵⁶ Norden, L.

⁵⁷ Hasen, R.

⁵⁸ Douglas, J.

⁵⁹ Senate Bill 90. Florida State Senate, (2021). <https://www.flsenate.gov/Session/Bill/2021/90/BillText/e1/PDF>. Last accessed 8 July 2021.

election, knowing there were robust laws in place to hold me accountable if I acted inappropriately. Exposing election officials to additional, unnecessary civil or criminal liability for such actions could make them afraid to act in such situations and less likely to want to stay in the field.

3 Conclusion

Election administrators have taken oaths to support the Constitution, which forms the basis of the rule of law. Elections must be administered in a genuine and democratic manner. Election administrators that are concerned for their safety or burnt out from the stress of the job are more likely to perform poorly or even leave the profession. Every citizen, regardless of political affiliation, should want to make sure these defenders of democracy are comfortable performing their legal obligations and administering safe, secure, and transparent elections. Governments that adopt more of this reports' suggestions could go a long way towards making this goal a reality, which in turn could make it easier to retain our best defenders of democracy.

Vote Secrecy and Voter Feedback in Remote Voting – Can We Have Both?

Arne Koitmäe¹, Jan Willemson²[0000–0002–6290–2099], and Priit Vinkel²[0000–0003–0049–1287]

¹ State Electoral Office, Lossi plats 1a, Tallinn, Estonia

² Cybernetica, Narva mnt 20, Tartu, Estonia

Abstract. The principle of secrecy is one of the most important tools to guarantee a voting process without undue influence to the voter. However, the concepts of the secret ballot and secret vote have strong ties to voting in a controlled environment in the polling station, and remote voting methods like postal voting or Internet voting need to employ special measures and approaches to achieve similar results. At the same time, limited options of observing the tallying process remotely potentially undermines the trust in remote voting. This paper looks at possible ways of giving the voter some feedback and assurance in the integrity of their vote, at the same time adhering to the freedom of voting principle. The Estonian Internet voting system is used as a model case for evaluation of a possible feedback channel architecture.

Keywords: Voting feedback, freedom of voting, secrecy of vote, Internet voting, remote voting

1 Introduction

Freedom of voting – the principle where the voter is able to cast his or her vote without undue influence – is one of the cornerstones of the democratic process. Secrecy of the vote is one of the most important tools to achieve this goal. However, the way we understand vote secrecy is closely related to the concept of traditional voting – the ballot is filled in privately in the voting booth, and then deposited into the ballot box. However, many voting methods also deviate from this scheme. One example is postal voting, where there is no control whether the ballot is filled in privately, and no solid guarantees can be given that the ballot sent through mail is not lost, opened, or tampered with.

In general, once the paper vote is cast in the ballot box (or the envelope with a ballot posted in mail) the voter has no way of verifying how their vote is processed and counted. Observation of voting and vote counting procedures are meant to ensure the integrity of the tally. While the voter's participation is recorded in the the voter list and the data of the voter lists can be compared to the final tally, the path of the vote itself – anonymous ballot – is untraceable by the voter. This is usually not a problem if the trust towards the election management is high enough. However, it can be a problem if the trust is low,

especially if there are doubts about the elections being conducted in a free and fair way.

Internet voting (i-voting) provides new challenges when implementing ballot secrecy. A well-implemented i-voting system can use cryptography to guarantee that the ballot is sent and received as intended, with its integrity untouched. An observer or an auditor can make sure that all the votes cast are accounted for, that the votes included in the tally are the same as cast, and that the votes were tabulated correctly. However, voters themselves cannot fully verify i-voting results and people need to have absolute faith in the accuracy, honesty and security of the whole electoral system [38]. The path of their vote is something voters cannot trace or observe directly, and this can undermine the trust in the i-voting system. Trustworthiness of i-voting is more and more connected to additional confirmations given to the voter about the vote being handled correctly and processed as required by law.

However, the more information we give to the voter about their vote, the more the secrecy of the vote is undermined. In order to ensure freedom of the vote, it should not be possible to use this information against the voter. Secrecy of the vote should remain intact and voters should not find themselves in a weaker position against possible malefactors because their voting information is revealed.

Another problem in regards to i-voting and vote secrecy is the voting environment, which should ensure voter privacy. This cannot be guaranteed by election administration when the voter is voting from the location of their choice using a personal computer. Hence there are inherent risks present, like a possibility of malware tampering with the vote, or taking over the electronic identity used to authenticate the voter and sign the encrypted ballot. The worst-case scenario is that a malicious actor casts the vote using voter's electronic identity without the voter even knowing it. The observers and auditors cannot review how the vote was cast at the location of the voter. This presents a need for additional checks available to the voter. Merely the confirmation that the i-voting tally is verifiably correct doesn't address this concern. This concern is not limited to i-voting either.

Therefore it would be beneficial to give voters further confirmation about how their vote is handled with a goal to increase the trust in voting in general. Another issue to consider is that such measures should not make voting arrangements too complex for the voter, as not to restrict access to voting. In this paper we will examine whether this can be achieved without significantly weakening vote secrecy.

In order to have a more concrete treatment of the topic, we will be using Estonian Internet voting as the example case study throughout this paper. In the Parliamentary and European Parliament elections of 2019, the share of i-votes cast was 43.8% and 46.7% of participating voters, respectively [14]. Thus legitimacy of elections in Estonia very much hinges on the perceived trust of i-voting. Estonian i-voting system features both individual verification (introduced in 2013 [32]) and server-side auditing (introduced in 2017 [29]). Swiss and

Norwegian i-voting solutions have implemented individual and universal verification solutions as well. The Swiss Post e-voting solution uses verification of votes cast both individually by voters and universally by the electoral commission [17]. The Norwegian i-voting system used return codes for individual verification and server side auditing [26]. Individual verification is limited to confirming that the voter’s vote was received as intended by the vote collecting service. Server-side auditing, on the other hand, allows to certify that the votes as a complete set have been tallied correctly. However, the popularity of i-voting in Estonia has initiated debate over 1) how freedom of vote and vote secrecy are guaranteed for Internet voting, and 2) what measures would increase general trust in the system [13]. Contributing to this discussion is the main motivation behind the current paper.

The paper is organised as follows. Section 2 presents a discussion on the concept of secret ballot that has been traditionally used to guarantee voting freedom. We also take a broader look at remote voting environments to understand how far is it reasonable to go with the vote secrecy requirement in this setting. Section 3 studies a possible additional feedback channel notifying the voter on the fact that a vote has been cast on their behalf. We analyse possible implementations of such a channel together with their impact on voting freedom. Finally, Section 4 presents some conclusions and sets directions for future work.

2 Concept of the secret ballot

2.1 Secrecy of the vote

Vote secrecy hasn’t always been a requirement when conducting elections. Before mid-19th century it was rather a standard to vote openly, e.g. via stating one’s preference out loud, or using visually distinguishable ballot sheets. Of course this also encouraged various coercive practices. To counter these, voting by secret ballot was introduced, with Australia being one of the first countries where it was systematically implemented [22, 41].

Today the requirement of vote secrecy has been stated in the highest level of international legislative acts. The United Nations International Covenant on Civil and Political Rights (UN CCPR) [3, Art 25], United Nations Universal Declaration of Human Rights [1, Art 21] as well as the European Convention of Human Rights (ECHR) [2, Art 3 of Prot I] state that voting shall be held by secret ballot. UN CCPR’s General Comment 25 [4] adds that states should take measures to guarantee the requirement of the secrecy of the vote during elections, implying that voters should be protected from any form of coercion or compulsion to disclose how they intend to vote or how they voted, and from any unlawful or arbitrary interference with the voting process.

On electronic voting, Article 3.2 (iv) of the Council of Europe (CoE) Venice Commission’s Code of Good Practice In Electoral Matters states that that the (electronic) voters should be able to obtain a confirmation of their votes and to correct them, if necessary, respecting secret suffrage [6]. The CoE recommendation CM/Rec(2017)5 [12] on standards for e-voting makes several suggestions

towards maintaining vote secrecy. Article 23 of the Appendix to CM/Rec(2017)5 states that an e-voting system shall not provide the voter with proof of the content of the vote cast for use by third parties. Article 24 states that e-voting shall ensure that the secrecy of previous choices recorded and erased by the voter before issuing his or her final vote is respected.

The Code of Good Practice in Electoral Matters elaborates on the concept of secret suffrage on the voter's side as well. It states that for the voter, secrecy of voting is not only a right, but a duty as well. It also requires that voting must be individual, and that the list of persons actually voting should not be published [6, Art 4]. In the explanatory report, the Venice Commission explains that the purpose of the secrecy of the ballot is to shield voters from pressures they might face if others learned how they had voted [6, Par 52]. Moreover, since abstention may indicate a political choice, list of persons voting should not be published [6, Par 54].

From the voter's point of view, perceived vote secrecy is not necessarily equal to formal vote secrecy interpreted and implemented by the Electoral Management Body (EMB). The voters must also believe that the election administration operates in a way that their choices are kept secret (psychologically secret ballot) [27]. I-voting adds another dimension here, since the voters must additionally believe that other voters respect privacy and secrecy of the vote. Additionally, voters might feel socially obligated to reveal their votes, or they can believe that other voters might do so (social secrecy of the ballot) [27].

In the jurisprudence of the model case of Estonia, the current thinking regarding secrecy and Internet voting is based on the teleological approach, meaning that constitutional principles should be understood through the problems these principles were meant to solve [24]. It was first noted in 2004 as the underlying motivation for the draft legislation allowing for Internet voting [24]. In addition to that, the second source of the current approach is the liberal idea of trusting the voter [24, 36]. The principle of secrecy would protect an individual from any pressure or influence against her or his free expression of a political preference. Thus, the principle of secrecy is a means, not an end goal [24, 37]. Influence resistance in the Estonian i-voting system is guaranteed by the possibility of re-voting, thus the principle of secrecy, the end goal, is actually achieved [36]. This approach has now been generally accepted and expanded on [35, 37, 38] as not just the reasoning behind the original draft legislation, but as the actual explanation to how Internet voting conforms to the principle of secret ballot.

There remains a question whether the second part of reasoning – that the voter should be trusted – is applicable to the principle of secrecy. Vote secrecy cannot be understood as just optional, i.e. it's not just up to the voter to decide [19], but remote internet voting requires rethinking of the privacy principle [36, 37]. In support of a more traditional approach, Buchstein in 2004 (before the first i-enabled Estonian elections in 2005) argued for the sanctity of the secret ballot, while admitting that Drechsler's and Madise's interpretation and Estonian constitutional debate comes in as a possible starting point for a paradigmatic change [23]. There were also concerns that the transition towards voting

more from home, the concept of election may change without a real discussion on how that may weaken the voters' consciousness of a secret and personal vote [40]. This paradigmatic change has occurred, to an extent, when considering i-voting initiatives in Estonia, Switzerland and Norway, but also the raise in popularity of postal voting in general. The aforementioned countries have developed their i-voting system in line with the international standards and recommendations, while monitoring the experiences of other countries [21]. The updated CoE recommendation on i-voting CM/Rec(2017), now at its second iteration, reflects this change as well.

In practice vote secrecy on voter's side has been difficult to enforce, as many voters do not care about secrecy or do want to make their choice known, because of the social secrecy of the ballot as described by Gerber *et al.* [27].

2.2 Secrecy of participation in voting

Additional consideration should be given to how the principle of secrecy relates to voter's participation in voting. The Venice Commission has explained that voter lists with information on who voted shouldn't be published and abstention is a form of political choice [6, Par 54].

At the same time, when we look at voting as a general process, full participation secrecy is impossible to implement as voting in the polling station is public by nature. In regards to social secrecy of the vote, voters are often encouraged to participate and make their participation known by election stakeholders. This can possibly lead to problems in maintaining vote secrecy as well. For example, in Sweden, where ballots are printed separately for each party, party activists hand out ballots in front of the polling place to their voters. If the voter takes just one ballot, the content of the ballot is then known to bystanders [25].

The act of voting and content of the ballot are not approached the same way by voters and election stakeholders. As a result, voter lists (at least individual data of a voter) do not really fall under the umbrella of maintaining vote secrecy. In the past, personalised data on Internet voters has even been studied by researchers [35].

As for our model case of the Estonian Internet voting system, the current regulations stipulate that all data on Internet voters shared for scientific purposes must be made anonymous (including voting logs) [8, Par 77-1 (2)]. As for polling station voter lists that have been traditionally on paper, access to them is limited to the voters (personal information only) and parties; candidates and their representatives must justify why they need access (e.g in case of an elections dispute) [8, Par 23 (2)]. Additionally, the data can be used for scientific purposes. Thus the data concerning the voter is always available to the person without limitations, but the voter list data cannot be published or released to third parties except in cases stipulated by the law.

2.3 Challenges of keeping vote secrecy while increasing voter trust in the modern voting environment

A modern voting environment can include several methods of voting that differ in how much direct control the EMB has over it. Voting in a polling station takes place in a standardized environment, under control of the polling station staff. At the same time, the ballot box voting arrangements at home, overseas or at hospitals can be less convenient for the voter. On the other side of the spectrum are off-site voting methods like postal voting and Internet voting, being conducted without any supervision of the election administration. The vote delivery channel (mail or Internet) is in such cases not controlled by the EMB either.

If we accept that:

1. maintaining vote secrecy is not just the task of the EMB, but also of the voter,
2. not all ballots are cast under the direct supervision of election administration,
3. vote secrecy is just means to achieve the principle goal of free elections,

voters should also have the appropriate tools to be able to achieve that goal.

There are already a few measures at the disposal of the voter (with the implementation details varying across jurisdictions), e.g.:

- The voter can vote on the election day at a polling station and then observe the election procedures up to the end of vote counting. This gives a certain level of confidence that the voter’s personal ballot (among other ballots) was not tampered with. Here the voter has to trust their own observation.
- Voters can check their data in the voter list, which includes information on whether they have voted, and possibly also the voting method that was used (i.e. Internet voting, voting outside the territory of their municipality or constituency). However, if the voter must personally access the voter list (or request the information from the EMB) then this requires action on voter’s part and the voters must also be aware of the possibility. Therefore it is unlikely to provide any statistically significant amount of verifiability to increase trust in elections in general.
- An Internet voter could verify that the vote cast was received and stored as intended. There are several ways to implement this. For example, in Estonia, a smart device application is used for verification [32], but it does not help in the case when the voter is unaware that someone has cast a vote on their behalf. Since this method requires action on the voter’s side, it hasn’t achieved wide usage. The share of i-votes verified by the voters has remained between 4-5 per cent of all i-votes since 2014 [14]. It can be used to detect certain mass attacks against i-voting (e.g. when malware is trying to manipulate active voting sessions), but not all of them (e.g. when malware itself initiates the sessions without voter participation).

- In case of postal voting in Finland, the postal voter and the voting procedure have to be accompanied by two independent witnesses who could attest in writing that the freedom of vote and vote secrecy have been adhered to in this process [33, 39].

None of these measures undermine vote secrecy, but the problem is that these methods are limited in scope and they presume significant extra actions from the voters.

In order to certify one’s vote, there are also other methods that are either discouraged by EMBs or not supported by legislation.

- Voters can take a photo of their ballots in the polling booth, or screen capture their choices in the Internet voting app or verification app. The voter can also live broadcast their voting from the polling booth [20]. This provides some (although quite a weak form of) proof that the vote has been cast correctly. This also lets the voter publish the image of the ballot taking, thus conflicting the vote secrecy principle.
- Voters can also mark their paper ballots in a way that it would be recognizable during the vote counting. If the voter (or some other informed party) then observes the count, they can make notice whether and how their vote was counted [42]. This is also possible for Internet voting, for example modifying the choice on the ballot in a way that the i-vote will be counted as invalid. As an example, there have been actual cases of sending in invalid votes in case of Estonian i-voting [30, 31].

The two above channels are violating the vote secrecy requirement, presenting proof of the contents of the ballot, thus making the voter more vulnerable to undue coercion. However, neither of the methods is something the EMB can directly block. In such cases it should be up to the legislation and EMB to determine if the act of vote is impermissible or the vote invalid.

In Finland, for example, the votes that contain extra markings on them are declared invalid by law [5, Par 85 (6)]. However, in Estonia, such a regulation does not exist. In fact the law stipulates that if the ballot is not filled correctly (e.g. the number of the candidate is not written on the correct spot), but the choice of the voter is otherwise understood (e.g. the name of the candidate was written on the ballot), the ballot is considered valid [8, Par 57 (6) 8)]. This presents an opportunity for the voters to get creative, enabling tracking of their votes. As for taking pictures of ballots (and publishing them), restricting these activities is even more complicated.

In the case of *stemfies* (ballot selfies), it is also apparent that the legislation and our general understanding of the secrecy have not kept up with the technological advancements [20]. It is unclear, whether and how such voter-initiated deviation from secrecy should be blocked and enforced by law, especially for remote voting. The consensus in this hasn’t been reached yet. For example Section 56 (6) 5a of German Federal Electoral Regulations states that the Electoral Board must turn away any voter whom they find taking photos or videos in the

voting booth [7]. At the same time, in The Netherlands taking ballot selfies is allowed, although not encouraged [11]. *Stemfies* can also spark debate about other human rights and freedoms. European Court of Human Rights has ruled [15] that forbidding to use a mobile app to publish voter’s ballot was in conflict with the Art 10 (Freedom of Expression) of the European Convention on Human Rights [2].

In summary, to improve voter’s control over how voting is handled, we should be looking for a solution that wouldn’t interfere with vote secrecy, give voters a way to verify their vote was handled correctly, and that would be universal enough to achieve statistically significant amount of checks.

A possible way to achieve the latter goal is to require as little action from the voter as possible. As we saw above, one of the main attack vectors not detected by the current verification mechanisms is malware that casts votes without the voter knowing about it. A similar problem occurs if the voter’s eID is taken over physically. To detect such attacks, the system can be augmented with a feedback channel that gets triggered every time a vote is cast on voter’s behalf. Next we will be studying the options of establishing such a channel.

3 Establishing a feedback channel

3.1 Feedback on the fact of casting a vote

When introducing a feedback channel, our goal is to give i-voters additional assurance that they have (or have not!) voted. On the other hand, we do not want to publish the proof in a way that it would render re-voting as a measure to maintain voting freedom inefficient.

Currently, the Estonian system allows to get feedback on several levels.

- Confirmation that the vote collecting service has received the i-vote and received it as intended. In Estonia this is currently implemented by the smart device verification app.
- Confirmation that the i-vote was included in the set of i-votes that are going to be tallied. Since the list of i-voters is created by the Internet voting system, a voter can check if their i-vote is included in this list, but this action is very inconvenient to the voters (see Section 2.3).
- Confirmation that the i-vote was amongst the i-votes tallied. Currently no feedback for the voter exists here, but the integrity of the i-vote set is verified by the EMB and auditors.
- Confirmation that the vote was counted as intended. Currently no feedback for the voter exists here, but the result can be verified by the tallying proof by the EMB, auditors and by anyone who has created an auditing application.

What is missing from this list is a passive method for getting information about the vote being received by the system. If such a feedback on voting participation only reveals the fact that the voter has voted at some point, then the clash with the principle of vote secrecy is minimal. It would, however, imply that

the voter has not abstained from voting. In such a way, giving a notification that a person has i-voted would be similar to situation when someone would take a photo of a voter leaving a polling station.

Introducing a voting fact feedback channel would benefit the voter in two main ways:

1. the voter would get assurance that the vote has been received and stored; and
2. even if the voter did not vote, absence of the voting notification would confirm that no-one else has not voted for them.

Both confirmations would be useful to both i-voters and paper ballot voters. The assurance for the voter that no-one has cast a vote on their behalf can hopefully increase trust in the elections, including Internet voting.

Recall, however, that the ability to withstand coercion attacks relies on the possibility to cast re-votes in the Estonian system. Thus, assurance about which vote was processed (tallied) would potentially weaken the position of the i-voter, since this would reveal whether the coerced vote was later changed or not.

In conclusion, the feedback notification should just acknowledge the fact of receiving a vote by the system, but not much else (including the exact time, or the information whether it was a re-vote or not; see Section 3.3 for further discussion). Such a confirmation would be the most in line with the current legislation, not requiring to rethink how vote secrecy should be understood and protected.

In Estonia, such a system would be relatively easy to implement, since from 2021, electronic voter lists will be deployed. Amongst other features, it would enable the possibility to give voters automatic feedback whether they have voted, since this information is entered in the electronic voter list in real time.

Electronic voter lists make it possible for all (i.e. both paper and electronic) voters to receive such notifications. This is a positive outcome, since equal treatment of paper ballot and Internet voters has been a source of disagreement in Estonia before [9].

3.2 Setting up the feedback channel and automation

The method of giving feedback should be considered as well. The feedback channel should be set up in a way that the information is easily accessible only to the voter. At the same time, it should be universal enough so that as many voters as possible are able to get this confirmation. An example would be an e-mail or SMS sent to the voter. The message can contain just the notification on the fact of voting, or an access link requiring further identification (eID in Estonia's case).

The biggest advantage of using automated feedback is that it would notify the voters if their credentials have been used to cast the vote. So if the voter's electronic ID has been compromised and a vote has been cast on the voter's behalf, the voter would be notified immediately and would be able to take action.

In Estonia, one logical solution would be to use State Portal eesti.ee to store and send receipts, as already suggested in the 2020 study on feasibility of mobile voting [16]. This is accessible to every voter using eID, and every ID-card user gets automatically an e-mail address at eesti.ee. Eesti.ee also includes a mail forwarding service which residents can set up to forward this information their main e-mail address. Other government services and the Population Registry share the data about residents' contacts with eesti.ee portal, making the voter contact database fairly accurate and up-to-date [10].³ An example of a current voting related service that uses eesti.ee portal is the possibility to order electronic voter cards instead of voter cards sent on paper by post.

Eesti.ee contact information enables to send messages to most of the voters, and the voters would get this information using their eID (recall that ID-cards in Estonia are mandatory). Hence, such a feedback method would be both relatively easy to implement and the message ("I voted") easy to understand. Since coercion-resistance measures can be difficult to implement or, indeed, difficult for the voters to understand [34], this is suitable as the next step towards giving voters more assurance about how their votes are handled. Using eesti.ee service as a gateway would also mitigate the problem that an attacker can send out fake notifications *en masse* [28].

3.3 Information provided by the feedback

As noted above, the Estonian re-voting scheme relies, amongst other features, on the element of uncertainty, assuming the malefactor has no way of knowing which was the last vote cast by the voter or whether the voter re-voted. This holds equally for both small- and large-scale coercion attacks (e.g. vote buying). Thus, it is important to give as much information as necessary and as little as possible in the feedback.

The electronic list of voters includes information on the date and time of voting, voting method used (including i-voting) and of course the fact of voting itself. Additionally, the voting system logs more data on the voter, including the age, the operating system used, IP-address etc. [18]. However, since we view the feedback channel as similar to checking voter's information in the list of voters, we restrict our interest to the types of information provided through this list only.

The minimal information included in the voting receipt would be the fact of voting, i.e. confirming that the person has been recorded as having cast a vote.

The method of voting used is another bit of information that is available in the list of voters, the most important distinction here being whether the voter voted over Internet or with a paper ballot. If we would provide this information,

³ The COVID-19 pandemic had a positive side effect in this regard, forcing the government agencies to update people's contact information in order to send out vaccination calls. As of May 2021, 1,260,203 people in the Estonian Population Registry had a valid e-mail address, and 238,162 did not. This means that about 84% of Estonian residents can be reached by email.

it could reveal when the person re-voted with a paper vote, thus weakening the coercion resistance property. On the other hand, this information would give the voter assurance that their (i-)vote has not been changed.

It is also possible to send another confirmation after the voting period has ended, confirming that the voter's i-vote was entered into the count. This differs from checking one's data in the list of voters, since that information can be retrieved only from the Internet voting system before the votes are anonymized. Such information is unavailable at all for paper ballots, which become anonymous once inside the ballot box. This wouldn't reveal more information to the malefactor besides the method of voting, but would give the voter assurance that the i-vote was actually tallied (and not misplaced), which in turn would hopefully increase the trustworthiness of Internet voting to some extent.

Since our goal is to just give confirmation on participating in voting, precise date and time of the vote should not be necessary, although the benefit of giving the voter assurance that their last vote was the one tallied is significant. However, the precise time of the cast of vote might be construed as proof of casting a specific vote which would be advantageous to the malefactor.

3.4 Timing of the feedback

If the feedback is given during the voting period, this would give the malefactor a slight advantage, enabling them to coerce the voter to cast the vote again. If we do not include the date and time of voting in the receipt, the advantage for the malefactor is insignificant, essentially amounting to knowing that the person has voted at some point. Revealing the method used to vote or, for example, the date of voting (without the exact time) gives some additional information, showing possibly that an i-voter has re-voted in the polling station.

If the voting receipt is given after the voting period, then this would give the malefactor even less advantage, since the voter cannot re-cast the vote any more.

However, the advantage of giving feedback during the voting period is that it enables the voter to either re-vote if necessary, or file a complaint with a chance that the complaint will be resolved during the voting period. Instant feedback would also notify the voter if a vote has been cast using their credentials, thus exposing malicious takeovers of voters' electronic ID. If the complaint is filed after the end of the voting period, the voter has essentially no recovery mechanisms available. Even if the National Election Committee and/or the Supreme Court accept that the electoral law has been violated, the voter cannot cast a new vote after the voting has ended. The existing individual vote verification mechanism can be easily extended so that it would also provide a partial integrity check [28].

4 Conclusions and future work

The debate on the secrecy of vote has often concentrated on the fact of secrecy of vote itself, as if the secrecy is the definitive measure to guarantee free and

fair elections. This is certainly commendable, but one should not forget that the concept of secret ballot does not exist in a vacuum. "Old" Western countries take some justified pride in how the understanding of vote secrecy is ingrained in their society. However, this concept works well only for on-site voting, but the modern voting environment encompasses different popular voting solutions for off-site voting as well. We agree with the interpretation suggested by Madise *et al.* that vote secrecy is not the ultimate goal, rather than a necessary means to achieve free and fair elections. Vote secrecy is just one part of the equation. We need to maintain trust in the voting system by addressing other possible issues as well. Voters are more and more moving away from the polling places and off-site voting methods like postal voting, voting at home and i-voting gain more and more traction. It is inevitable that some conflict is built in here, but even so we must try to seek for a good balance in regards to vote secrecy and transparency.

One of the weak points is the voters' and observers' inability to observe and track the path of their ballot. In a way, i-voting has opened a Pandora's Box which made voters question voting methods and trustworthiness of elections in general. Whether aforementioned inability is real or perceived doesn't even matter, since trust is ultimately based on what people think, not what they are told by the election authority. Recent debates in Estonia (but also surely in many other countries) have shown the need to consider voter's trust in the system as a whole and to address these concerns. Therefore we propose to augment the system with a feedback channel allowing the voter to detect misuses of the voting credentials.

We recommend giving automatic feedback to voters on their voting: the method they used to vote as well as the day (but not the time) they voted. This would enable the voters to get assurance that their vote was cast and received as intended, that their vote was not changed later and, in case of abstention, no one voted using the voter's credentials. Making this feedback automatic (e.g. in Estonia through state portal eesti.ee) guarantees that most of the electorate will receive this notification, creating a new layer of verifiability for the system. The ballot count will still remain anonymous and a voter cannot link their vote to a counted vote, a necessary concession to support secrecy and coercion-resistance of the vote.

Establishing such an automated personal feedback channel to voters is not necessarily in conflict with the principle of secret suffrage when restricted just to the fact of voting. It is similar to a voter accessing one's data in the voter list, although the final verdict depends on the amount of data revealed. Determining a good balance between secrecy and transparency is a subject for further discussion. It would also seem that a feedback channel requires some amendments to the legislation, since it concerns processing voting data. Working out the exact nature of such amendments remains the subject for future research as well. We also hope that the debate over secrecy of the vote, what this entails and on how to handle this in a modern voting environment, will continue.

Acknowledgements This paper has been supported by the Estonian Research Council under the grant number PRG920. The authors are grateful to the Estonian Information System Authority and State Electoral Office for their support to the research process.

References

1. Universal Declaration of Human Rights (1948), <https://www.un.org/en/about-us/universal-declaration-of-human-rights>, united Nations
2. European Convention on Human Rights (1950), https://www.echr.coe.int/documents/convention_eng.pdf, European Court of Human Rights
3. International Covenant on Civil and Political Rights (1966), <https://www.ohchr.org/en/professionalinterest/pages/ccpr.aspx>, united Nations
4. CCPR General Comment No. 25: Article 25 (Participation in Public Affairs and the Right to Vote), The Right to Participate in Public Affairs, Voting Rights and the Right of Equal Access to Public Service (1996), https://ccprcentre.org/page/view/general_comments/28883, united Nations Committee on Human Rights
5. Vaalilaki, last amended 1.01.2021 (1998), <https://finlex.fi/fi/laki/ajantasa/1998/19980714>, parliament of Finland
6. Code of Good Practice In Electoral Matters: Guidelines and Explanatory Report (2002), <https://rm.coe.int/090000168092af01>, European Commission for Democracy Through Law (Venice Commission)
7. Federal electoral regulations (2002), https://www.bundeswahlleiter.de/en/dam/jcr/e146a529-fd3b-4131-9588-8242c283537a/bundeswahlordnung_engl.pdf, bundestag
8. Riigikogu Election Act, RT I 2002, 57, 355; RT I, 03.01.2020, 2 (2002), <https://www.riigiteataja.ee/en/eli/514122020002/consolide>, parliament of Estonia
9. Constitutional judgment 3-4-1-13-05: Petition of the President of the Republic to declare the Local Government Council Election Act Amendment Act, passed by the Riigikogu on 28 June 2005, unconstitutional (2005), <https://www.riigikohus.ee/en/constitutional-judgment-3-4-1-13-05>, supreme Court of Estonia
10. Vabariigi Valitsuse määrus Eesti teabevärava eesti.ee haldamise, teabe kättesaadavaks tegemise, arendamise ning kasutamise nõuded ja kord, RT I, 25.03.2021, 5 (2013), <https://www.riigiteataja.ee/akt/125032021005>, government of Estonia
11. ECLI:NL:RBDHA:2014:5657, Rechtbank Den Haag (RBDHA) (2014), <https://e-justice.europa.eu/ecli/ECLI:NL:RBDHA:2014:5657>, court of the Hague, Netherlands
12. Recommendation CM/Rec(2017)5 of the Committee of Ministers to member States on standards for e-voting (2017), <https://rm.coe.int/090000168092af01>, council of Europe Committee of Ministers
13. E-valimiste turvalisuse tööühma koondaruanne (2019), Estonian Ministry of Economic Affairs and Communications, https://www.mkm.ee/sites/default/files/content-editors/e-valimiste_tooruhma_koondaruanne_12.12.2019_0.pdf, in Estonian
14. Statistics about Internet voting in Estonia (2019), <https://www.valimised.ee/en/archive/statistics-about-internet-voting-estonia>
15. Case ECH-2020-1-002 Magyar Kétfarkú Kutya Párt v. Hungary (2020), <http://www.codices.coe.int/NXT/gateway.dll/CODICES/precis/eng/EUR/ECH/ECH-2020-1-002>, european Court of Human Rights

16. Mobile voting feasibility study and risk analysis (2020), report number T-184-5, Cybernetica AS, https://www.valimised.ee/sites/default/files/uploads/eng/2020_m-voting-report.pdf
17. E-voting: Online voting and elections (2021), <https://www.post.ch/en/business-solutions/e-voting>
18. Vabariigi Valimiskomisjoni otsus "Tehnilised nõuded elektroonilise hääletamise üldpõhimõtete tagamiseks", RT III, 27.01.2021, 6 (2021), <https://www.riigiteataja.ee/akt/327012021006>, estonian National Electoral Committee
19. Annus, T.: Riigiõigus. Juura (2006), in Estonian
20. Benaloh, J.: Rethinking voter coercion: The realities imposed by technology. In: 2013 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections, EVT/WOTE '13, Washington, D.C., USA, August 12-13, 2013. USENIX Association (2013), <https://www.usenix.org/conference/evtwote13/workshop-program/presentation/benaloh>
21. Binder, Nadja Braun Binder; Krimmer Robert; Wenda, G.D.H.F.: International Standards and ICT Projects in Public Administration: Introducing Electronic Voting in Norway, Estonia and Switzerland Compared. *Halduskultuur: The Estonian Journal of Administrative Culture and Digital Governance* **19(2)**, 8–21 (2019)
22. Brent, P.: The Australian ballot: Not the secret ballot. *Australian Journal of Political Science* **41(1)**, 39–50 (2006)
23. Buchstein, H.: Online Democracy, Is it Viable? Is it Desirable? Internet Voting and Normative Democratic Theory. In: Kersting, N., Baldersheim, H. (eds.) *Electronic Voting and Democracy: A Comparative Analysis*, p. 97–108. Palgrave Macmillan UK (2004)
24. Drechsler, W., Madise, Ü.: Electronic Voting in Estonia. In: Kersting, N., Baldersheim, H. (eds.) *Electronic Voting and Democracy: A Comparative Analysis*, p. 97–108. Palgrave Macmillan UK (2004)
25. Elklit, J.: Is voting in Sweden secret? An illustration of the challenges in reaching electoral integrity. In: IPSA World Congress, University of Brisbane (2018)
26. Barrat i Esteve, J., Goldsmith, B., Turner, J.: Compliance with International Standards (2021), https://www.regjeringen.no/globalassets/upload/krd/prosjekter/e-valg/evaluering/topic4_assessment.pdf
27. Gerber, A.S., Huber, G.A., Doherty, D., Dowling, C.M.: Is there a secret ballot? Ballot secrecy perceptions and their implications for voting behaviour. *British Journal of Political Science* pp. 77–102 (2013)
28. Heiberg, S., Krips, K., Willemson, J.: Planning the next steps for estonian internet voting. In: *Proceedings of E-Vote-ID 2020*. p. 82 (2020)
29. Heiberg, S., Martens, T., Vinkel, P., Willemson, J.: Improving the Verifiability of the Estonian Internet Voting Scheme. In: *Electronic Voting - First International Joint Conference, E-Vote-ID 2016*, Bregenz, Austria, October 18-21, 2016, *Proceedings. Lecture Notes in Computer Science*, vol. 10141, pp. 92–107. Springer (2016). https://doi.org/10.1007/978-3-319-52240-1_6
30. Heiberg, S., Parsovs, A., Willemson, J.: Log Analysis of Estonian Internet Voting 2013–2015. *Cryptology ePrint Archive*, Report 2015/1211 (2015), <https://eprint.iacr.org/2015/1211>
31. Heiberg, S., Willemson, J.: Modeling threats of a voting method. In: *Design, Development, and Use of Secure Electronic Voting Systems*, pp. 128–148. IGI Global (2014)
32. Heiberg, S., Willemson, J.: Verifiable internet voting in Estonia. In: *6th International Conference on Electronic Voting: Verifying the Vote, EVOTE 2014*,

- Lochau / Bregenz, Austria, October 29-31, 2014. pp. 1–8. IEEE (2014). <https://doi.org/10.1109/EVOTE.2014.7001135>
33. Jääskeläinen, A.: The Finnish Election System: Overview (2020), Oikeusministeriö
 34. Krips, K., Willemson, J.: On practical aspects of coercion-resistant remote voting systems. In: International Joint Conference on Electronic Voting. pp. 216–232. Springer (2019)
 35. Madise, Ü., Martens, T.: E-voting in estonia 2005. the first practice of country-wide binding internet voting in the world. In: Krimmer, R. (ed.) Electronic Voting 2006 – 2nd International Workshop, Co-organized by Council of Europe, ESF TED, IFIP WG 8.6 and E-Voting.CC, pp. 15–26. Gesellschaft für Informatik e.V., Bonn (2006)
 36. Madise, Ü., Priit, V.: Constitutionality of remote internet voting: The Estonian perspective. *Juridica Int'l* **18**, 4 (2011)
 37. Madise, Ü., Vinkel, P.: Internet Voting in Estonia: From Constitutional Debate to Evaluation of Experience over Six Elections. In: Kerikmäe, T. (ed.) *Regulating eTechnologies in the European Union: Normative Realities and Trends*, pp. 53–72. Springer International Publishing (2014)
 38. Madise, Ü., Vinkel, P.: A judicial approach to internet voting in Estonia. In: *E-Voting Case Law*, pp. 135–158. Routledge (2016)
 39. Nemčok, M., Peltoniemi, J.: Distance and trust: An examination of the two opposing factors impacting adoption of postal voting among citizens living abroad. *Political Behavior* pp. 1–25 (2021)
 40. Vollan, K.: Voting in uncontrolled environment and the secrecy of the vote. In: *Electronic Voting 2006–2nd International Workshop, Co-organized by Council of Europe, ESF TED, IFIP WG 8.6 and E-Voting. CC. Gesellschaft für Informatik eV* (2006)
 41. Wasley, P.: Back When Everyone Knew How You Voted. *Humanities* **37**(4) (2016)
 42. Willemson, J.: Bits or paper: Which should get to carry your vote? *J. Inf. Secur. Appl.* **38**, 124–131 (2018). <https://doi.org/10.1016/j.jisa.2017.11.007>

Australian Senate Scrutiny Reform Proposal

Ian Brightwell¹

¹ Consultant DH4 Pty Ltd

Abstract. The Australian Senate is one of the two houses of the Australian Federal Parliament and uses a Single Transferable Vote with Proportional Representation count method to elect its 76 members. The election of senators is done by state and overseen by the Australian Electoral Commission's (AEC) state managers. The election count is complex with a contest having up to 5 million ballots. The count currently is completed by using a computer to both capture paper ballot preferences and distribute vote preferences. The current election scrutiny process only allows scrutiny to be done by electors appointed by candidates for a given contest. The scrutineers can only view ballots as they are scanned or keyed into the ballot capture system and then view scanned images of the ballots as they are manually keyed or checked. Current scrutiny does not allow for a systematic audit of the ballot capture system or independent validation of the ballots scanned against ballots captured, nor does it allow validation of the final distribution of preferences. This paper proposes reforms to election laws to create a new independent body to undertake specialised election audits to confirm the ballots cast are reflective of the ballots captured in the count system and also undertake an independent check of the preference distribution process.

Keywords: Australian Senate, Scrutiny, Audit, proportional representation.

1 Australian Senate Elections

1.1 Senate Election History

The Australian Senate is one of the two houses of the Australian Federal Parliament. The Senate currently consists of 76 senators, twelve from each of the six states and two from the two mainland territories. Each state and territory elect senators in separate contests run at general elections. The returning officer for each state's election are the state managers of the Australian Electoral Commission (AEC). The election count is complex and can be large, with the New South Wales (NSW) state contest having up to 5 million ballots. Typically, only half the senate is elected at a given general election event resulting in senators typically having two terms. However occasionally when there is a parliamentary deadlock a double dissolution election is called and all 76 Senators are elected at one time.

The system for electing senators has changed several times since Federation in 1901. Initially a 'first-past-the-post' voting method was used, then in 1919 it was replaced with a preferential block voting method [1]. In 1948 electing senators was changed to a Single Transferable Vote with Proportional Representation (STV-PR) method with contests on a state-by-state basis [2]. Then, in 1983 Group Voting Ticket (GVT) were introduced to the STV-PR method to address high rates of informal voting.

1.2 Group Voting Tickets (GVT)

The use of GVT had the anticipated effect of simplifying the voting process by allowing voters to vote formally by simply marking a single preference for one Above the Line (ATL) group (see Annexure 1 in reference [3] for example ballot paper). Voting for single preference ATL had the effect of marking preferences in accordance with the registered GVT for that group.

Each group was required at time of candidate nomination to list all the BTL votes they wanted marked as a result of an ATL vote for their group. This meant that an elector that made an ATL vote for a group effectively was voting for all the BTL votes previously nominated by that group at the time of nomination. The main problem with GVT voting was that many voters did not know most of the BTL candidates they actually voted for beyond those in the ATL group for which they voted. This is because electors rarely knew what was on the GVT for the given group for which they had voted.

Notwithstanding the introduction of the GVT feature voters were still able to vote directly for individual candidates BTL. This was done by giving every BTL candidate a sequential preferences number starting at 1. The challenge for voters was to number every BTL box with a sequential number without a break or repeat [4]. It was found for elections between 1984 to 2013 only about 3% [5] of voters actually voted successfully BTL. Interestingly, there was a very high rate of informality for BTL ballots due to the challenge of marking preferences for all BTL candidates sequentially when potentially hundreds of candidates could be on a ballot.

Given the lack of BTL votes for elections between 1984 to 2013 the counting of these ballots was largely a manual process. The 97% of ATL ballots were marked as a single preference only vote and could be manually checked for formality and tallied by ATL group. These manually tallied ballots were then entered into the EasyCount computer system as one bulk figure for each ATL group. The remaining 3% of BTL votes were captured by keying all their BTL preferences directly into EasyCount ballot by ballot.

EasyCount would then combine these BTL votes with expanded ATL group votes using their associated GVT. The expansion was done by applying the GVT to the ATL vote to give each ballot's BTL candidate preferences. Once all the ATL and BTL vote preferences were available by candidate in EasyCount, EasyCount was used to perform the STV-PR distribution of preferences (DoP) and identified the candidates the Returning Officer would declare elected.

1.3 Gaming the System

GVTs were first used in the 1984 election. Initially its introduction was considered a great success as informal voting decreased. However, it gradually started to produce perverse election results. The main problem with the use of GVTs was that micro parties (Groups) could collectively 'game' the system by participating in 'preference harvesting' arrangements. Preference harvesting relies on disaffected or poorly informed voters, voting for parties (groups) which have populist names like 'Fishing and Lifestyle' and 'Smokers Rights'. These voters believed they were making a statement for these causes but in effect their vote was harvested to potentially elect a

candidate for a collaborating party they may have no interest in supporting e.g. ‘Monitoring Enthusiast Party’ [6].

This perverse outcome was achieved by several micro parties (which had group voting squares ATL) banding together and agreeing to cross preference each other’s GVTs. The effect of this collaboration was to keep their vote preferences within the collaborating parties, allowing these group’s votes to accumulate significantly more preferences than they would have individually. This meant that one of the candidates from the collaborating parties may be elected by the accumulated preferences of these parties, exceeding the residual preferences held by any of the major party candidates still in the count.

This type of ‘gaming’ was in part responsible for the 2013 Western Australian (WA) senate election being rerun. The WA 2013 election had a large number of populist minor and micro parties with ATL voting squares participating in ‘preference harvesting’ arrangements. This practice resulted in a ballot paper where micro parties engineered their GVTs in such a way that they won seats initially in the senate even though the elected candidate had very few first preference votes [4].

The second issue was that the DoP result was close for the last candidate elected and the next candidate, as a result the Electoral Commissioner decided to undertake a recount [7]. During the recount of the 2013 WA senate election some 1375 ballots out of 1.31 million ballots were lost in transport between the initial count centre and the recount centre. The recount found that when 14 of these 1375 missing ballots were omitted from the count two of the six candidates elected in the initial count changed in the recount [8].

Obviously, the proportionality of the change in candidates elected (2 out of 6) compared with the number of lost ballots (1375 vs 1.31M) highlighted the intrinsic instability of the STV-PR when combined with GVTs. The AEC then lodged a petition with the High Court which sought an order to declare the WA Senate election void, thus requiring a rerun election.

As a result of the debacle with the 2013 WA senate election, the parliament removed the use of GVT from the STV-PR method to elect senators. The subsequent election in 2016 replaced GVT with optional preferential voting for both above and below the line [2]. The result of these changes allowed electors to assign their preferences to a minimum number of groups above the line, or a minimum number of candidates below the line, and were not required to fill all of the ATL or BTL boxes.



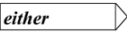

This change effectively resolved the intrinsic instability of the previous STV-PR system when combined with GVT. However, this change also resulted in 100% of ballots needing to be captured by computer, because the manual counting of single ATL preferences associated with a GVT was no longer effective. Computer capturing of 100% of ballots was a significant change from the 3% captured by computer prior to 2016.

1.4 Senate Ballot

The Senate election comprises a paper-based vote which is captured and counted in a computer. Contests are state-based so each State runs its own senate election independently. Figure 1 below shows a template of a ballot paper for a current typical senate election i.e. elections held in 2016 and 2019 used this ballot layout. This new

ballot has two parts, electors could choose to either vote ATL or BTL, but not both. The ballot in Figure 1 is for a state-based half Senate election so the elector only needs to number a minimum of 6 preferences above the line or 12 below. That means that even for a half senate election there will be a minimum of 36 million preferences to capture and count when dealing with 5 million ballots, which could be expected in a New South Wales senate election.

Figure 1. Typical Senate Ballot Paper for all elections held after 2013.

		SENATE BALLOT PAPER (5) ELECTION OF (6) SENATORS							
<p> </p> <p>You may vote in one of two ways</p>									
<p>either </p>									
<p>By numbering at least 6 of these boxes in the order of your choice (with number 1 as your first choice)*</p>									
	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	
	A	B	C	D	E	F	G	H	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	(2)	(2)	(2)	(4)	(4)	(2)	(4)	(4)	
<p>or </p>									
<p>By numbering at least 12 of these boxes in the order of your choice (with number 1 as your first choice)**</p>									
	A	B	C	D	E	F	G	H	Ungrouped
	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	
	<input type="checkbox"/> (1) <input type="checkbox"/> (3)	<input type="checkbox"/> (1) <input type="checkbox"/> (3)	<input type="checkbox"/> (1) <input type="checkbox"/> (3)	<input type="checkbox"/> (1) <input type="checkbox"/> (4)	<input type="checkbox"/> (1) <input type="checkbox"/> (4)	<input type="checkbox"/> (1) <input type="checkbox"/> (3)	<input type="checkbox"/> (1) <input type="checkbox"/> (4)	<input type="checkbox"/> (1) <input type="checkbox"/> (4)	
	<input type="checkbox"/> (1) <input type="checkbox"/> (3)	<input type="checkbox"/> (1) <input type="checkbox"/> (3)	<input type="checkbox"/> (1) <input type="checkbox"/> (3)		<input type="checkbox"/> (1) <input type="checkbox"/> (4)	<input type="checkbox"/> (1) <input type="checkbox"/> (3)	<input type="checkbox"/> (1) <input type="checkbox"/> (4)	<input type="checkbox"/> (1) <input type="checkbox"/> (4)	
	<input type="checkbox"/> (1) <input type="checkbox"/> (3)	<input type="checkbox"/> (1) <input type="checkbox"/> (3)	<input type="checkbox"/> (1) <input type="checkbox"/> (3)			<input type="checkbox"/> (1) <input type="checkbox"/> (3)			

(1) Here insert name of a candidate.
(2) Here insert name of a registered political party or composite name of registered political parties if to be printed.
(3) Here insert the name of a registered political party if to be printed.
(4) Here insert name of a registered political party or word 'Independent' if to be printed.
(5) Here insert name of State or Territory and year of election.
(6) Here insert number of vacancies.
(8) Here insert the logo of a registered political party if to be printed.

* If the ballot paper has 6 or fewer squares above the line, replace the instruction with "By numbering these boxes in the order of your choice (with number 1 as your first choice)".
** If the ballot paper has 12 or fewer squares below the line, replace the instruction with "By numbering these boxes in the order of your choice (with number 1 as your first choice)".

1.5 Senate Counting by Computer

The use of computers for capturing and distributing senate preferences is not new. Given the heavy computational nature of STV-PR election counting, computers were used from the middle of the 1990s to count elections. Below is an extract from a letter provided by the AEC to the Inquiry into the 2013 federal election[9].

In 1995-1996 the AEC developed the EasyCount software used for Senate vote counting as in-house software. This had been precipitated by the JSCEM's Report of the Inquiry into the Conduct of the 1993 Election which recommended that the Electoral Act be amended to permit the Senate scrutiny to be performed either by the existing manual processes or by computerisation.

When the EasyCount system was introduced 97% of ballots were single preference ATL and could be counted manually with their results 'bulk data entered' into EasyCount for distribution. This meant that scrutineers could watch 97% of ballots being manually counted in much the same way they had since 1984 when STV-PR when GVT was introduced. However, the remaining 3% of BTL ballots needed to have their preference markings captured into the EasyCount system. Given these ballots typically had little impact on the outcome of elections there was no need to im-

prove scrutiny of the preference capture process as it was of little electoral significance.

However, a significant change to ballot counting happened at the 2016 election when all ballot papers had to have their preferences captured individually into a computer system for distribution. The AEC had to develop this new system in a very short time. This system would have to be able to capture every preference on every ballot regardless of whether the elector voted ATL or BTL.

This change to voting meant that 100% of ballot preference markings would have to be processed in a computer system. Therefore, a computer was trusted to determine the electoral outcome from the time of ballot capture to the point where candidates were declared elected. Previously 97% of the senate votes were counted manually in regional counting centres with oversight from scrutineers. This manual counting meant all elected candidates could be identified from raw bulk entry data (with reasonable certainty) prior to the final computer-based DoP.

To deal with the need to capture and count all senate ballots using a computer, the AEC developed a new system in partnership with Fuji Xerox Document Management Solutions. This system scanned every ballot and automatically recognised each preference mark using optical character recognition technology. The system was developed in a very short period of time by outsourcing partner Fuji Xerox. A flow diagram for the new system is provided in IDM article “New Senate Count Solution in 12 Weeks” [11] and Figure 4.1: Simplified diagram of Senate scanning system of the “ANAO report Australian Electoral Commission’s Procurement of Services for the Conduct of the 2016 Federal Election” [12].

EasyCount was still the software used at the 2016 election to perform DoP. EasyCount was modified to cope with the removal of GVT from the process but essentially was the same unscrutinised system it had been since its creation [10].

2 Senate Scrutiny Practices

2.1 Appointment of Scrutineers

In Australia candidates are not allowed to scrutinise their own election contest, so candidates rely exclusively on appointed scrutineers to provide independent scrutiny of parliamentary elections. Scrutineers are appointed by candidates with one scrutineer allowed per candidate per venue. Appointment is done by completion of an application form [14]. Scrutineers are typically family and friends of candidates or for candidates of major parties they may be party members. Details of the current scrutiny process are defined in the AEC’s scrutineer’s handbook [13]. Typically, family and friend scrutineers have little knowledge of the electoral process, while party scrutineers typically have more knowledge of electoral process but only tend to look at a limited number of individual ballot marks not the integrity of the overall end to end electoral process and typically have no experience in auditing complex computer systems.

2.2 Scrutiny in Polling Places

A polling place is a voting venue which takes attendance votes on election day and undertakes a provisional count on election night. At a federal election there are over 10,000 Polling Places on election day open for voting between 8am to 6pm across Australia. Over 60% of votes taken at federal elections are taken at Polling Places on election day.

Scrutineers at polling places are typically party workers for given candidates. These people during the course of election day work at the polling place and hand out 'how to vote' pamphlets to voters as they attend to vote. Each candidate can appoint one of these party workers to be their scrutineer at each polling place.

On election day a scrutineer may perform several functions at a polling place. These include:

- Inspect and confirm the ballot box is empty before it is sealed closed in the morning before 8am;
- Observe voting inside the polling place;
- Assist disabled voters vote when requested;
- Observe when the ballot box is being opened after 6pm;
- Observe the unfolding and counting of ballots;
- Keep a tally of votes counted to try and determine who may win; and
- Observe the sealing of counted ballots into bundles ready for transport for subsequent final counting.

Unfortunately, most scrutiny in polling places is done for the lower house ballots which are counted first on the night. The independent scrutiny of senate ballots is negligible as most scrutineers leave before the Senate provisional count is finished. This is because the election of government in Australia is determined by control of the lower house hence once the provisional count is done for the lower house there is little interest on election night, for volunteers after a long day, to stay to scrutinise the upper house.

Notwithstanding this practice of scrutineers leaving before the senate is provisionally counted, the nature of staffing of polling places is such that, the staff themselves are in effect a relatively independent pool of observers of the election process and as such provide a level of independent scrutiny to the senate count process.

Given the above I suggest that the current partisan scrutiny processes used by the AEC for federal elections is an efficient and effective process in Polling Places and should not be changed. However, it may be appropriate to provide some incentive for scrutineers to stay on election night until the Senate provisional count is completed and securely bundled.

2.3 Scrutiny in Regional Count Centres

After election day all ballots are taken to regional count centres for further sorting and counting. The Senate Polling Place ballots are only checked to ensure no ballots have been lost in transit and that they are correctly packed for processing at the state Central Senate Scrutiny (CSS) centre.

Other senate ballots from early voting centres and declaration votes (e.g. postal, absent and other provisional votes) are provisionally counted in these regional count centres. The counting process used is the same process used in Polling Places on election night. Once counted the senate ballots are bundled ready for transport to the CSS in the same way ballots from Polling Places are bundled.

The count centres are often large warehouses with hundreds of people performing tasks on piles of ballots. Scrutineers can attend and observe operations at Regional Count Centres, however there is little opportunity to perform any effective scrutiny of the processes. The size of the count centres is such that effective scrutiny of the count process by an individual scrutineer is virtually impossible. It is difficult for a scrutineer to identify any failures in the process or lost ballots due to the lack of meaningful data about the overall process. Effective scrutiny of these types of operations could only be undertaken by someone who is skilled in auditing large manual operations and had access to adequate count data and significant analytical capability. Scrutiny in these centres needs to be upgraded if it is to be effective.

2.4 Central Senate Scrutiny (CSS)

Once senate ballots are bundled at the Regional Count Centres they are forwarded to the state based CSS for scanning of all preferences. The scanning process is outlined in Figure 2 and uses scanning equipment and management system provided by the AEC's contractor Fuji Xerox. Figure 2 clearly shows that most of the work done in CSS is under Fuji Xerox control.

Scrutiny of the CSS is currently ineffective as scrutineers are only allowed to watch ballots being scanned from a distance, then view individual ballots being keyed and/or validated by Fuji Xerox or AEC operators. The AEC does not provide any statistical information about the process nor does it perform any cross checking of ballots entered against ballots captured.

The author was appointed to be a NSW Senate scrutineer for a minor party candidate at the 2019 federal election and requested the information below to allow an audit of the scanning process at the CSS [15].

1. What percentage of ballots scanned are streamed to "Data Entry #1", "Perfect Capture" and "Unmarked"?
2. Please confirm all ballot's preferences are captured by keying in "Data Entry #2"?
3. What percentage of the ballots that pass through the "Compare data entry" process are found to be "Mismatched"?
4. What percentage of the "Mismatched" ballots are passed to the "Exception Check" process?
5. What percentage of the "Exception Check" ballots are escalated to the "AEC Adjudication" queue?
6. I understand that the "AEC Adjudication" queue is also fed from ballots that have a "shield" which cannot be detected or are blank but have been placed in a formal batch and these ballots are assessed only by an AEC staff member without further scrutiny. Is this correct and if so what percentage of all ballots processed are assessed in this manner?
7. What is the variance threshold between the total number of HoR and senate ballots counted for a given polling place or dec vote type (for a given division) which

head office requires before CSS staff and/or DRO staff are requested to recount ballots or search for missing ballots?

8. Where discrepancy between HoR and senate are outside tolerance and the discrepancies can be attributed to ballots being moved between polling places, are these discrepancies addressed by ballots being moved between polling places and batches rescanned?
9. Will the AEC be undertaking cross checking of a statistically significant sample of paper ballots against the corresponding preference data in the CSS's output file? If so will this happen before the declaration of the poll and will scrutineers be able to witness this process?

The AEC did not provide any of the requested information above which would in the author's view be the minimum amount of information required to allow even the most superficial assessment of the 2019 CSS operations. The AEC justified their action by relying on the Electoral Act which did not specifically require this information to be provided. The AEC's position was that unless they are specifically required to provide information under their legislation, they will not provide it.

A similar request for information was made to the AEC at the 2016 senate election. Again, this request was effectively ignored by the AEC. The author and colleagues made several submissions to JSCEM for the 2016 election which are listed in references [16] [17].

3 Current Scrutiny Risks

3.1 National Audit Office Report

The Australian National Audit Office (ANAO) reported on the integrity of the AEC's senate count processes in their report "Australian Electoral Commission's Procurement of Services for the Conduct of the 2016 Federal Election" in January 2018 [12]. In this report the ANAO concluded:

7. *The AEC addressed risks to the security and integrity of ballot paper data through the design and testing of the Senate scanning system. **The AEC accepted IT security risk above its usual tolerance.** Insufficient attention was paid to ensuring the AEC could identify whether the system had been compromised.*

8. *The Senate scanning and transport suppliers delivered the services as contracted. **The AEC had limited insight into whether its contractual and procedural risk treatments were effective.** Going forward, the AEC needs to be better able to verify and demonstrate the integrity of its electoral data.*

The ANAO audit found:

5.46 *The feedback to the Joint Standing Committee on Electoral Matters indicated that scrutineers generally found it more difficult to confirm the integrity of the Senate count when conducted by the semi-automated system than by the previous manual process.*

This resulted in Recommendation no. 4 of the report (Paragraph 5.47):

When the Australian Electoral Commission uses computer assisted scrutiny¹ in future federal electoral events, the integrity of the data is verified and the findings of the verification activities are reported.

The Australian Electoral Commission's response was to agree with the recommendation with qualification. They said:

5.48 The AEC remains confident that the range of measures put in place for the 2016 federal election ensured the integrity of the Senate count. For future events, the AEC will continue to evaluate and if appropriate, implement additional verification mechanisms to maintain the integrity of the count. The results of verification activities undertaken at future electoral events may be reported in support of the scrutineering process.

Notwithstanding the AEC's undertakings above, there did not appear to be any appreciable change in its approach to the issues raised at the subsequent 2019 election.

3.2 Ballot Capture Risks

One of the main challenges facing any electoral body which captures ballot preference using a computer system is to ensure the system faithfully captures and holds ballot preferences for the DoP process. This may sound a simple task but it is actually very hard to guarantee. No process can be applied to the development and/or operation of a ballot capture system which would guarantee that the ballots as presented to the captured system will be faithfully represented digitally to the count system.

The NSW Electoral Commission (NSWEC) discovered developing an error free capture system was hard when they developed a new proportional representation count system (PRCC) for the 2011 State General Election (SGE). The new system was designed to capture and count both the NSW upper house votes for state elections and local government Council elections. Both these elections used a similar proportional representation count method and captured directly about 20% of the election ballots using a double keying approach. The system in many respects is similar in operation and complexity to the senate count system used by the AEC at their CSS.

The PRCC system was a complete redevelopment of the previous LCLG system and incorporated many new ballot capture features to improve count centre operations and reduce DoP system code complexity. Notwithstanding these improvements the system is complex and like all such systems had a potential for software "bugs" which could impact the election result.

The system was first used at the 2011 state general election. Typically, in these elections, about 80% of ballots are single preference ATL votes. These single preference ATL ballots can be counted manually with appropriate scrutiny supervision and entered as a bulk figure e.g. Group A single preference ATL ballots entered as one figure for a given polling place or District and declaration vote type.

The remaining ballots either had multiple ATL preferences or only had BTL preferences. These ballots required all their preferences to be data entered (keyed) in batches of 50 ballots. The data entry method used double blind keying where two independent operators keyed the same batch and then it was cross checked by PRCC

¹ The use of the word 'scrutiny' in this quote refers to the capture and counting of ballots by the AEC not the oversight by independent scrutineers.

to determine if the two entries matched. If they did not match an independent verification operator adjusted the entered batches until they both matched the ballot paper.

To ensure the system capturing the ballot preferences were keyed correctly, a manual cross-check process was established which involved printing a report from the database to be used by the count. This report showed the preferences for each ballot in a given batch of ballots. Election staff would then work in pairs with one reading out loud the ballot preferences from the report and the other checking it against the preferences on the actual ballot keyed. The objective of this cross-check was to ensure that the data was captured faithfully and was not corrupted by human error/corruption or computer fault.

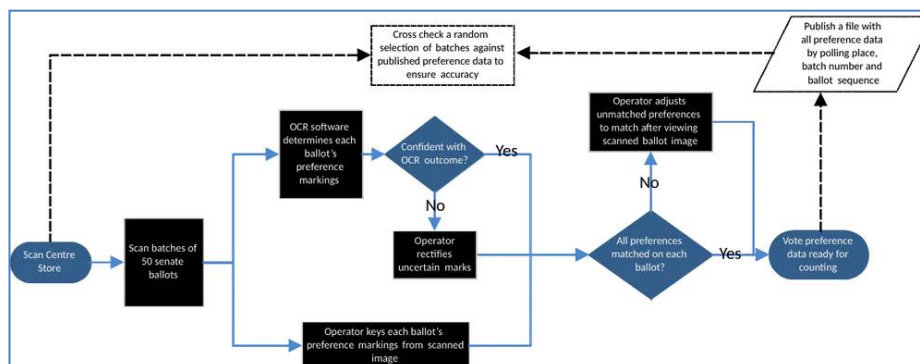
As it turned out at the 2011 SGE, discrepancies were found during cross-checking of ballots. These discrepancies were found to be due to a “bug” in PRCC program which performed the batch reconciliation process. The errant batches were relatively rare and as such cross-checking took several days to identify them and the code fix took a further few more days to implement and test. The impact of this delay was that the data entry process had keyed hundreds of batches which had erroneous preferences captured.

Fortunately, the NSWEC could determine which batches had errors because they knew the nature of the programming error and could search the database to find which batches had a preference pattern which would be impacted by this coding error. They therefore could make a list of batches which had to be rekeyed. This list was used to retrieve erroneous batches from storage for rekeying. Batch retrieval and rekeying was done in the count centre right in front of the scrutineers.

Interestingly, none of the scrutineers in the count centre questioned why the NSWEC were rekeying batches or even appeared to notice the rekeying was happening. This is because scrutineers were only looking at data entry operators keying individual ballots not the overall integrity of the data capture process. This lack of awareness demonstrated that the current scrutiny processes for capture of ballots into PRCC at large count centres simply does not deal with all potential preference capture issues.

It is generally acknowledged by academics working in this area [20][21] that the only viable way to provide a level of certainty that the ballots captured for the senate are the same as those being counted is for independent scrutineers to cross-check a sample of the processed ballots against the same ballots captured digitally. Figure 2 shows how this process would work relative to the current senate ballot capture system.

Figure 2. Senate vote capture process (recommended additional steps shown with dashes) [20]



One of the challenges in implementing a cross-check process for a STV-PR system is to determine the size of the sample needed to ensure an acceptable level of confidence in the election outcome. There has been some research in this area which has shown how complex this problem can be, and the research has provided some guidance on how to arrive at a suitable sample size [21]. It is fair to say however, that any level of cross-check would be better than the current AEC senate CSS practice which is to do no cross-checking!

3.3 Preference Distribution Risks

The STV-PR distribution of preference (counting) algorithm by its nature is complex and difficult to test. This is particularly true for the STV-PR system used by the NSWEC for local government and state elections up to 2016. This algorithm used a random sampling technique for applying the transfer of surplus votes [23]. The random aspect of this count meant that any computer system which undertakes such a count would potentially get a different answer every time it was run. This randomness made testing using standard test cases and outcomes impractical.

Notwithstanding these challenges, the NSWEC took great care in developing and testing the PRCC system. However, the counting code was found in 2016 to have a very subtle “bug”. This “bug” was revealed when several researchers developed their own counting system using the specification provided by the NSWEC [26]. They ran this program with the 2012 election preference data provided by the NSWEC. The outcome of this count was cross-checked against NSWEC results revealing the discrepancies in at least one election contest [22].

Ultimately the NSWEC acknowledged that there was a coding error. This error was fixed for subsequent elections. It should also be noted that this work also identified that the random sampling approached used to pick ballots transferred in the STV-PR count gave unreliable election outcomes and as such also changed. A new system for transferring ballots was implemented for elections after 2016, which made the count more deterministic and helped testing of code used for STV-PR in these elections.

The problems experienced by the NSWEC could be experienced by any election body that uses computers to capture and count votes. The AEC EasyCount system is a similar program to PRCC and equally susceptible to having “bugs”.

There are two actions which can be taken to improve the reliability of the computer based STV-PR count system. The first is making the source-code available for review. The second is encouraging a count system to be developed and operated independent of the AEC to validate the count.

Several researchers have argued that EasyCount code base should be made open-sourced to allow it to be publicly scrutinised. This request was denied by the AEC for a range of reasons [10]. The author’s view is that although open-source code would appear to be desirable from a transparency perspective, it would not in itself necessarily achieve a more reliable system but could consume a lot of valuable AEC resources answering questions asked by people who just have an interest in the area but no real technical expertise or knowledge of the STV-PR method. As such, these people would not be able to add value to the integrity of the system.

I do however believe that providing the code to select expert groups who have agreed to use responsible disclosure could improve system integrity. These groups would have to have suitable technical qualifications to understand the code and be able to build and run the system. They would also need to be provided appropriate support from the AEC when reviewing the code.

In addition to code review, the AEC could provide publicly detailed specifications for the system such that independent groups could build a system which could be run in parallel using the current election preference data. This approach would allow an independent validation of the DoP process and the candidates elected against AEC results. This type of independent check-count could be included into the election scrutiny process and reported to the AEC before declaring candidates elected and subsequently to the Electoral Matters Committee after the election.

To support this type of checking process the NSWEC publishes ballot/preference input data, in a form which is suitable for counting by an independently developed count system. The AEC currently publishes senate preference data, but the format of this data is difficult to use for counting votes and is missing some information such as ballot formality which is needed to run an independent DoP.

4 Recommended Scrutiny Changes

As a result of the general inability to scrutinise the 2016 federal election senate count, submissions were made to the Joint Standing Committee on Electoral Matters (JSCEM) which resulted in their report [18] making the following specific recommendation:

Recommendation 3 - The Committee recommends that a non-partisan independent expert scrutineer be appointed to each Central Senate Scrutiny Centre in each state and territory and be responsible for:

- auditing the computer systems and processes used to capture and count votes;

- *undertaking randomised checks between captured data and physical ballot papers throughout the count at a level that provides surety as to the accuracy of the system; and*
- *providing reports to candidate scrutineers about their findings on a regular basis during the count.*

At the time of writing the government had not provided a response to this report [19]. Notwithstanding that a senior Government member had chaired JSCEM and had produced the report. Also, it should be noted that parliament has a standing order which requires governments to respond to a committee report within six months of it being tabled in parliament. As advised above, no response had been provided to the JSCEM report, even after two years. Note the above JSCEM recommendation is still in the view of the author appropriate and should be implemented for all future elections.

The above recommendation could be achieved by implementing an independent Election Technology Review Board to scrutinise systems used to capture and count votes. This board should be selected before an election is called and comprise suitably qualified people i.e. people with a security engineering and/or technology management and/or election background. These appointments should be made by an organisation at arm's length from the AEC. A suitable entity may be JSCEM with the support of the ANAO. Members could be drawn from academia, professional bodies that have skilled members in system audit.

Should the government consider amendments to legislation to establish an independent board the following issues should be addressed;

- The Board should be able to request and be provided sufficient information about the underlying technology and processes to be able to assess information regarding the adequacy of the system's design, implementation, configuration and testing.
- The Board should be permitted to scrutinise the operation of the system to allow them to determine with appropriate confidence that the system is performing as required.
- The Board should be able to investigate and witness verification of a cross-check process demonstrating that the intent of the voter has been processed in accordance with the legislation and the accuracy of the process is sufficient to be confident the correct candidates have been elected.
- The Board should report on the adequacy of technological processes and digital information produced.
- The Board should provide a publicly available report to the electoral matters committee after each election.
- The Board should have appropriate investigative powers and access to enable it to carry out its functions.

The use of specialist Boards to deal with technology issues in election processes has been implemented in other jurisdictions. In particular, Norway implemented an Internet Election Committee (IEC) for their internet voting election trials in 2013 which had oversight of the trials with a particular focus on security. More information about the committee's work can be found in The Carter Centre's report [24]. Also, Canada has an independent body to oversee elections [25]. Some aspects of this entity's struc-

ture and function may also be applicable in the Australian environment to address the increased complexity of the electoral processes.

To ensure that the Review Board attracts competent people, there is an argument they should receive nominal compensation for their effort (in line with normal government board fees). They also need to be provided adequate secretarial support to assist in writing reports and analysing data.

5 Conclusion

The only viable way to ensure effective senate election scrutiny is for the scrutiny process currently being used in polling places and count centres to be augmented in count centres with a more holistic scrutiny process which examines the efficacy of end-to-end senate election process. This type of scrutiny should be provided by experts who have the skills and resources to perform an effective systems-oriented scrutiny process not the current candidate appointed partisan scrutineers.

This solution would involve a body independent of the AEC to both cross-check ballots captured and to supervise a check of the final DoP. This count-check should be run at about the same time as the AEC runs their count and be used to validate the AEC count. Additionally, the body should be able to review all preliminary count results against final results to determine if any ballots are missing.

Unfortunately changes of this nature require political appetite for electoral reform. The last legislative change of significance was to address the debacle caused by lost ballots and use of GVTs at the 2013 WA senate election. This resulted in an expensive rerun election and GVTs being abolished which resulted in the need to scan all senate ballots.

Notwithstanding this change, there was no associated change in the scrutiny processes. The current scrutiny situation is much like it was in the early 1900s when all the ballot counting was done manually. Unfortunately, the legislative change associated with the removal of GVTs and introduction of computer capture and DoP did not also come with any changes to improve scrutiny. This lack of change to the scrutiny process is, in the author's view, an oversight and should be addressed by a legislative amendment.

At present there does not appear to be sufficient political will for election reform in this area. Therefore, the real challenge to get this type of election reform is to create a political environment for reform. The Australian public has a very high level of trust in the AEC and does not believe it would do anything intentionally to distort election outcomes.

I personally agree that the current AEC is a trustworthy organisation. However, I do not know what the AEC of the future will be like and as such, controls need to be put in place now to protect our democracy against future problems.

Regrettably, history shows that the appetite for change only appears after some type of electoral debacle or public disquiet occurs due to election irregularities. Let us hope reform happens before we really need the type of reforms I have proposed in this paper.

References

1. D. R. Elder, Australian Parliament House, House of Representatives Practice (7th edition), Ch 3, Method of voting, May 2018.
https://www.aph.gov.au/About_Parliament/House_of_Representatives/Powers_practice_and_procedure/Practice7
2. Odgers' Australian Senate Practice, Australian Parliament House, 14th edition, Ch 4, Elections for the Senate, as updated 31 July 2020.
https://www.aph.gov.au/About_Parliament/Senate/Powers_practice_n_procedures/Odgers_Australian_Senate_Practice
3. Australian Electoral Commission, "Inquiry into the 2013 WA Senate Election", December 2013
https://www.aec.gov.au/About_AEC/Publications/files/inquiry-into-the-2013-wa-senate-election.pdf
4. Antony Green, "Is It Time for a Fundamental Review of the Senate's Electoral System?", Australian Parliament House, Papers on Parliament No. 62, October 2014.
https://www.aph.gov.au/~/~/~link.aspx?id=6EAB2F2521E8462CBBBF9EAE79C5229C&_z=z
5. Australian Electoral Commission, Results Website for 2013 Election, Updated: Friday, 01 November 2013 07:22:25 PM.
<https://results.aec.gov.au/17496/Website/SenateUseOfGvtByState-17496.htm>
6. Antony Green, "Ricky Muir's Strange Path to the Senate", ANTONY GREEN'S ELECTION BLOG, Posted Thu 7 Aug 2014 at 12:00am Thursday 7 Aug 2014 at 12:00am, updated Tue 20 Mar 2018 at 2:21pm.
<https://www.abc.net.au/news/2014-08-07/ricky-muir-strange-path-to-the-senate/9388474>
7. Rob Lundie, "The disputed 2013 WA Senate election", Australian Parliament House, Posted 20/11/2013.
https://www.aph.gov.au/About_Parliament/Parliamentary_Departments/Parliamentary_Library/FlagPost/2013/November/The_disputed_2013_WA_Senate_election
8. Antony Green, "Antony Green says WA Senate result comes down to just one vote", Australian Broadcasting Commission, Posted Fri 8 Nov 2013 at 6:05pm.
<https://www.abc.net.au/news/2013-11-08/wa-senate-result-came-down-to-just-one-vote/5080162>
9. Tom Rogers, Acting Australian Electoral Commissioner, Letter to Electoral Matters Committee in "Responses to Questions taken on Notice at the Public Hearing on 31 July 2014", 22 September 2014.
<https://www.aph.gov.au/DocumentStore.ashx?id=8f4b3fca-a2e6-44c2-9f94-3d289dbd2dc2>
10. Tom Rogers, Acting Australian Electoral Commissioner, Letter to Mr Cordover "LS4883 OUTCOME OF INTERNAL REVIEW OF THE DECISION TO REFUSE YOUR FOI REQUEST No. LS4849", 9 December 2013.
<https://www.aec.gov.au/information-access/foi/2014/files/ls4912-1.pdf>
11. Information and Data Management, "New Senate count solution in 12 weeks", Friday, June 2, 2017 - 12:23.
<https://idm.net.au/article/0011523-new-senate-count-solution-12-weeks>
12. ANAO, "Australian Electoral Commission's Procurement of Services for the Conduct of the 2016 Federal Election", AUDITOR-GENERAL REPORT NO.25 OF 2017-18, Monday 22 January 2018.
<https://www.anao.gov.au/work/performance-audit/aec-procurement-services-conduct-2016-federal-election>
13. Australian Electoral Commission, "Scrutineers Handbook", Version 10, 2 October 2020.
<https://www.aec.gov.au/Elections/candidates/files/scrutineers-handbook.pdf>
14. Australian Electoral Commission, "Scrutineer appointment and undertaking form", EF107 140119.

- <https://www.aec.gov.au/Elections/candidates/files/scrutineers-appointment-form.pdf>
15. Ian Brightwell, Submission 10, Electoral Matters Committee, Inquiry into and report on all aspects of the conduct of the 2019 Federal Election and matters related thereto, 28 August 2019.
<https://www.aph.gov.au/DocumentStore.ashx?id=b6f70b16-4427-4704-9804-450ede29f831&subId=669021>
 16. Richard Buckland, Submission 56, Electoral Matters Committee, Inquiry into and report on all aspects of the conduct of the 2016 Federal Election and matters related thereto, 30 October 2016.
<https://www.aph.gov.au/DocumentStore.ashx?id=fc1aa59b-f4f7-408d-acfd-0f312793e59a&subId=459558>
 17. Richard Buckland, Supplementary Submission 56, Electoral Matters Committee, Inquiry into and report on all aspects of the conduct of the 2016 Federal Election and matters related thereto, 23 November 2016.
<https://www.aph.gov.au/DocumentStore.ashx?id=44eac9f0-f3fe-47cc-84e1-d34453285981&subId=459558>
 18. Australian Parliament House, "Report on the conduct of the 2016 federal election and matters related thereto", November 2018, Recommendation 3.
https://www.aph.gov.au/Parliamentary_Business/Committees/Joint/Electoral_Matters/2016Election/2016_election_report/section?id=committees%2freportjnt%2f024083%2f26083#s26083rec3
 19. Australian Parliament House, Speakers schedule of outstanding Government Responses to Committee reports, accessed 28 June 2021.
<https://www.aph.gov.au/SpeakersSchedule>
 20. Richard Buckland, "Is the new Senate vote capture system as risky as electronic voting?", The Conversation, July 19, 2016 1.00pm AEST.
<https://theconversation.com/is-the-new-senate-vote-capture-system-as-risky-as-electronic-voting-62436>
 21. Berj Chilingirian, et al, "Auditing Australian Senate Ballots", Cornell University, November 8, 2016.
<https://arxiv.org/abs/1610.00127>
 22. Andrew Conway, et al, "An analysis of New South Wales electronic vote counting", Cornell University, November 7, 2016.
<https://arxiv.org/pdf/1611.02015.pdf>
 23. Antony Green, "Preference Counting in Local Government Elections in NSW", NSW Electoral Matters Committee - Inquiry into Counting at NSW Local Government Elections, 10 October 2017.
<https://www.parliament.nsw.gov.au/ladocs/submissions/58838/Submission%2023%20-%20Mr%20Antony%20Green.pdf>
 24. Expert Study Mission Report, The Carter Center, Internet Voting Pilot: Norway's 2013 Parliamentary Elections, 19 March 2011
<https://www.cartercenter.org/resources/pdfs/peace/democracy/carter-center-norway-2013-study-mission-report2.pdf>
 25. Commissioner of Canada Elections, About Us, Accessed 10 July 2021.
<https://www.cef-cce.ca/content.asp?section=abo&document=index&lang=e>
 26. NSWEC, "Functional Requirements Specification for the Vote Count", 08 September 2016, Version: 3.4.

Security of Electronic and Paper Based Voting

Cyber Awareness Training for Election Staff using Constructive Alignment

Thomas Chanussot and Carsten Schürmann

¹ IFES

tchanussot@ifes.org

² Center for Information Security and Trust

IT University of Copenhagen

carsten@itu.dk

Abstract. Cybersecurity awareness and cyber hygiene trainings are becoming standard practice for employees of election administrations. Election Management Bodies (EMBs) have specific needs: elections are cyclic with regards to the tasks and their associated risks, they are high value targets during a short time window, and they suffer from high turnover of staff making sustainable training difficult. With lots of training methodologies and training programs targeting election observers, officials, etc., there are limited quantifiable measures for the efficiency of this type of training. Evaluating the adequacy of the training objectives and methodologies to the specific needs of election administration is becoming a necessity. We propose to use constructive alignment for designing and evaluating cybersecurity awareness trainings.

1 Introduction

Elections worldwide have become the battlefield between national and foreign actors and cyber-criminals on one side, and election management bodies and political parties on the other. The objectives of threat actors vary from financially motivated extortion schemes to destructive attacks as proxies to delegitimize the electoral process. This trend has increased in frequency, with an intensification of cyber activities towards the date of the actual election. Most incidents are triggered by human behavior [dbir20], foremost successful phishing attacks, with victims neglecting simple rules as the primary factor paving the way. It is now acknowledged that minimizing human errors through cybersecurity awareness training is paramount for achieving any reasonable level of cybersecurity in election administration. What is not so clear, however, is how to design and evaluate training programs that are specifically tailored for election officials.

As cybersecurity awareness and cyber hygiene trainings are becoming more readily available, it has also become standard practice for employees of election management and temporary staff hired for election administration to partake in such trainings. To this end, many organizations rely on third-party providers for delivering cyber-hygiene training through online platforms to large customer bases. Cybersecurity experts and consultancies do have the inside knowledge to

keep training materials up to date with respect to the latest threats. Overall, it is acknowledged that cybersecurity training has to be offered on a regular basis to ensure that the employees' cybersecurity understanding and knowledge is always up to date. Elections are cyclic events with regards to tasks and their associated risks, they are high value targets for a limited amount of time, they typically suffer from high turnover of staff making sustainable training difficult, and they are critical to preserve public confidence in the quality of the democratic process.

To assess the quality of cybersecurity awareness and cyber hygiene training and to measure their effectiveness in addressing these challenges, we develop in this paper a framework for evaluating existing and designing new courses based on the principle of constructive alignment [biggs03]. Cyber hygiene refers to the ability of election administrators (1) to understand that their respective behaviors, online and also in relation with other persons has a direct effect on the security and credibility of the electoral process, (2) to learn to identify and react to common threats and therefore minimize their respective damage, and (3) to be prepared to respond to not-yet identified and future cyber threats.

Inspired by the Structure of Observed Learning Outcome (SOLO) taxonomy [biggs1], we define four levels of understanding which encompass awareness, understanding of risks, comprehension of threats, and defense skills, empowering course participants to deal with future and yet unknown cyber threats. Our framework suggests that trainings should be assessed based on (1) the consistency of course prerequisites and how they accommodate participants with varying pre-existing knowledge, (2) the quality of the intended learning outcomes (ILOs) for each module and assess their consistency across several modules, (3) the intended retention policies, for each module, (4) the impact on the participants' cybersecurity behavior and understanding, and (5) expectation management.

When applying our framework, there are several insights to be gained how to design cybersecurity awareness training curricula. First, course participants usually work harder the more closely learning objectives are aligned with their assigned working tasks. Hence it is beneficial to distinguish between learning objectives that are general in nature and require additional mechanisms to capture the course participants' attention and those that are specific and targeted to a particular task. The latter can be aligned with particular threats anticipated by the EMB, and prepare the participant with the knowledge necessary to prevail in their professional role. Highly specialized curricula covering topics critical to running a credible election can be targeted to small selected audiences, the ILOs should be carefully aligned with the participants professional responsibilities. Curricula covering the basics of cyber hygiene training are usually designed for broader audiences with more diverse backgrounds and less consistent pre-existing knowledge and the ILOs need to be designed accordingly, especially if there are participants that have undergone similar trainings in preparation for earlier elections.

Second, online education can reach a much broader audience than facilitative-based teaching ever could. This is of particular interest for the electoral domain,

where many election officials need to be educated prior to an election. It is possible to require election officials to have passed cybersecurity training before becoming an election official, for example, by presenting a certificate of the successful completion of his/her cyber-hygiene course. Online cybersecurity training can also be organized in alternative modules suitable for different backgrounds. A total beginner module on social engineering, for example, would spend effort to explain the motivation of an attacker to the course participant, whereas a module for the advanced participants could go more in detail about the different techniques used by a social engineer attacker. Another example would be a module highlighting privacy, data security, and integrity as well as protection against disclosure of confidential data for operators working on voter registration activities. Accidental disclosure [**comelecleak**] of voter lists [**maltaleak**] are not uncommon and have been reported numerous times [**usaleak**].

In this paper, we propose an evaluation and design methodology for cybersecurity awareness training in Section ?? that is adapted to the specificities of electoral administrations. Using two existing training programs, as examples, we demonstrate how the methodology can be used to identify areas of improvement and to structure mature cybersecurity awareness trainings in Section ?. Finally, we assess results and conclude in Section ?.

2 Methodology

Cybersecurity awareness training has a bad reputation for being ineffective and boring. This, however, is not necessarily true, as demonstrated in prior work [**schurmann1**], which argues that if the training is tailored to the right audience with the right content, it is possible to measure the effectiveness of the training by relating pre-tests with post-tests. In this section, we push this point further and relate it to the theory of constructive alignment, which is a principle used for devising teaching and learning activities, and assessment tasks that directly address the intended learning outcomes (ILOs) in a way not typically achieved in traditional lectures, tutorial classes, and examinations [**biggs03**]. Constructive alignment applies to in-person training as well as online training and the literature is extensive [**Brabrand2008**, **Trigwell14**, **Walsh07**]. It is a modern teaching philosophy based on cognitive psychology that is increasingly used at universities to guarantee a pleasant and effective learning experience for each and every student. The central idea of constructive alignment is that the course content is organized in such way that enables the teacher to make a deliberate alignment between the planned learning activity and the ILOs. We believe that this observation is central for designing effective cybersecurity awareness training programs, and it provides guidance for evaluating existing programs.

Compared to higher-education, cybersecurity training for election officials and poll-workers brings along additional challenges, such as the body of course participants is usually extremely diverse with different academic backgrounds, different skills, and different expectations. In practice, this heterogeneity presents quite a challenge and the solution requires a well-thought out and principled

methodology that we structure according to five dimensions that we describe next.

2.1 Pre-existing knowledge

To guarantee the effectiveness of cybersecurity training requires the organizers of the training to control the heterogeneity of the course participants. If the backgrounds of the participants are too diverse, some will be bored while others struggle to keep up. Skilled facilitators and trainers can accommodate the curriculum depth to the individual participant's needs by making it more immediate and more closely relatable. In self-paced online training programs, on the other hand, the participant can adjust the pace, for example by quickly skimming through content he is already familiar with, but will not be able to change the depth of the content. Through effectively understanding the pre-existing knowledge of each student allows the grouping of students with similar baselines. Suitable content can then be tailored for each group, for example, by identifying and presenting relevant modules of the training program. The group can then proceed at a comfortable pace. This also takes care of the challenge of course participants already having taken earlier editions of the same course for past elections. The design of the training should therefore be modular, which means that it can be reorganized into a personalized learning experience that engages effective participation.

Thus, a *first dimension* to assess quality of cyber-hygiene awareness training on is to evaluate if pre-existing knowledge is collected and used to adapt the training experiences to individual needs.

2.2 The relevance and specificity of the learning objectives.

Intended learning objectives (ILOs) that accompany constructive alignment [biggs1] serve two goals. First, every course participant is given the opportunity to identify with the learning objectives before training commences, which makes learning effective and allows expectations between facilitator and participant to be aligned. Second, learning objectives define the structure of entire training program, they specify in a way, how one module builds upon another. In fact, ILOs guarantee a satisfactory progression with respect to the participants' level of understanding from awareness to skills [beyer1].

In Figure ??, we propose a simple taxonomy that distinguishes four levels of understanding, (1) *awareness*, which means the course participants are able to learn to identify and describe individual topics that are central to cyber-hygiene, (2) *understanding of risks*, which refers to participants being able identify cyber-risks rendering cyber-hygiene necessary, (3) *comprehension of threats*, which means that course participants should learn to understand intent and objectives of an adversary, and lastly (4) *defense skills*, which allows course participants to recognize cyberattacks, take counter measures, and adopt personal behavior to minimize the risk of cyberattacks specific to electoral operations, even those not discussed explicitly in the course.

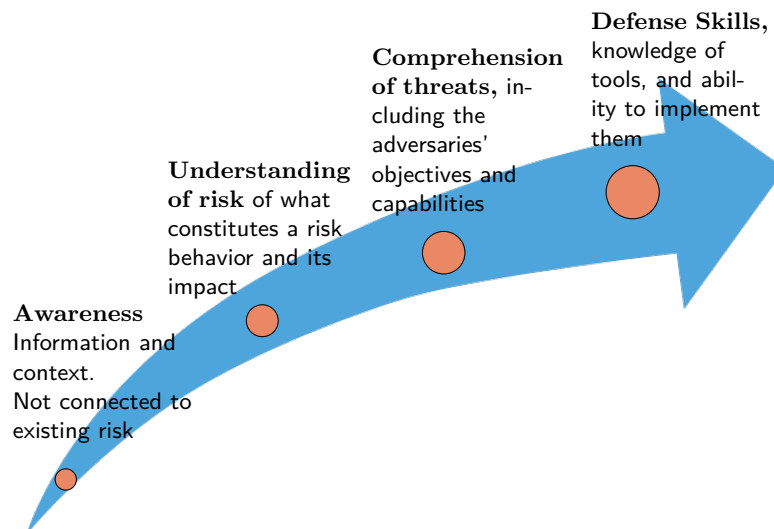


Fig. 1. Taxonomy

Thus, a *second dimension* to evaluate cybersecurity awareness training is to evaluate if the learning objectives for the individual modules are clearly stated, consistent with respect to the level of understanding, and aligned with the needs of the participants with regards to their profession/role.

2.3 Retention Period

One of the challenges of cybersecurity awareness training is that topics can vary significantly in abstraction and relevance. Some threats and good practices seem intuitive, while others seem remote and unlikely. To cope with this enormous spread between the concrete and the abstract, each module should define an expected retention period that presupposes for how long the knowledge gained through the training should be actively applicable for the course participant.

Being explicit about the retention period has several advantages, including what topics and which materials can and should be covered in a module, and what mechanisms should be used to guarantee retention. When the expected retention period is shorter, emphasis should be given to materials that are immediately relevant to the participants learning experience. The very fact that interesting and relevant topics are covered is then usually enough to motivate the participants to perform well [schurmann1]. Longer expected retention periods will allow to engage with more general knowledge and good practices, which means that participants may require additional incentives, such as practical simulations, additional tests, etc. to ensure that the longer retention period can be guaranteed.

Thus, the *third dimension* to evaluate cybersecurity awareness training along is to verify that the expectations regarding retention have been set and made explicit, that the course content is compatible with the prescribed retention periods, and that the choice of mechanisms to boost retention are in line with the overall module design. Ideally, one should also conduct user studies with the course participants, to collect statistical evidence indicating the training was effective.

2.4 Measuring Behavioral Change

Establishing good cyber-hygiene requires not only awareness and knowledge, but users most likely will also have to adapt their behavior and adjust their attitudes towards technology and security. The challenge is that new practices learned during a training are often difficult for participants to retain for extended periods of time, as participants often tend to relapse and revert to insecure practices with time if they ever adopted secure ones in the first place, because “it’s just easier”. The goal of cyber-hygiene trainings is to provide knowledge, tools, and incentive to adopt sustainable secure practices. Changes in the users’ attitude are difficult to measure. In a controlled corporate environment, it might be possible to use technology to measure behavioral change, by tracking the number of successful phishing attacks, for example. But in the larger context, one must rely on self-reported questionnaires to understand whether knowledge translated into improved behavior. These questionnaires, in which respondents select a response by themselves without interference, are inherently biased. Acknowledging this limitation, mechanisms to measure the adoption of a limited and pre-defined set of safe behavior is a *fourth dimension* through which the effectiveness of cyber-hygiene training should be evaluated.

2.5 Expectations Management

When election officials and poll workers are asked to participate in cybersecurity awareness training, there might be different expectations in play, which need to be considered. Poll-workers and election official usually are assigned different tasks, and serve in different roles. This requires that the training program can cater to different needs, and be adaptive to different expectations. It is prudent, that the expectations of the participants are properly managed, which leads to the *fifth and final dimension* of our evaluation methodology, which is to what extent does the training program provide mechanisms to identify and integrate different expectations?

In summary, we propose an evaluation methodology for cyber hygiene training based on five dimensions, (1) the consistency of the course prerequisites and how they accommodate participants with varying pre-existing knowledge (2) the quality of the ILOs for each module and assessing their consistency across several modules, (3) the intended knowledge retention policies for each module, (4) the impact on the participants’ cybersecurity behavior and understanding, and (5) expectation management. This methodology can also be used as a guide

when developing new modules and courses, or when restructuring existing cyber-hygiene training into either several courses, or one course with several modules, so that returning participants will find a tailored and exciting curriculum to partake in.

3 Practical application: evaluating cyber-hygiene trainings

To test our evaluation and design framework, we have applied our methodology to two training programs that have been provided to electoral administrations: IFES cybersecurity awareness training for EMBs (Regional Election Administration and Political Process Strengthening – REAPPS) and Cyber-hygiene for the Danish Election administration.

3.1 The IFES Cybersecurity Awareness Training

IFES' cybersecurity awareness training was developed in late 2018, it has been conducted with several hundreds of officials in Eastern Europe and Balkan States.

Pre-existing knowledge. The course was developed for election staff with little to no exposure to cybersecurity. It does not test or categorize participants according to their pre-existing knowledge but aims to offer general awareness and an introduction to cybersecurity concepts as they apply to the electoral context. This means that no mechanism is provided to identify participants who have already taken this course in the past. As this training was specifically designed for participants with no prior exposure to cybersecurity concepts, this criteria does not directly apply, but would need to be integrated as the training reaches an audience with more mature cybersecurity awareness and skills.

The relevance and specificity of the ILOs. The content of the training can be classified in modules with different expected levels of understanding, as indicated in the table below. This classification of the topics highlights the priorities set forth during the training: identifying phishing and measures to protect accounts (passwords and multi-factor authentication in particular). These ILOs are aligned with global threats faced by EMBs.

Awareness	Understanding	Comprehension	Defense
Software patching; Antivirus tools; End-point protection; Backup strategies; BYOD; Public WiFi risks; USB security; Social media	Global Threats; Cyber-attacks	Election specific threats; Multi-factor authentication	Passwords management Detecting phishing red-flags

Retention period. This training does not explicitly state an expected retention period. It has usually been conducted as an introductory course disconnected

from the electoral cycle, or ahead of an election operation, for which the training is well suited as the defensive skills are aligned with types of attacks election staff must be ready to detect and respond to during the election period. The expected retention period can be derived from the mapping of the ILOs: *password management* and *phishing detection* are expected to be of immediate use to the participants, for these topics, the training goes in more depth and deliver practical and engaging learning exercises.

Measuring behavioral change. Pre-tests and post-tests measure the retention and understanding of the course material, improvement on the behaviors and practices of the participants is not measured over time. Reminders are sent to the participants on a regular basis and are used to reinforce key messages and good practices learned during the course.

Expectation management. The course was designed to provide an introductory course to cyber hygiene and cybersecurity awareness. It is not specifically tailored to a particular group of participants and to specific classes of risks. It does align with participants expectations who had low previous exposure to cybersecurity concepts.

Overall evaluation: IFES' cybersecurity awareness training has been developed with a dual purpose as clearly visible from the table of ILOs. It offers a generic, low level awareness of threats, risks and security good practices on the one hand, and more advanced ILOs with practical defense skills for threats that are considered global and highest risk. This clear distinction could provide a roadmap for future trainings, as several topics could be elevated from awareness to practical defense skills based on the risk environment, period of the electoral cycle, and specific needs of the EMB related to the adoption of new technologies for example.

3.2 Training Denmark's Digital Election Secretaries

Moving to Denmark, the cybersecurity training for digital election secretaries [schurmann1] was organized in a principled fashion follow a particular methodology for training design. By catering to a very narrow and homogeneous target group, the content of the training was defined by the role that the participants play during the election: the digital election secretaries were responsible for the voter registration technology deployed in polling stations. In the case of a cyberattack, it is the digital election secretary who decides to abandon the use of technology and move to the paper backup system. Attack trees were used to explore the threat space and then course modules were derived in response to the overall ILO that the participants should be able to recognize threats and to defend against them. The resulting course consists of three modules, (1) an introductory module to create a joint level of understanding among the participants, (2) an introduction to man-in-the middle attacks against election technologies, and (3) a module dedicated to spotting and mitigating social engineering attacks. All three modules were tailored to the particular role of the course participants.

Pre-existing knowledge. This course was organized as a pilot study that does not make any assumptions about pre-existing knowledge of the course partic-

ipants although all participants are public servants who work for the city of Copenhagen, and most like had already been exposed to cybersecurity awareness training. As this was the first time the course was offered, no attention was paid to the fact that some of the course participants might have taken this training in the past.

The relevance and specificity of the ILOs. No ILOs were explicitly mentioned in the course descriptions, but because the modules are directly derived from the role that the participants play during the election, they can be easily inferred. The training materials can be organized into the four classes of understanding. As this training is targeted to the practitioners, it aim to achieve a high level of understanding when addressing the ability of the digital election secretary to react efficiently in the case an attack is noticed.

Awareness	Understanding	Comprehension	Defense
BYOD; Ransomware; Disinformation	Election specific threats.	Man-in-the-middle, USB security. Key logger. Computer virus. Theft. Social Engineering: Impersonation; Social Engineering: Authority; Urgency; Trust; Shoulder-surfing; Spear-phishing.	Fall-back procedures. Immediate response. Escalation. Proactive security. Securing Evidence.

Retention period. Although the training does not explicitly state an expected retention period, it is clear that the course is designed to be taken a few days before election day activities commence and it is expected that the participants will remember the content until after the election. Since the course is offered online, it is easily possible to retake the course in preparation for another election.

Measuring behavioral change. This training requires the participants to take two tests, one before the training commences, and the other right after. These tests are organized to measure if new knowledge was acquired while taken the training or not. Improvement on the behaviors and practices of the participants is not measured over time.

Expectation management. Since the course participants will play a similar role in the election, their expectations are closely aligned, and also their interest is heightened, because it is considered important to take such a training in preparation for assuming the role.

Overall evaluation: The Danish course for training digital election secretaries has clear ILOs on a rather high level of understanding. The course will only work for its intended audience, and should not be mistaken for a general-purpose cybersecurity awareness training. The course could be improved by taking into account the pre-existing knowledge of the participants, possibly by integrating the course into a larger election official training program, where assumptions about pre-existing knowledge and prerequisites has been made explicit.

4 Conclusions

While cybersecurity awareness trainings are becoming standard practice for EMBs, there has been little study of how they answer the specific needs and unique threats during the election cycle. As election operations are increasingly digitalized (with a steep increase following the 2020 Covid pandemic), EMBs are adapting their stance with regards to cybersecurity.

Election administrations are unique in their threat model, with different risks inherent to different activities during different phases of the electoral cycle, and they are also unique in terms of providing verifiable results. As elections are becoming increasingly a battleground for cyber-attacks and disinformation campaigns, EMBs must rely on effective training methodologies with well-defined intended learning objectives (ILOs), and move rapidly to mature training and education programs that are consistently organized and well-defined along the dimensions that we have presented in this paper. We believe that evaluating cybersecurity awareness programs this way can not only ensure the adequation of the training content with the specific needs of election administration, it can also help trainers develop mature training programs in which content and objectives are quickly adjusted in terms of content and depth (from awareness to defensive skills). We believe that the evaluation strategy presented here applies to most elections and electoral systems, in developed as well as developing and post-conflict countries.

We conclude that *trainings need to incorporate the participants' profiles and their respective backgrounds* and identify and respond to their specific needs and risks. This will be a requirement for trainings targeting users with prior knowledge and exposure to cybersecurity issues.

Furthermore, *evolving learning objectives that range from awareness and information to higher levels of skills and know-how* will become key to maintain sustained users' engagement on cybersecurity issues. Cybersecurity evolves rapidly, new threats emerge, such as supply chain attacks, and new tools are being developed to mitigate these new risks. Cyber-hygiene is a long-term engagement and trainings should be conducted continuously upon entering new phases of the election cycle. Learning objectives should evolve and be refined to maintain ongoing engagement for users who have already received a training in the past.

In the case of heterogeneous groups of participants and roles, *questionnaire and tests prior to the training should influence the curriculum*. Collecting the necessary information regarding pre-existing knowledge of the participants could be done via a pre-training questionnaire using behavioral questions (what can participants do wrong on scenario-based questions). Knowledge based questions seem to remain the best method to measure success of the training.

Cyber-hygiene courses in general need to be conducted periodically to re-engage users and update them on the latest threats and techniques to mitigate cybersecurity risks. We believe that *the frequency of trainings should be aligned with the expected retention period and the electoral cycle*. Good practice in many industries put the periodicity of re-engaging users with cyber-hygiene practices

around one year. However, election administration is cyclic, and subject to different types of threats. To increase the efficiency of the cyber-hygiene training and establish cybersecurity as a strategic objective of secure election preparation, planning of training periods should be based on the election cycle. Reminders in the forms of newsletter, posters, calendars are very important and should be ongoing, they support the training but do not provide new information.

Election officials are often under the pressure of an incoming electoral operation, without strong involvement of the management, cyber-hygiene training often receives little attention. A strong management support is a pre-requisite to any successful cyber-hygiene training. Therefore, *cyber-hygiene should be part of an overall security strategy*. Furthermore, a culture of trust is needed to ensure that election officials receive proper training and support rather than be blamed for cybersecurity incidents. Each EMB faces different threats, has a different risk acceptance level, and cybersecurity maturity, EMBs need to understand and formalize their needs and determine clear training course objectives.

Academic study and international good practice have also demonstrated that awareness and trainings can only go so far if they are not backed up by organizational policies. Cyber-hygiene cannot happen in a vacuum, election management administration need to *ensure that the training aligns with the cybersecurity objectives* and that recommendations are backed back appropriate administrative controls. The way to manage cyber risks due to the human factor is by high quality trainings and these are best designed and analyzed following the theory of constructive alignment.

And Paper-Based is Better? Towards Comparability of Classic and Cryptographic Voting Schemes*

Marc Nemes², Rebecca Schwerdt¹, Dirk Achenbach², Bernhard Löwe, Jörn
Müller-Quade¹

¹ Karlsruhe Institute of Technology

² FZI Research Center for Information Technology

In today's real-world elections the choice of the voting scheme is often more subject to tradition than the result of an objective selection process. As a consequence, it is left to intuition whether the chosen scheme satisfies desired security properties. Employing a scientific selection process to decide on a specific voting scheme is currently infeasibly cumbersome. Even those few schemes which have been thoroughly analyzed do not provide easily comparable analysis results. Hence there is a strong need to increase meaningful comparability, allowing democracies to choose the voting scheme that is best suited for their setting.

Contrary to common conception there is not *the* classic paper-based voting scheme. We highlight examples of significant differences between systems in key areas. Possibilities for authentication are identification via ID-card (mandatory^{3,4} or by request only^{5,6,7}) or handing in specially issued voting documents^{5,6} which are compared to the eligible voters lists. Ballots are often marked with a pen by crosses^{3,6,8} but in order to prevent recognizably marked ballots, ballots with additional markings are often declared invalid.^{6,7,8,9} Alternatively, choices may be indicated by handing in preprinted voting ballots which only contain the selected choice and do not have to be marked at all.¹⁰ On the other hand, there are countries where write-in candidates are allowed or ballots with individual markings are still valid.³ Sometimes everyone is allowed to watch the whole voting process in any polling station,^{3,5,8} giving voters some form of verifiability. In other cases only eligible voters are admitted into the polling station and they have to leave again as soon as they are done casting their vote.^{6,7} In these systems usually only very specific observers are allowed—like representatives of the parties nominated for election.^{3,4,7,9,10} Mostly, ballot boxes are opened and tallied directly within the individual polling stations^{3,4,5,6,7,9} with partial results

* The full version of this paper is available at <https://eprint.iacr.org/2021/1122.pdf>

³ Elections Commission of the Maldives: The Presidential Election Regulation

⁴ India: The Representation of the People Act, Act No. 43

⁵ German Federal Election Regulations, BGBl. I S. 1769 (ber. 258), 1328, 1329

⁶ Folketing (Parliamentary) Elections Act of Denmark, Consolidated Act No. 369

⁷ The Constitution of the Republic of Ghana (Amendment) Act, 526th Act

⁸ German Federal Election Law, BGBl. I S. 1288, 1594, 2395

⁹ UK: Representation of the People Act (Consolidated Version), Schedule 1

¹⁰ Israel: Knesset Elections Law (Consolidated Version), No. 40 (5729-1969)

being announced there.^{3,5,6,7} This exchanges the need to transport ballot boxes in a secure way for the need to securely communicate partial results. Other systems tally in one or multiple central locations⁶ or explicitly require ballots from one ballot box to be mixed with those of others.⁹ Note that these are merely illustrating examples. A detailed comparison of individual schemes would show even more differences—not only in the processes, but also the resulting properties. We do not claim this to be a drawback of the current voting landscape. The scientific and political voting community should, however, be aware how heterogeneous the landscape of classic paper-based voting system is and refrain from comparisons to just “classic paper-based voting”.

This heterogeneous field of paper-based voting systems does not preclude meaningful comparisons, as long as the exact system something is compared to is specified. Unfortunately, comparing another voting system to current solutions is impeded by several other factors. Some building blocks are used with clear and known (security) intentions: a list of voters is essential to allow at most *one vote per eligible voter*, the voting booth supports the *privacy of the ballot* and the ballot box helps that *ballots cannot be removed or linked to the voter*. Unfortunately, this level of reasoning is commonly the extent of any “security analysis” of classic paper-based voting schemes. The first problem we encounter is lack of specification. Even though election legislation is often very complex and detailed, most countries lack precise (publicly known) instructions. If we do not fully know a scheme, we can not analyse it. In addition, knowing and understanding every aspect of such a protocol not only exceeds the comprehension of numerous voters but is not possible at all if not every detail of the process is specified and publicly known. Secondly, regarding properties, constitutions and international agreements on voting rights give only broad goals like freedom of voting choice or secrecy and equality of votes—but leave it to courts to decide whether they are fulfilled or not. As long as clear definitions are lacking, it is impossible to rigorously analyze whether a voting scheme satisfies the required properties. Thirdly, we find a lack of security analyses and even security intentions. Most schemes were developed a long time ago and modified often, making it unnecessarily infeasible to get a comprehensive documentation of the reasoning behind some design decisions. While some fundamental ideas of the protocols are clear, legislation on the voting process only states what is to be done without any explanation on why certain provisions are taken, let alone what they actually achieve. Not knowing the reason and consequences of design decisions does not only encumber security analyses but allows for much easier coercion as voters are less sure about the information an adversary may plausibly obtain.

We have seen that meaningful comparison between newly proposed voting schemes and classical paper-based voting is currently infeasible because of the largely ignored fact that the field of currently employed paper-based solutions is very heterogeneous. More importantly we found that, as of yet, classical paper-based schemes lack the necessary rigorous and thorough specification as well as formal security and privacy analysis to serve as a basis for comparison. We think it is an important societal goal to close these gaps.

Improving the Accuracy of Ballot Scanners Using Supervised Learning

Sameer Barretto William Chown David Meyer
Aditya Soni Atreya Tata J. Alex Halderman

University of Michigan

{sambar, chownwil, davidmey, adisoni, artata, jhalderm}@umich.edu

Abstract. Most U.S. voters cast hand-marked paper ballots that are counted by optical scanners. Deployed ballot scanners typically utilize simplistic mark-detection methods, based on comparing the measured intensity of target areas to preset thresholds, but this technique is known to sometimes misread “marginal” marks that deviate from ballot instructions. We investigate the feasibility of improving scanner accuracy using supervised learning. We train a convolutional neural network to classify various styles of marks extracted from a large corpus of voted ballots. This approach achieves higher accuracy than a naive intensity threshold while requiring far fewer ballots to undergo manual adjudication. It is robust to imperfect feature extraction, as may be experienced in ballots that lack timing marks, and efficient enough to be performed in real time using contemporary central-count scanner hardware.

1 Introduction

Hand-marked paper ballots counted by optical scanners are the most popular voting method in the United States, used by jurisdictions home to about 70% of registered voters [29], and they are becoming even more prominent due to the rapid expansion of postal voting spurred by the COVID-19 pandemic [13]. Despite its importance, optical scan voting faces two significant integrity challenges. First, deployed scanners suffer from a host of well-documented vulnerabilities (e.g., [11, 14, 2, 15, 18]). Second, and the focus of this study, even in the absence of an attack, traditional scanning techniques sometimes fail to accurately count some voter marks [12]. In principle, risk-limiting audits can address both problems by ensuring that any fraud or error sufficient to change the outcome of a contest is likely to be detected [24, 17], but widespread adoption of RLAs, even for Federal contests, may be a decade or more in the future. Given that many major contests will not be subject to rigorous audits anytime soon, it is important to ensure that scanners themselves count ballots as accurately as practically possible.

Today’s ballot scanners typically employ variations of a relatively simplistic technique [12, 27]. After creating a digital image of the ballot, they identify the voting targets and calculate the average shading within each target area, s_i . For a predefined threshold α , target i is treated as marked whenever $s_i \geq \alpha$. Some modern scanners make use of a second threshold, β . If $\beta \leq s_i < \alpha$, the target is treated as an ambiguous or *marginal* mark, and the ballot is set aside for officials to manually determine the voter’s intent, in a process known as *adjudication*.



Fig. 1: Voted targets from Humboldt (*top*) and Pueblo (*bottom*) datasets. These scans originate from Hart InterCivic and Dominion scanners, respectively. This difference is reflected in the style of the targets and the quality of the scans.

This technique performs well on ballots that have been properly marked, but it sometimes falls short when handling ballots where the voter has not followed the instructions precisely [12], as in many of the samples in Figure 1. Often, voters disregard ballot instructions and use other marks such as X-marks or check marks to indicate their intent. As discussed in Section 3.1, we found that roughly 8.5% of marks in one large corpus of voted ballots were not filled as directed. While humans can easily identify these “marginal” marks and typically interpret them correctly, they may be challenging for current optical scanning systems to process accurately. If marks are not dark enough, they may not meet either threshold will therefore be ignored by current systems. Even in the case where marks fall within the adjudication range, tabulating them imposes increased labor costs for resource-constrained voting jurisdictions.

We investigate the feasibility of improving scanner accuracy and reducing adjudication costs by applying supervised learning techniques. Using real voted ballots, we train a convolutional neural network to classify a variety of mark styles, including both properly marked targets and common marginal marks. Compared to a generic implementation of mark recognition based on intensity thresholds, our model achieves more accurate classification and lower rates of adjudication. We further validate our technique using a second real-world ballot corpus for which we have the results of scanning and adjudication reported in the election, and achieve identical results in every case. These findings suggest that our approach could improve scanner accuracy while reducing election costs.

2 Related Work

The challenging nature of ballot mark recognition has long been recognized and is discussed at length by Jones [12] and Toledo et al. [27].

A number of previous studies have investigated methods for improving ballot scanning. Several groups have approached the problem by combining computer vision for feature extraction with human judgement for checking the interpretation of marks. In 2010, Cordero et al. proposed a method for efficiently verifying the scanner’s mark interpretations by having humans review batches of ballot images automatically superimposed on each other [6]. Wang et al. later developed OpenCount, a system that similarly automated feature extraction and provided

interactive tools for classifying voter marks [30]. Although our goal is to improve automatic mark recognition and reduce reliance on operator input, these earlier works could complement our techniques and result in further efficiency gains, if applied to the ballots that our approach determines require manual adjudication.

More closely related to our approach, other prior work has applied supervised learning to mark recognition. In 2009, Xiu et al. briefly investigated a classification approach generally similar to ours, but based on modified quadratic discriminant functions (MQDFs) instead of convolutional neural networks (CNNs) [31]. Although they reported strong performance, their dataset consisted of only a few hundred ballots, making comparisons with real-world scanner performance difficult. A 2015 NIST study further benchmarked several ML-based approaches for categorizing marginal marks [1], but their primary goal was to improve testing of optical scanners rather than to surpass intensity-based mark detection.

3 Methods

In recent years, convolutional neural networks (CNNs) have become the industry standard for image classification [26]. CNNs use a divide and conquer strategy to classify images, attempting to gain a localized understanding of an image’s structure to identify key characteristics which are then used to classify the image as a whole. For instance, in classifying marks on a ballot, one feature a CNN might identify is lines at a 45-degree angle, corresponding to X-marks. We chose to use a two-dimensional CNN, since it allows for the detection of multidimensional structures, in contrast to a one-dimensional fully-connected network which would immediately flatten the image, losing the ability for the network to extract this type of structural feature from the data. Another advantage of CNNs is that they use comparatively fewer parameters than fully connected networks, since they reuse their parameters several times. This means that the model is easier to train because it requires less data to achieve a higher accuracy and takes less time.

We developed our own CNN model and then tested it on ballot scans collected from actual elections, evaluating its performance relative to a simple threshold-based approach. It was not possible to obtain a currently marketed optical scanner to use as a baseline for comparison, so we wrote our own implementation closely modeled on the Dominion ImageCast scanner system, as described in patents and court documents [22, 7]. The Dominion system, which is used in parts of 28 states [29], defaults to $\alpha = 35\%$ and $\beta = 12\%$, which we adopted for our implementation. One advantage of using this baseline model rather than an actual optical scanner was that both models used the same extracted features, allowing for a truer comparison of their mark detection methods.

We decided to build a model that would classify individual targets as features rather than examining entire pages. This way our model generalizes well across different contests and page types, so long as the targets are the same shape and size. Since the two datasets we used (described below) had differently shaped targets, we used a separate model for each. Both models used the same CNN architecture, but each was trained on different data.

3.1 Data

The ballot scans we used came from two datasets: the November 2009 election in Humboldt County, California and the November 2020 election in Pueblo County, Colorado [23]. Initially, we used a representative subset of the Humboldt data, consisting of 23,846 out of the 28,383 non-blank pages, which contained 149,394 voting targets. Later, to validate our approach, we used a subset of the Pueblo dataset, which provided ballot scans as well as the official interpretation of each target resulting from the real scanners and adjudication process. This allowed us to directly compare the CNN model’s output to real election practice. From the 89,098 Pueblo County ballots, we used a representative subset of 1,719 ballots that contained 147,121 voting targets. Each ballot consisted of multiple contests. Some Humboldt contests allowed for only one vote while others allowed multiple choices to be selected. Additionally, the ballots in both datasets did not have a straight-ticket option, so most contests contained marked targets.

Labeling To provide ground truth for the Humboldt data, we manually labeled all of the targets in our subset. We started by labeling each ballot page type; for purposes of this study, a page type is defined as a set of scans that contain the same contests in the same relative locations on each page. We then labeled the individual targets in two passes, according to two labeling schemes. In the first pass, we labeled targets by the mark type, and in the second, by perceived voter intent (0 for no vote, 1 for vote). The first schema is presented in Figure 2, along with a summary of the first pass of labeling. Approximately 69% of targets were unmarked, 29% were properly marked, and 2.7% contained a marginal mark.

We verified our labels by comparing the election results published by Humboldt County [10]. There was near perfect agreement for contests that had been labeled completely, with the maximum difference being 15 out of 6529 votes (or 0.2%). Most contests were either in complete agreement or differed by only 1 or 2 votes. In all the contests where there was a mismatch, our vote totals were less than the official counts. Upon investigation, most of the discrepancies were due to malformed or flipped scans, which we did not label. The small residual disagreement could be due to inaccuracies in the original count or our own human error.

Unlike the Humboldt scans, which were stored as grayscale images, the Pueblo scans were 1-bit black and white. There may have been some faint marginal marks on ballots that were undetected by the optical scanners and were also missed by our manual labeling. In this case, all models would have misclassified this type of mark, since it was lost at the scanning stage rather than the interpretation stage.

Feature Extraction The ballots in the Humboldt dataset lack timing marks, and we found that the position and orientation of the ballot relative to the scanned image was inconsistent across scans. To overcome this, we created a template for each page type that indicated the location of each voting target relative to the top-left corner of a rectangular printed border that surrounds the ballot. We used OpenCV’s contour detection algorithm [3] to obtain the coordinates of the corners of the border in each scan, then aligned the appropriate

Mark Type	Count
No Mark	100,999
Properly Marked	42,165
Marginal Mark:	
X-Marked	1,115
Check-Marked	93
Lightly Marked	1,903
Partially Marked	489
Marked and Crossed Out	316
Bad Scan / Wrong contest	2,242
Other	72
Total	149,394

Fig. 2: Number of marks of different types in Humboldt dataset, as determined by manual classification. 8.5% of the marks in this dataset were marginal marks.

template to extract all of the voting targets. This method accounted for the common case of vertical and horizontal shifts of the ballot within the scans. However, this method is not able to account for other kinds of scanning artifacts, including ballots with nonlinear distortions due to misfeeding.

For the Pueblo dataset, the ballots contained timing marks, which provided four points of reference for each individual target, giving us an extremely accurate position for extraction. For each page type, we used OpenCV to identify the timing marks corresponding to each target and used them to extract the target regions. This was highly resilient to rotations and other scanner distortions.

Partitioning Into Training and Test Data We used a subset of the labeled targets from each dataset for training and the remainder for testing. Of labeled targets from the Humboldt dataset, 54% (corresponding to 12 out of 17 page types) were used for training. For the Pueblo dataset, 75% were used for training. These differing splits were a matter of convenience. Both models exhibited excellent performance, but we note that the larger amount of training data may have benefited the performance of the Pueblo model relative to the Humboldt model.

3.2 Baseline Model

We sought to compare our methods to the commonly used intensity-threshold technique. Since we did not have access to a deployed ballot scanner, we created our own implementation modeled after the Dominion system described in Section 2. For each ballot, our baseline model considers all the extracted targets in a given contest and predicts each target as either no vote, vote, or adjudicate. In practice, a single adjudicated mark will result in the entire contest on that ballot being reviewed by humans, so if any mark was predicted as adjudicate, we labeled all the targets in that contest the same way.

Dominion’s scanners create 1-bit-per-pixel bitmaps, as shown in Figure 1. In order to replicate this behavior using the grayscale Humboldt scans, we applied

Floyd–Steinberg dithering (a common graphics algorithm provided by the imaging library we used [4]) to reduce the grayscale images to black and white while approximately maintaining the average intensity within local regions.

The next step was to calculate the number of marked pixels inside the target area. However, each feature consisted of not only the voter’s mark (inside the target), but also the pre-printed target border and the area immediately outside it. To account for this, we first converted our thresholds into raw pixel counts, leveraging the fact that all targets had the same dimensions. Then we subtracted the average number of black pixels occupied by the unmarked target border.

To allow for imperfect feature extraction, our baseline implementation considered a target area that is somewhat larger than the printed targets. Some fielded scanners are known to do so as well, but to our knowledge the specifics of this behavior are not well documented by any manufacturer. We note this as a limitation of our baseline model. It is possible that real scanners differ in such aspects and so would sometimes produce different results; however, we expect variations based on marks outside the printed target to be uncommon. In our datasets, such marks rarely occurred except in cases where the shading within the printed target alone would have clearly been an intended mark.

3.3 CNN Model

Preprocessing Before we could train our model, we needed to transform our dataset. In order to decrease computational costs, we resized the cropped target areas to 28×28 pixels with 8-bits-per-pixel of depth. We then stored them in a three-dimensional array, X , parallel to their associated labels, y . Finally, we normalized the pixel values in X to a 0–1 scale.

Our manual classification rubric included “lightly,” “partially,” and “properly” marked labels, but we later realized that the distinction between these classes varied depending on who was assigning the label. Due to the subjectivity, we merged these classes prior to training. All three labels indicated that the voter intended a mark; we reviewed the entire contest when making these classifications, and in each case the voter’s intent was clear.

Finally, we made a second partition of the targets from those that were set aside for training, reserving 85% for training and a standard 15% for validation. This allowed us to train our model using various parameter combinations and determine which were best by examining performance on the validation set. We followed this process for both datasets independently.

Model Structure The model we chose consisted of a single convolutional layer with 25 filters of kernel size 3×3 , stride 1, and no padding. The output was passed through a ReLU nonlinearity, followed by a fully connected layer with ReLU, and finally a second fully connected layer that culminated in seven neurons. We used the softmax function to create a probability distribution from the final layer weights and outputted our prediction as the class with the highest probability.

A primary consideration while designing the model was the number of convolutional layers. Models today can have upwards of 50 layers [9], but excess layers can

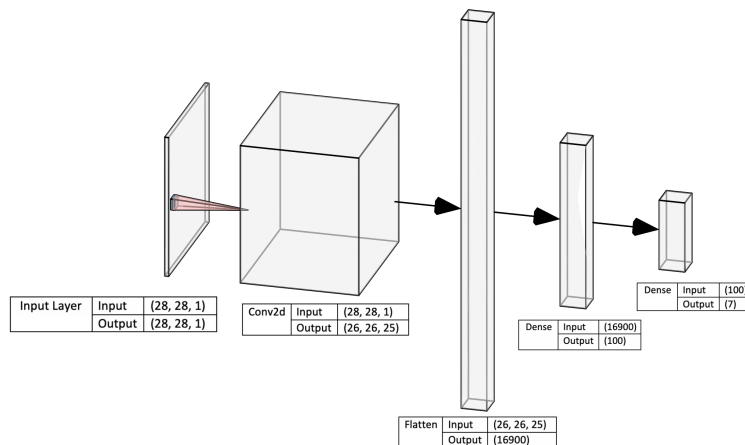


Fig. 3: The CNN architecture we used. Pictured layers appear from left to right in the order they were applied. (*Image generated using [16].*)

cause overfitting. Our dataset was relatively uncomplicated, with X-marks, check marks, and marked and crossed-out marks being the most complicated structures. We wanted a model capable of learning these structures but also general enough to categorize all X-marks, regardless of their shape, size or orientation, as an X-mark. We initially made the assumption that more layers would result in higher accuracy, but in evaluating our model, we noticed that our training loss was significantly lower than testing loss, and our validation accuracy was low, which suggested that the design was overfitting. This led us to use a shallower model, reduced to one convolutional layer and with an increased number of convolutional filters. We observed that this approach reduced overfitting and significantly increased validation accuracy.

Before trying a shallower model, we experimented with hyperparameter tuning, as well as regularization methods such as dropout. We also attempted to add a pooling layer, to downsample, and to reduce the number of parameters, but found that these features were unnecessary due to the already low spatial size of our images. The shallower model we settled on also had the side benefit of faster training, allowing more iteration in our model development process.

We implemented our model using Keras and TensorFlow. We were able to take advantage of the built-in convolutional and fully-connected layers while having the flexibility to write our own evaluation metrics.

Hyperparameter Selection One important hyperparameter was the evaluation metric. Our ultimate goal is to produce a vote tally that comes as close as possible to the collective will of the voters, and our model also should be intelligible to voters, allowing people to understand how their votes are counted. With these criteria in mind, accuracy is the most logical evaluation metric. For training the model, however, simply trying to optimize for accuracy has its drawbacks.

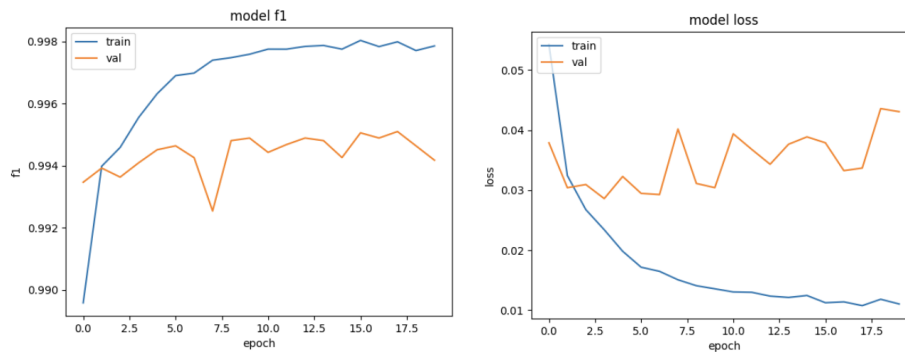


Fig. 4: Using 17 epochs optimizes validation F1 score while retaining low loss.

Since marginal marks account for such a small percentage of the data relative to properly marked and blank marks, a model trained for accuracy would not learn to classify these marks as well as their more prevalent counterparts. To address this, we chose to use a model that optimizes F1 score, the harmonic mean of precision and recall, which puts more weight on correctly classifying these marginal marks. By optimizing for F1 score, we were able to produce a model that had a higher overall accuracy compared to one that optimized for accuracy directly.

The other traditional hyperparameters we selected were batch size and the number of epochs. Based on a number of trial runs, we expect that a fairly wide range of batch sizes would be appropriate; we chose 32. For the number of epochs, we chose 17, which testing determined was past the point of diminishing marginal returns for the F1 score while maintaining low loss, as shown in Figure 4.

The final important hyperparameter was a threshold for confidence, which we used to apply our trained model to an entire contest rather than individual targets. That is, how confident did we need to be that all the targets in a contest were classified correctly in order to not designate that ballot for adjudication? To utilize this threshold, we first obtained the product of the label probabilities for each of those targets, and then compared that value to the threshold. Similarly to the baseline model, if this value was lower than the threshold then we would send the entire contest for adjudication. We tested several threshold values and obtained the best results with a threshold of 0.95 combined with adjudicating any contest in which the classifier found three or more different types of marks.

3.4 Differences for Pueblo Dataset

Although the model structure for the Pueblo dataset was broadly similar to the Humboldt model, we did not use the baseline model to evaluate it since we had the scanner’s actual interpretation as ground truth. Each ballot in the dataset included the officially counted votes (and the results of adjudication, if applicable) as a final page in the scan, a feature that Dominion calls AuditMark [8]. We extracted these results using the Pytesseract optical character recognition library.

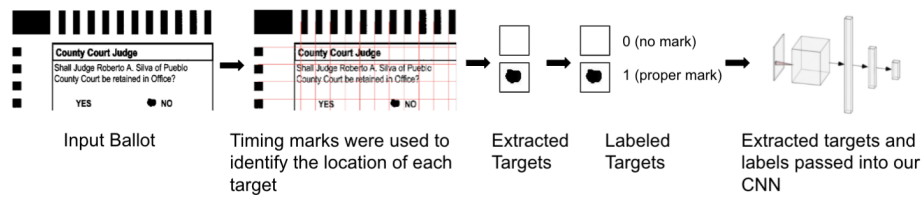


Fig. 5: For Pueblo ballots, we used timing marks to extract targets, manually labeled them, and passed these features to our CNN model.

Through manual and automated inspections of the Pueblo dataset, we established that it contains far fewer marginal marks than the Humboldt data. This may be due in part to Pueblo County acting to protect voter privacy by removing ballots with unusual styles of marks that were flagged for adjudication. For this reason, we used the Pueblo dataset to test how a CNN model would perform compared to current scanning systems under “ideal” ballot conditions—i.e., post adjudication, limited marginal marks, and clear ballot instructions. Our goal was to establish whether a CNN-based system would perform as well as current systems even under the circumstances where current systems are most accurate.

4 Evaluation and Results

To compare ballot scanning models, there are three distinct metrics to consider: classification accuracy, number of ballots that require adjudication, and computational cost. First, it is important that a model is as accurate as possible because it is vital that the tabulated results match the intent of the voters. Second, it is important to minimize ballots that require adjudication. In many states such as Colorado, where the Pueblo dataset originated, ballots that are “kicked” by scanners must be adjudicated by a bipartisan team of election judges who determine how the vote should be counted by a set of criteria [5]. This process is slow and potentially subjective. If a ballot scanner kicks too many ballots, counting will be cost prohibitive. Finally, if the model is too slow, using it in practice (such as in real-time as ballots are scanned) may be difficult.

Before accuracy could be computed, we needed to determine how targets labeled as adjudicated should be handled when calculating accuracy. Our models assigned each target one of three labels—vote, no vote, or adjudicate. By contrast, each target in the dataset was labeled as either a vote or a no vote. When computing accuracy, we assumed all adjudicated ballots would be correctly classified by the adjudication process. We separately evaluated the number of ballots that required adjudication. We show results for these metrics in Figure 6.

4.1 Baseline Model Performance

The baseline model performed better than we anticipated; however, it still struggled where we expected. First, it sometimes classified targets with small or light

Model	Targets Accurately Classified	Flagged for Adjudication
Baseline	68,540 (99.895%)	2,181 (3.179%)
CNN	68,588 (99.965%)	1,465 (2.135%)
Hybrid #1	68,597 (99.978%)	3,242 (4.725%)
Hybrid #2	68,557 (99.920%)	430 (0.627%)

Fig. 6: Performance of each model on the Humboldt dataset. The CNN misclassifies 67% fewer targets and flags 33% fewer ballots for adjudication versus the baseline.

marks as no votes because these marks did not contain enough dark pixels to pass either threshold and be classified as a vote or flagged for adjudication. Second, the model often classified targets with marks that were filled in and crossed out as votes, because these targets contained a higher percent than the second threshold of dark pixels. Figure 7 shows examples of misclassified targets.

4.2 CNN Model Performance

By comparison, the CNN model outperformed the baseline model in both overall accuracy and number of ballots sent to a human. It had 66.7% fewer misclassifications and 32.8% fewer ballots flagged for adjudication versus the baseline.

The cases where the CNN model produced inaccurate classifications fell into a few categories. First, it appeared to be more sensitive than the baseline model to poor feature extraction and struggled off center targets. Fortunately, there exist more sophisticated techniques for ballot feature extraction than was used in this study [19]. Second, our model struggled with some of the X-marked targets. The CNN model occasionally labeled these targets as empty, causing it to predict no vote where a vote should have been. Figure 7 shows examples where the CNN model failed, but we emphasize that its overall performance was clearly superior to the baseline’s when comparing accuracy or adjudications.

Figure 8 shows how each model performed on targets the other classified correctly, incorrectly, or adjudicated. Notably, all marks that the CNN model misclassified were also misclassified or flagged for adjudication by the baseline model.

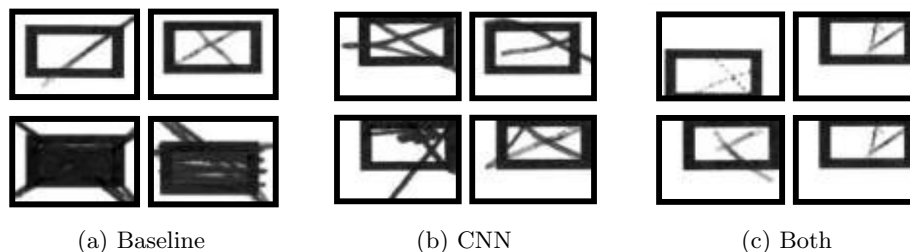


Fig. 7: Examples of misclassified targets from Humboldt ballots. The CNN performed better overall, but it failed in some cases with X marks or off-center targets.

		Baseline		
		Correct	Adjudicate	Incorrect
CNN	Correct	65,355	1,742	26
	Adjudicate	1,004	430	31
	Incorrect	0	9	15

Fig. 8: Overlapping performance of each model on 68,612 Humboldt targets.

4.3 Computational Costs

An additional metric to consider is computational cost. For the CNN, the most computationally expensive step was training the model. However, training need only be done once for each type of scanner hardware and style of voting target.

Ideally, a pre-trained model can predict labels for ballots at least as fast as they are scanned in, ensuring that the model is not a limiting component of the device as a whole. Today, a typical speed rating for a high-speed central-count optical scanner is on the order of 300 ballots per minute [28]. Different ballots contain vastly different numbers of targets, but an upper bound estimate for a traditional-style ballot might be 128 targets per page. With double-sided ballots, the high-speed scanner would need to process 1280 targets per second to keep up. Both the pre-fitted CNN and the baseline model far exceeded this rate, taking less than a second on a mid-line laptop to label the extracted, preprocessed features from the 68,612 targets in our test dataset. (Although feature extraction adds additional costs, these are the same with both models.) This indicates that the CNN approach can outperform the baseline in both accuracy and adjudication frequency while performing fast enough to keep pace with modern scanners.

4.4 Hybrid Models

After examining the results from the baseline and CNN models, we considered additional models that involved combining the two. We optimized the first of these hybrid models for accuracy. In this model, we flagged a contest’s targets for adjudication if either the baseline model or the CNN model labeled any of that contest’s targets as adjudicate, or if the two models disagreed on their predictions. This hybrid achieved a higher overall accuracy than either model alone. However, it also required adjudicating significantly more targets than either model alone did. Since this model was better than the CNN model by accuracy but worse by number of ballots adjudicated, it is not clearly an improvement. It is also worth noting that similar results might be possible from the CNN alone by increasing the confidence threshold at which the CNN model flags ballots for adjudication.

The second combined model we considered strove to maintain accuracy while reducing the number of ballots adjudicated. In this hybrid, we used the CNN model as a primary classifier, and when the CNN model chose to adjudicate, we used the baseline model to try to classify the target first. By accuracy, this model was still better than the baseline model but not as good as the CNN alone. By number of adjudications, this method was highly effective. It would be interesting

to investigate if one could increase the accuracy of this type of hybrid model by increasing the confidence threshold of the CNN. Like the first hybrid model, since this model was better than the CNN in one aspect but worse in the other, we cannot conclude which is decisively better. Figure 6 shows results for both hybrids.

4.5 Optimized Baseline Model

In addition to the performances of combined CNN and baseline models, we also investigated how a baseline model with different thresholds would have performed compared to the CNN. By starting with the Humboldt voting results and working backwards, it was possible to use a brute force approach to calculate which thresholds would produce the optimal results for this specific dataset given either a minimum accuracy or maximum adjudication rate condition.

If we insist that the baseline model achieves a lower adjudication rate than the CNN, $\alpha = 13.2\%$ and $\beta = 8.1\%$ maximized accuracy. This modified baseline achieved an accuracy of 99.862%—worse than even the original baseline model—and an adjudication rate of 2.035%. Likewise, if we modify the baseline model to have an accuracy higher than the CNN model, $\alpha = 99.8\%$ and $\beta = 1.7\%$ minimized adjudication. While this model had an accuracy of 99.968%, it would be virtually pointless as 97.042% of all contests required adjudication. This strongly suggests that there are types of marks, such as those marked and then crossed out, that simply cannot be correctly identified by a model that only looks at the shading of the target area.

4.6 Pueblo Test Results

We used the Pueblo dataset to more directly compare the CNN model to a deployed election system and to address concerns about whether a CNN could sometimes harm results. That is, in elections where current scanners perform well, would a CNN achieve a comparable accuracy? Once we had determined the efficacy of a CNN for the relatively messy Humboldt dataset, we retrained our model on the comparatively clean Pueblo dataset. Retraining was necessary, because the datasets use different styles of voting targets, and the raw scans, which were captured on different types of hardware, have vastly different intensity response characteristics. We used the same model architecture, only changing input/output sizes for the model layers. This model achieved similar training accuracy and loss to the Humboldt CNN model.

The Pueblo CNN model found 36 contests on 24 ballots with an overvote. When combined with 161 targets where feature extraction failed, this amounted to 0.0067% of the targets in the test dataset, and after accounting for the overvotes, the Pueblo model agreed with the post-adjudication ballot interpretations from the real election for every target in the test dataset of about 35,000 targets. This suggests that a CNN can produce accuracy as good as state-of-the-art deployed systems, while potentially requiring fewer ballots to be adjudicated.

Since the Pueblo dataset had extremely few marginal marks, the baseline also had a very high accuracy and made almost no mistakes, leaving little room to

improve upon the accuracy. However, our previous experiments showed that on datasets with a larger variety of marks, such as the Humboldt ballots, our CNN approach can achieve significant improvements to accuracy.

5 Discussion

We trained CNN models on the Humboldt dataset and the Pueblo dataset and found that they match or outperform the baseline threshold-intensity approach in terms of the number of correctly labeled targets and the number of ballots that require adjudication by election officials. A similar approach could be implemented in future elections. Scanner manufacturers could each train a model once on ballots that reflect their particular style of voting targets (e.g., ovals or rectangles) and hardware imaging characteristics (e.g., grayscale or one-bit black and white), then implement the model in a software update for their machines. This would potentially benefit future elections in multiple ways.

The benefits to increased labeling accuracy are clear. Better target classifications mean election results will better match voter intent. Demonstrated accuracy improvements may also increase public trust in the election process. Additionally, despite the expert consensus regarding the importance of rigorous post-election audits as a defense against both fraud and error [20], many states still do not require any form of tabulation audit, and very few perform risk-limiting audits [21]. As a result, the outcomes of the vast majority of contests currently depend on the accuracy of ballot scanners. Even when audits or manual recounts are applied, it is important for initial machine counts to be accurate, because if the audit or recount shows different counts, public confidence is likely to be eroded.

One of the biggest benefits of adjudicating fewer ballots is the time saved. When an absentee ballot is sent for review, election officials need to analyze it in the presence of multiple observers, determine voter intent, and then (for manual adjudication processes) copy the voter intent onto a new ballot and scan it. Reducing adjudication will save administrative costs and improve the speed at which election results are tabulated—which may help further increase voter confidence. Moreover, reducing the number of times voter intent needs to be determined by humans will reduce the potential for bias, subjectivity, and disputes.

5.1 Future Work

Our results suggest that application of machine learning techniques can achieve substantial improvements for the ballot scanning process, but we emphasize that far more work is possible. While our model was able to classify targets correctly with greater than 99.9% accuracy, outperforming the baseline model, there are numerous improvements that can be made to further enhance the performance of supervised learning techniques and better understand voter intent.

First, although our CNN model matched the performance of an actual scanner for the Pueblo dataset, which had very few marginal marks, further work is needed to more rigorously quantify the gains from CNN techniques against actual

deployed scanners when marginal marks are more common. The baseline model we implemented may be more capable towards marginal marks than some currently deployed tabulators, since it considers intensity within a fairly large region around the voting target, and so may underestimate the potential improvements.

Second, performance can very likely be enhanced by improving on the rather basic feature extraction methods that we used in the bulk of our experiments. Most of the mistakes in the Humboldt model originated from our target crops not being centered. A model trained on more structurally uniform, less variable data should better classify targets. In the Pueblo dataset, our feature extraction used timing marks and was more accurate than the Humboldt extraction. However, not all ballots utilize timing marks, and those that do not would benefit from the application of more sophisticated existing target extraction techniques (e.g., [30]).

Third, the performance of the CNN model can likely be greatly improved by training on a larger corpus of marginal marks, particular X-marks, check marks, and marked-and-crossed-out marks. With more data from these classes, models will be even better equipped to correctly classify these less common marks. Election officials could help accelerate this process by making larger and more complete datasets of scanned ballots available for research.

Fourth, more research is needed to investigate how ML techniques might provide even greater flexibility in understanding voter intent, such as by recognizing and processing marks that are not in the voting targets or in the small area around them. We found several examples of voters making marks and even writing in the margins of the ballots. These marks get ignored by both the current system and by our model. Scanners could potentially make better use of these marks for deciphering voter intent, whether by intelligently processing them or merely recognizing when they call for adjudication.

Finally, there is some evidence that demographic disparities exist in the rate of voter error when using existing ballot scanners [25, p. 19]. Since CNN models perform better when interpreting marginal marks, they might help reduce this bias. Research is needed to fully understand the causes and extent of bias in existing systems and to test how adopting a CNN model would affect it.

6 Conclusion

Marginal marks are a common feature on hand marked paper ballots, and current ballot scanning systems do not adequately account for them. In one dataset, we found that 8.5% of marked targets were not filled in completely, but rather consisted of X-marks, check marks, lightly filled targets, partially filled targets, and various forms of crossed-out targets. While traditional intensity-threshold methods are often able to classify such marginal marks correctly, we identified numerous cases where they either fail or require unnecessary human intervention.

By accounting for different kinds of marks and using a CNN trained to identify them, we were able to make ballot scanning more accurate. Compared to the baseline, we found that our model correctly classifies more targets and reduces the number of ballots sent to humans for review. While additional work is needed,

our research indicates that supervised learning has the potential to make ballot scanning smarter by counting ballots both faster and more accurately.

Acknowledgments

The authors thank Marilyn Marks and Harvie Branscomb for assistance acquiring ballot images and the students of EECS 498.5: Election Cybersecurity (Fall 2020) for their suggestions and feedback. We thank the Humboldt County Election Transparency Project and Pueblo County Elections for making ballot images available. We also thank our anonymous reviewers and our shepherd, Catalin Dragan. This material is based upon work supported by the Andrew Carnegie Fellowship, the U.S. National Science Foundation under grant number CNS-1518888, and a gift from Microsoft.

References

1. Bajcsy, A., Li-Baboud, Y.-S., and Brady, M.: Systematic Measurement of Marginal Mark Types on Voting Ballots. Tech. rep., <https://nvlpubs.nist.gov/nistpubs/ir/2015/NIST.IR.8069.pdf>. NIST (2015)
2. Bowen, D.: Top-to-Bottom Review of voting machines certified for use in California. Tech. rep., <https://www.sos.ca.gov/elections/voting-systems/oversight/top-bottom-review/>. California Secretary of State (2007)
3. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
4. Clark, A.: *Pillow (PIL Fork) Documentation*, (2015). <https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf>. 2015
5. Colorado Secretary of State Elections Division: *Voter Intent: Determination of Voter Intent for Colorado Elections*, (2013). <https://www.broomfield.org/DocumentCenter/View/11702/Voter-Intent-Guide>. Sept. 2013
6. Cordero, A., Ji, T., Tsai, A., Mowery, K., and Wagner, D.: Efficient User-Guided Ballot Image Verification. In: Electronic Voting Technology/Workshop on Trustworthy Elections. EVT/WOTE (2010)
7. *Curling v. Raffensperger*, Civil Action No. 1:17-cv-2989-AT (N.D. Ga. Oct. 11, 2020)
8. Dominion Voting Systems: *AuditMark*, <https://www.dominionvoting.com/democracy-suite-ems/>
9. He, K., Zhang, X., Ren, S., and Sun, J.: Deep Residual Learning for Image Recognition. CoRR **abs/1512.03385** (2015). arXiv: 1512.03385. <http://arxiv.org/abs/1512.03385>
10. Humboldt County: *November 3, 2009 UDEL Election: Official Canvass Precinct Report*, <https://humboldt.gov/DocumentCenter/View/3941/November-3-2009-UDEL-Election---Official-Canvass-Precinct-Report-PDF>
11. Hursti, H.: *Critical Security Issues with Diebold Optical Scan Design*, The Black Box Report (2005). July 2005
12. Jones, D.W.: On optical mark-sense scanning. In: Towards Trustworthy Elections, pp. 175–190. Springer (2010)
13. Kamarck, E., Ibreak, Y., Powers, A., and Stewart, C.: *Voting by mail in a pandemic: A state-by-state scorecard*, Brookings Institution (2020). <https://www.brookings.edu/research/voting-by-mail-in-a-pandemic-a-state-by-state-scorecard/>. 2020

14. Kiayias, A., Michel, L., Russell, A., and Shvartsman, A.: *Security assessment of the Diebold optical scan voting terminal*, (2006). https://voter.engr.uconn.edu/voter/wp-content/uploads/uconn_report-os.pdf. Nov. 2006
15. Kiayias, A., Michel, L., Russell, A., Shashidhar, N., See, A., and Shvartsman, A.: An authentication and ballot layout attack against an optical scan voting terminal. In: USENIX/ACCURATE Electronic Voting Technology Workshop (EVT) (2007)
16. LeNail, A.: NN-SVG: Publication-Ready Neural Network Architecture Schematics. *Journal of Open Source Software* **4**(33), 747 (2019). DOI: 10.21105/joss.00747. <https://doi.org/10.21105/joss.00747>
17. Lindeman, M., and Stark, P.: A Gentle Introduction to Risk-Limiting Audits. *IEEE Security and Privacy* **10**, 42–49 (2012)
18. McDaniel, P., Blaze, M., and Vigna, G.: EVEREST: Evaluation and Validation of Election-Related Equipment, Standards and Testing. Tech. rep., <http://siis.cse.psu.edu/everest.html>. Ohio Secretary of State (2007)
19. Nagy, G., Lopresti, D., Smith, E.H.B., and Wu, Z.: Characterizing Challenged Minnesota Ballots. In: 18th Document Recognition and Retrieval Conference (2011)
20. National Academies of Sciences, Engineering, and Medicine: *Securing the Vote: Protecting American Democracy*. The National Academies Press, Washington, DC (2018). <https://www.nap.edu/catalog/25120/securing-the-vote-protecting-american-democracy>
21. National Conference of State Legislatures: *Post-Election Audits*, (2019). <http://www.ncsl.org/research/elections-and-campaigns/post-election-audits635926066.aspx>. Jan. 2019
22. Poulos, J., Hoover, J., Ikonomakis, N., and Obradovic, G.: *Marginal Marks with Pixel Count*, U.S. Patent 9,710,988 B2 (2012). 2012
23. Pueblo County Elections: *Ballot Images, November 2020 Election*, <https://county.pueblo.org/clerk-and-recorder/ballot-images>
24. Rivest, R.: On the notion of ‘software independence’ in voting systems. *Phil. Trans. R. Soc. A* **366**(1881), 3759–3767 (2008)
25. State of Georgia: *Report of The 21st Century Voting Commission*, (2001). https://voterga.files.wordpress.com/2014/11/21st_century_report.pdf. July 2001
26. Sultana, F., Sufian, A., and Dutta, P.: Advancements in Image Classification using Convolutional Neural Network. *ICRCICN* (2018). DOI: 10.1109/icrcicn.2018.8718718. <http://dx.doi.org/10.1109/ICRCICN.2018.8718718>
27. Toledo, J.I., Cucurull, J., Puiggalí, J., Fornés, A., and Lladós, J.: Document Analysis Techniques for Automatic Electoral Document Processing: A Survey. In: *International Conference on E-Voting and Identity - Volume 9269. E-VoteID 2015*, pp. 129–141. Springer-Verlag, Bern, Switzerland (2015)
28. U.S. Election Assistance Commission: *Central Count Optical Scan Ballots* (2008). https://www.eac.gov/sites/default/files/document_library/files/Quick_Start_Guide_-_Central_Count_Optical_Scan_Ballots.pdf
29. Verified Voting: *The Verifier: Polling Place Equipment*, (2021). <https://www.verifiedvoting.org/verifier/>. 2021
30. Wang, K., Kim, E., Carlini, N., Motyashov, I., Nguyen, D., and Wagner, D.: Operator-Assisted Tabulation of Optical Scan Ballots. In: *Electronic Voting Technology/Workshop on Trustworthy Elections. EVT/WOTE* (2012)
31. Xiu, P., Lopresti, D., Baird, H., Nagy, G., and Smith, E.B.: Style-Based Ballot Mark Recognition. In: *10th International Conference on Document Analysis and Recognition* (2009)

Enhancing Privacy in Voting

Who was that masked voter? The tally won't tell!

Peter Y. A. Ryan¹[0000–0002–1677–9034], Peter B. Roenne¹[0000–0002–2785–8301],
 Dimiter Ostrev¹[0000–0002–4098–0969], Fatima-Ezzahra El Orche^{1,3}, Najmeh
 Soroush¹[0000–0002–9837–2376], and Philip B. Stark²[0000–0002–3771–9604]

¹ Interdisciplinary Centre for Security, Reliability, and Trust, SnT, University of
 Luxembourg

² Department of Statistics, University of California, Berkeley, CA, USA

³ ENS, CNRS, PSL Research University, Paris, France

{firstname.lastname}@uni.lu, stark@stat.berkeley.edu,
 fatimaezzahra.elorche@ens.fr

Abstract. We consider elections that publish anonymised voted ballots or anonymised cast-vote records for transparency or verification purposes, investigating the implications for privacy, coercion, and vote selling and exploring how partially masking the ballots can alleviate these issues.

Risk Limiting Tallies (RLT), which reveal only a random sample of ballots, were previously proposed to mitigate some coercion threats. Masking some ballots provides coerced voters with plausible deniability, while risk-limiting techniques ensure that the required confidence level in the election result is achieved. Risk-Limiting Verification (RLV) extended this approach to masking a random subset of receipts or trackers.

Here we show how these ideas can be generalised and made more flexible and effective by masking at a finer level of granularity: at the level of the components of ballots. In particular, we consider elections involving complex ballots, where RLT may be vulnerable to pattern-based vote buying. We propose various measures of verifiability and coercion-resistance and investigate how several sampling/masking strategies perform against these measures. Using methods from coding theory, we analyse signature attacks, bounding the number of voters who can be coerced. We also define new quantitative measures for the level of coercion-resistance without plausible deniability and the level of vote-buying-resistance without “free lunch” vote sellers.

These results and the different strategies for masking ballots are of general interest for elections that publish ballots for auditing, verification, or transparency purposes.

1 Introduction

Some voting systems, including many end-to-end verifiable systems and some conventional elections, publish the (plaintext) ballots. If these ballots are suitably anonymised, by for example verifiable mixes published on a bulletin board, then this is typically quite safe. But in some contexts, revealing such information may

be problematic: certain corner cases, such as unanimous votes or absence of any votes for a candidate and coercion threats, such as signature attacks.

In [4] the idea of Risk-Limiting Tallies (RLT) and Risk-Limiting Verification (RLV) was proposed to mitigate such threats. The idea is to shroud a proportion of the (anonymised) votes so voters can plausibly claim to have complied with the coercer, even though no votes appear for the candidate demanded by the coercer or no ballot with the pattern demanded by the coercer shows up in the tally. The proportion left shrouded can be adjusted using risk-limiting techniques to ensure that the confidence in the announced outcome achieves the required threshold, e.g, 99%. The idea extends to the verification aspects: shrouding some proportion of receipts or trackers. This proves particularly effective in for example the Selene scheme to counter the “sting in the tail”: the coercer claiming that the voter’s fake tracker is his own.

In this paper we note that, despite the pleasing features of the constructions of [4] there are still some drawbacks, in particular if the ballots are rather complex. While RLT may disincentivize *coercion*, there may still be an incentive for *vote buying*: the voter might still cast the required pattern vote in the hope that it will be revealed. Further, it has been suggested that RLT is arguably undemocratic in that some voters’ ballots do not contribute to the final tally. The second objection can be countered by arguing that every vote has an equal probability of being included in the count and that the outcome will be, with whatever confidence level required, a correct reflection of all votes cast. Nonetheless, it is an aspect that some people find troubling. A pleasing side effect of our construction is that all ballots are treated on an equal footing.

These observations suggest exploring different ways to apply RLT and RLV when ballots are complex: rather than shrouding entire ballots at random, we shroud, at random, some preferences on each ballot. In effect we are filtering the tally horizontally rather than vertically. This hits both of the issues above: the chance any given pattern remains identifiable after the filtering is reduced, and every ballot contributes to the outcome, albeit not necessarily to every contest. In the *full tally* construction below, every ballot contributes fully to the announced outcome, but we shroud the link between the tracker and some components of the ballots. For tracker-based schemes, the voters can verify some but not all of their selections. This paper seeks to quantify these effects and explore trade-offs among them.

Our techniques allow us to state and prove bounds on the number of voters an adversary is able to attack using pattern-based or “signature” attacks. Note that assigning the same, or similar, complex ballot pattern to many voters is counterproductive for the adversary: if even a few voters comply, the rest can point to the signature ballots that already appear and claim compliance. Thus, an adversary who wants to influence many voters with a signature attack must be able to produce many distinguishable ballot patterns. This observation motivates us to prove lower and upper bounds on the number of distinguishable patterns an adversary can construct. We prove these bounds using a connection to a well-studied problem in the theory of error-correcting codes.

This ballot-masking method and its privacy implications are interesting not only in for RLT and RLV but for all schemes where all or some ballots are published for auditing, verification, or transparency. As an example, Colorado is currently redacting cast-vote records (CVRs) by removing entire CVRs, e.g., for rare ballot styles; partial masking has been considered as an alternative. We note, however, that masking parts of the ballot might make it hard to detect ill-formed, e.g., over-votes etc.

We also note that this idea has similarities to the SOBA constructions for Risk-Limiting-Audits (RLAs), [1], which also publishes each audited ballot “disassembled” into different contests, whereas the auditors will see the intact ballot. The VAULT approach [2] also uses homomorphic encryption of the cast-vote records to achieve the SOBA goals more easily. (VAULT was used for the first time in a risk-limiting audit in Inyo County, California, in 2020.) The purpose and the underlying cryptographic constructions are quite different, but our analysis applies to these cases as well.

For some tally algorithms, we can separate ballots into their atomic parts and reveal these independently after anonymising them, which effectively counters signature attacks. However, that reduces public transparency and may reduce public confidence in the election result. For Selene, where voters verify their votes via trackers, this separation provides a method to verify without revealing individual ballots: we simply assign a distinct tracker to each element of the ballot. Voters can then verify some or all components of their ballot using those trackers. A coerced voter could use the Selene tracker-faking mechanism to assemble a ballot that matches the coercer’s instructions. Technically this is straightforward but from a usability standpoint seems problematic. Moreover, even if the voter were prepared to go the effort of concocting such a fake ballot, the necessary ingredients might not be available, so coercion threats will remain, and the probability that one of atomic trackers is the same as the coercer’s increases. Thus it makes sense to look for alternatives.

Below, we present the main ideas and analyse differences in privacy, coercion-resistance, and receipt-freeness for the different methods. Section 2 introduces the idea of partially masking ballots. Section 3 describes how it can be used in masked RLT and RLV. Section 4 defines a distinguishing distance between randomly masked ballots, establishes a connection to the Hamming distance, characterizes the class of masking strategies for which this connection holds, and proves bounds on the number of voters that can be approached with a pattern-based attack. It provides another application of the distinguishing distance: to quantify the effect of masking on individual verifiability. Section 5 considers quantitative game-based notions of privacy, coercion-resistance, and receipt-freeness. Section 6 concludes.

2 Masking Complex Ballots

Many elections use simple plurality voting: the voter selects at most one candidate from a set, in the simplest case, a referendum, a choice between “yes” and “no.” The next level of complexity is single-winner plurality, aka “first past the post.”

More complex social choice functions and correspondingly more complex ballots are common. Perhaps the next level in complexity are *approval voting* in which the voter can cast votes for several candidates for a single office, and multi-winner plurality, in which a voter can vote for up to k candidates for k offices. In some cases voters may have a quota of votes and is allowed to cast more than one vote for a given candidate, up to some limit. Some methods allow voters to give a preference ranking to the candidates.

Common to all of these social choice functions, if the ballots are published, is that they are vulnerable to signature attacks (also known as “Italian” attacks), i.e. a coercer chooses a particular, unlikely, pattern, instructs the victim to mark a ballot with that pattern and checks whether a ballot with that pattern appears in the tally.

Let us assume that the ballots are of the form (v_1, v_2, \dots, v_k) with k the number of candidates and v_i taking values from a specified set \mathcal{V} . \mathcal{V} might for example just be $\{0, 1\}$ or a set of integers plus a blank: $\{1, \dots, s\} \cup \{\text{blank}\}$ etc.

In many types of elections, these ballot-level selections, or subsets thereof, will reappear as part of the tally procedure (e.g. in electronic mixnet tallies), as part of an audit trail or for transparency (electronic scans of paper ballots), in Risk-Limiting Audits using samples of votes, or verification procedures (e.g. in tracker-based schemes such as Selene). In order to preserve privacy, the mapping between the published votes and the voter is normally anonymised.

As mentioned above, revealing these ballots may endanger the receipt-freeness of the election. With *Masked Tallies*, introduced here, only parts of each ballot are revealed:

$$(\text{mask}_{i1}(v_1^{(i)}), \text{mask}_{i2}(v_2^{(i)}), \dots, \text{mask}_{ik}(v_k^{(i)})) \quad \text{for } i = 1, \dots, n.$$

The functions mask_{ij} are either the identity, displaying the component of the vote, or a constant, .e.g. $*$ ($\notin \mathcal{V}$), masking the component. n is the number of ballots cast.

Risk-Limiting Tallies [4], involved unmasking only as many randomly selected ballots as are needed to determine the election result with a chosen risk limit. The remaining ballots were kept completely masked. Here we suggest a generalization, allowing partial masking of the ballots, and we will discuss the impact on risk limits, privacy, coercion-resistance, and resistance to vote-buying.

3 Partially Masked RLTs and RLVs

We reprise risk-limiting tallies and verification, RLT and RLV [4], before extending these to general masks. First we recapitulate the idea of tracker-based verification in terms of Selene.

Outline of Selene Selene [8] enables verification by posting the votes in the clear on the BB along with private tracking numbers. Voters are only notified of their tracker some time after the vote/tracker pairs have been publicly posted, giving a coerced voter the opportunity to choose an alternative tracker to placate

the coercer. The voter is able to fake the tracker and related cryptographic data using a secret trapdoor key. The notification of the trackers is carefully designed to provide assurance to the voter that it is their correctly assigned tracker, i.e. unique to them, while being deniable to any third party.

Assuming that votes are encrypted component-wise, at the end of the mixing we will have encrypted votes and trackers on the bulletin board:

$$(\{tr_i\}_{PK}, (\{v_1^{(i)}\}_{PK}, \{v_2^{(i)}\}_{PK}, \dots, \{v_k^{(i)}\}_{PK}))$$

where $\{\cdot\}_{PK}$ denotes encryption under the public key PK . These ballots can now be verifiably decrypted to reveal the vote/tracker pairs that can be checked by the voters, and anyone can compute the tally directly on the plaintext votes.

Risk-Limiting Tallies and Verification with Partially Masked Ballots

In the original approach to RLT (where ballots are without trackers) and RLV (with trackers for individual verification), see [4], the idea was to only decrypt a random subset of the ballots. The number decrypted being controlled by a risk-limit that bounds the probability that the announced election result will be wrong.

In the new masked RLV and RLT approach, we instead reveal randomly selected components of the ballots (and the trackers for RLV). If there is more than one contest on the ballot, the contests can be treated independently. How much we reveal will again be governed by a specified risk limit, as in [4]. A natural choice is to first decrypt m of the k entries in each ballot at random, and to increase m if necessary to meet the risk limit. This is simplest and will be used in the analysis below. In practice, it may make sense to dynamically change the rate of openings per candidate, e.g. if a candidate is popular we might be able to decrease the rate of unmasking of votes for that candidate, maintaining the risk limit while improving coercion-resistance.

Using this masked approach for RLV with tracker verification, the masking means that only parts of the ballot can be verified, but unlike to the original RLV every voter can verify *something*. We will quantify how much.

Full Tally with Partial Verification (FTPV) A social choice function is *separable* if, for the purposes of tallying, the components of each vote can be considered separately. Plurality, approval, and Borda count are separable; instant-runoff voting and single transferrable vote are not. For separable social choice functions, it is possible to compute the full tally, i.e. achieve 100% confidence in the outcome while partially masking selections. For each ballot, we randomly select some components. All selected components for all ballots are gathered in another part of the BB and subjected to a full, componentwise shuffling before decryption. Their positions in the original ballots are replaced by *. Thus, the way that these selected components appeared in the original ballots is lost.

The FTPV approach above might still hit corner cases, for instance if no vote was cast for a particular candidate. This suggests using a hybrid approach in which we use the approach above but reveal a random subset of the components

separated out from the ballots. Thus we reveal enough of each ballot linked to the tracker to make verification meaningful while mitigating coercion threats, while a larger portion of the ballots is revealed without a link to the trackers to attain the required risk limit for the tally.

4 Distinguishing distance and applications to signature attacks and individual verifiability

In this section, we define a metric on the set of complex ballots that characterizes how well pairs of strings can be distinguished under random masking. We then observe that in some cases this metric is a monotone transformation of the Hamming distance used in coding theory. We also precisely characterize the cases when this occurs. Next, we use the connection to coding theory to answer the following question: how many simultaneous signature attacks can a coercer and/or vote-buyer launch? Finally, we give another application of the distinguishing distance: we use it to quantify the effect of a masking strategy on individual verifiability.

Throughout this section, we consider complex ballots with k components taken from the set \mathcal{V} ; thus, the set of possible ballots is \mathcal{V}^k . We ignore here any constraints on what constitute valid ballots. For $x \in \mathcal{V}^k$ and $S \subset \{1, \dots, k\}$, we denote by x_S the substring of x on the positions in S .

4.1 Definition and basic properties of distinguishing distance

How distinguishable are pairs of elements of \mathcal{V}^k under masking? For every probability distribution p_S over subsets of $\{1, \dots, k\}$, for every $x \in \mathcal{V}^k$ there is an induced probability distribution q_{S, x_S} of the pair (S, x_S) , given by $q_{S, x_S}(s, \alpha) = p_S(s)\delta_{x_s, \alpha}$. If we keep p_S fixed and consider a pair $x, y \in \mathcal{V}^k$, we can define the distance between x and y as the statistical distance of q_{S, x_S}, q_{S, y_S} ; thus, we take

$$d_{p_S}(x, y) = \frac{1}{2} \|q_{S, x_S} - q_{S, y_S}\|_1 = \sup_D |\Pr(D(S, x_S) = 1) - \Pr(D(S, y_S) = 1)|, \quad (1)$$

where the supremum is over distinguishers D . We can obtain the following formula for d_{p_S} :

Proposition 1. *For all distributions p_S , for all $x, y \in \mathcal{V}^k$,*

$$d_{p_S}(x, y) = \sum_{s: x_s \neq y_s} p_S(s) = \sum_s p_S(s) \mathbb{I}(s \cap t \neq \emptyset)$$

where t is the set of positions on which x, y differ and the operator \mathbb{I} transforms the true/false value of a statement to 1, 0 respectively.

Proof.

$$\begin{aligned}
d_{p_S}(x, y) &= \frac{1}{2} \|q_{S, x_S} - q_{S, y_S}\|_1 = \sum_{(s, \alpha): q_{S, x_S}(s, \alpha) > q_{S, y_S}(s, \alpha)} (q_{S, x_S}(s, \alpha) - q_{S, y_S}(s, \alpha)) \\
&= \sum_{(s, \alpha): q_{S, x_S}(s, \alpha) > q_{S, y_S}(s, \alpha)} (p_S(s)\delta_{x_s, \alpha} - p_S(s)\delta_{y_s, \alpha}) \\
&= \sum_{s: x_s \neq y_s} p_S(s) = \sum_s p_S(s) \mathbb{I}(s \cap t \neq \emptyset)
\end{aligned}$$

□

Under the mild assumption that each position is revealed with strictly positive probability, d_{p_S} is a metric on \mathcal{V}^k .

Proposition 2. *For all p_S , d_{p_S} is symmetric, satisfies the triangle inequality and satisfies $\forall x, d_{p_S}(x, x) = 0$. If in addition $\forall i, \Pr(i \in S) > 0$, then $d_{p_S}(x, y) = 0 \implies x = y$.*

Proof. The first three claims follow directly from (1). For the last claim, take any i , any $v \in \mathcal{V}$, any x, y with $d_{p_S}(x, y) = 0$. Consider the distinguisher D given by “On input s, α , if i is among the revealed positions and the corresponding entry is v output 1, else output zero.” Then,

$$\Pr(i \in S)\delta_{x_i, v} = \Pr(D(S, x_S) = 1) = \Pr(D(S, y_S) = 1) = \Pr(i \in S)\delta_{y_i, v}.$$

Therefore, $\forall i \forall v, x_i = v \iff y_i = v$, so $x = y$. □

Now, we look at another question: how to find an optimal distinguisher between a pair of strings. For each $x \in \mathcal{V}^k$, define distinguisher D_x by “On input (s, α) , if $x_s = \alpha$, output 1, else output 0.” This is optimal regardless of the particular p_S , and regardless of the particular second element y .

Proposition 3. *For all distributions p_S , for all $x, y \in \mathcal{V}^k$,*

$$d_{p_S}(x, y) = \Pr(D_x(S, x_S) = 1) - \Pr(D_x(S, y_S) = 1)$$

Proof.

$$\begin{aligned}
&\Pr(D_x(S, x_S) = 1) - \Pr(D_x(S, y_S) = 1) \\
&= \sum_s p_S(s) (\Pr(D_x(s, x_s) = 1) - \Pr(D_x(s, y_s) = 1)) \\
&= \sum_s p_S(s) (1 - \delta_{x_s, y_s}) = \sum_{s: x_s \neq y_s} p_S(s) = d_{p_S}(x, y)
\end{aligned}$$

□

4.2 Distinguishing distance and Hamming distance

From Proposition 1, we see that for any p_S , $d_{p_S}(x, y)$ does not depend on all details of the strings x, y , but only on the set of positions where x, y differ. It turns out that there is a class of distributions p_S such that d_{p_S} does not even

depend on all details of the set of positions where x, y differ, but only on the Hamming distance between x and y , $d_H(x, y) = |\{i : x_i \neq y_i\}|$. This class of probability distributions is precisely those that assign equal weight to subsets of equal size.

Theorem 1. *For all p_S , the following are equivalent:*

1. *There exists a probability vector $(r(0), \dots, r(k))$ such that $\forall s, p_S(s) = \frac{r(|s|)}{\binom{k}{|s|}}$*
2. *There exists a function f_{p_S} such that for all $x, y \in \mathcal{V}^k$, $d_{p_S}(x, y) = f_{p_S}(d_H(x, y))$.*

We prove the forward direction of Theorem 1 by computing an explicit formula for the function f_{p_S} .

Theorem 2. *Suppose $\exists(r(0), \dots, r(k)) \forall s, p_S(s) = \frac{r(|s|)}{\binom{k}{|s|}}$ Then,*

$$d_{p_S}(x, y) = \sum_{i=1}^{d_H(x,y)} \sum_{j=0}^{k-d_H(x,y)} \frac{\binom{d_H(x,y)}{i} \binom{k-d_H(x,y)}{j} r(i+j)}{\binom{k}{i+j}}$$

Proof (Theorem 2). Take any x, y and let t be the subset of positions where x, y differ. Then,

$$d_{p_S}(x, y) = \sum_{s: x_s \neq y_s} p_S(s) = \sum_{s: s \cap t \neq \emptyset} p_S(s) = \sum_{i=1}^{|t|} \sum_{j=0}^{k-|t|} \frac{\binom{|t|}{i} \binom{k-|t|}{j} r(i+j)}{\binom{k}{i+j}} \quad \square$$

To prove the reverse direction of Theorem 1, we think of the $2^k - 1$ dimensional vector space over \mathbb{C} with entries indexed by non-empty subsets of $\{1, \dots, k\}$, we think of the subspace

$$W = \{w \in \mathbb{C}^{2^k - 1} : |s| = |t| \implies w(s) = w(t)\}$$

and we also think of the $(2^k - 1) \times (2^k - 1)$ matrix M with entries $M(s, t) = \mathbb{I}(s \cap t \neq \emptyset)$ indexed by non-empty subsets of $\{1, \dots, k\}$.

From Theorem 2, we see that $w \in W \implies Mw \in W$, that is, M leaves the subspace W invariant. Next, we observe that M is self-adjoint, and that M is also invertible:

Theorem 3. *For all $k \in \mathbb{N}$, the matrix M_k with entries $M_k(s, t) = \mathbb{I}(s \cap t \neq \emptyset)$ indexed by non-empty subsets of $\{1, \dots, k\}$ is invertible.*

a fact that we will prove at the end of this subsection. From this, we see that M^{-1} also leaves subspace W invariant.

Now, assume $\exists f_{p_S}, \forall x, y : d_{p_S}(x, y) = f_{p_S}(d_H(x, y))$. Form the vector $w \in W$ with entries $w(t) = f_{p_S}(|t|)$. The relation $d_{p_S}(x, y) = f_{p_S}(d_H(x, y))$ and Proposition 1 imply $\forall t, w(t) = \sum_{s \neq \emptyset} M(t, s) p_S(s)$. Then, $(p_S(s))_{s \neq \emptyset} = M^{-1} w \in W$, so p_S assigns equal weight to subsets of equal size. This completes the proof of Theorem 1, assuming Theorem 3 holds.

It remains to prove Theorem 3. The proof is by induction on k . When $k = 1$, $M_1 = (1)$ is invertible. Assume now M_k is invertible and consider M_{k+1} . We order subsets according to the following: a subset corresponds to a string of 0s

and 1s, and this encodes an integer between 1 and $2^{k+1} - 1$. With this ordering of the subsets, the matrix M_{k+1} has the following block form:

$$\begin{pmatrix} M_k^{(2^k-1) \times (2^k-1)} & 0^{(2^k-1) \times 1} & M_k^{(2^k-1) \times (2^k-1)} \\ 0^{1 \times (2^k-1)} & 1^{1 \times 1} & 1^{1 \times (2^k-1)} \\ M_k^{(2^k-1) \times (2^k-1)} & 1^{(2^k-1) \times 1} & 1^{(2^k-1) \times (2^k-1)} \end{pmatrix}$$

where the sizes of the blocks are indicated in the superscript, and a 0 or 1 indicates that all entries of that block are 0 or 1.

Now we consider the following elementary row operations: subtract the middle row from all the bottom rows, then subtract the top block of rows from the bottom block of rows. We arrive at the matrix

$$\begin{pmatrix} M_k^{(2^k-1) \times (2^k-1)} & 0^{(2^k-1) \times 1} & M_k^{(2^k-1) \times (2^k-1)} \\ 0^{1 \times (2^k-1)} & 1^{1 \times 1} & 1^{1 \times (2^k-1)} \\ 0^{(2^k-1) \times (2^k-1)} & 0^{(2^k-1) \times 1} & (-M_k)^{(2^k-1) \times (2^k-1)} \end{pmatrix}$$

and this is invertible by the inductive hypothesis. Hence, M_{k+1} is also invertible.

4.3 Bounds on the number of simultaneous signature attacks

We consider a coercer and/or vote buyer who wants to launch signature attacks on multiple voters simultaneously. Thus, the adversary chooses r signatures $x_1, \dots, x_r \in \mathcal{V}^k$ and approaches many voters requiring each to submit one of the signature ballots.

What is the largest number r_{max} of different signatures that a coercer can use subject to the natural constraint that the strings x_1, \dots, x_r are pairwise distinguishable under random masking? We use the connection to coding theory from subsection 4.2 to answer this question.

First, we prove some properties of the function f_{p_S} from Theorem 1.

Lemma 1. *For every p_S that satisfies $\exists(r(0), \dots, r(k)) \forall s, p_S(s) = \frac{r(\binom{|s|}{k})}{\binom{|s|}{k}}$, the function f_{p_S} is non-decreasing, $f_{p_S}(0) = 0$, and $f_{p_S}(k) = 1 - p_S(\emptyset)$.*

Proof. Take any $i < j \in \{0, \dots, k\}$. Take $x, y \in \mathcal{V}^k$ that differ in the first i positions and $x', y' \in \mathcal{V}^k$ that differ in the first j positions. Using Proposition 1 we get $f_{p_S}(i) = f_{p_S}(d_H(x, y)) = d_{p_S}(x, y) = \sum_s p_S(s) \mathbb{I}(s \cap \{1, \dots, i\} \neq \emptyset) \leq \sum_s p_S(s) \mathbb{I}(s \cap \{1, \dots, j\} \neq \emptyset) = d_{p_S}(x', y') = f_{p_S}(d_H(x', y')) = f_{p_S}(j)$.

For the other two claims, take $z, w \in \mathcal{V}^k$ that differ in all positions. Then,

$$\begin{aligned} f_{p_S}(0) &= f_{p_S}(d_H(z, z)) = d_{p_S}(z, z) = 0 \\ f_{p_S}(k) &= f_{p_S}(d_H(z, w)) = d_{p_S}(z, w) = \sum_s p_S(s) \mathbb{I}(s \cap \{1, \dots, k\} \neq \emptyset) = 1 - p_S(\emptyset) \quad \square \end{aligned}$$

The properties of f_{p_S} established in Lemma 1 allow us to define a partial inverse of f_{p_S} . Take $g_{p_S} : [0, 1 - p_S(\emptyset)] \rightarrow \{0, 1, \dots, k\}$ given by

$$g_{p_S}(q) = \min\{d \in \{0, 1, \dots, k\} : f_{p_S}(d) \geq q\}$$

so that we have

$$f_{p_S}(d) \geq q \iff d \geq g_{p_S}(q) \quad (2)$$

Now, we are ready to state and prove our bounds on the number of simultaneous signature attacks under a pairwise distinguishability constraint.

Theorem 4. *For every finite set \mathcal{V} , for every $k \in \mathbb{N}$, for every probability distribution p_S on subsets of $\{1, \dots, k\}$ satisfying $\exists(r(0), \dots, r(k)) \forall s, p_S(s) = \frac{r(|s|)}{\binom{k}{|s|}}$, for every $q \in [0, 1 - p_S(\emptyset)]$, let $r_{max}(\mathcal{V}, k, p_S, q)$ denote the size of the largest collection $\{x_1, \dots, x_r\}$ with the property $\forall i \neq j, d_{p_S}(x_i, x_j) \geq q$. Then*

$$\frac{|\mathcal{V}|^k}{\sum_{j=0}^{g_{p_S}(q)-1} \binom{k}{j} (|\mathcal{V}| - 1)^j} \leq r_{max}(\mathcal{V}, k, p_S, q) \leq \frac{|\mathcal{V}|^k}{\sum_{j=0}^{\lfloor (g_{p_S}(q)-1)/2 \rfloor} \binom{k}{j} (|\mathcal{V}| - 1)^j}$$

Proof. We use the same argument that is used in coding theory to establish the Gilbert-Varshamov lower bound and the Hamming upper bound on the maximum number of codewords subject to a pairwise Hamming distance constraint.

First, we observe that a collection $\{x_1, \dots, x_r\}$ satisfies $\forall i \neq j, d_{p_S}(x_i, x_j) \geq q$ if and only if it satisfies $\forall i \neq j, d_H(x_i, x_j) \geq g_{p_S}(q)$. This follows from the relation $d_{p_S}(x_i, x_j) = f_{p_S}(d_H(x_i, x_j))$ and the property (2) of the partial inverse g_{p_S} .

Now, take a collection $\{x_1, \dots, x_{r_{max}(\mathcal{V}, k, p_S, q)}\}$ with the maximum number of elements subject to the constraint $\forall i \neq j, d_H(x_i, x_j) \geq g_{p_S}(q)$. To prove the upper bound, note that the Hamming balls of radius $\lfloor (g_{p_S}(q) - 1)/2 \rfloor$ around $x_1, \dots, x_{r_{max}}$ must be disjoint, that each such ball contains $\sum_{j=0}^{\lfloor (g_{p_S}(q)-1)/2 \rfloor} \binom{k}{j} (|\mathcal{V}| - 1)^j$ elements, and that the total number of elements in all these balls must not exceed the size of the whole set \mathcal{V}^k .

To prove the lower bound, note that the Hamming balls of radius $g_{p_S}(q) - 1$ around $x_1, \dots, x_{r_{max}}$ must completely cover \mathcal{V}^k , or else another element could be found that has Hamming distance $\geq g_{p_S}(q)$ to all of $x_1, \dots, x_{r_{max}}$ and this would contradict the choice of $\{x_1, \dots, x_{r_{max}(\mathcal{V}, k, p_S, q)}\}$ as having the maximum number of elements. Now, we have r_{max} Hamming balls with $\sum_{j=0}^{g_{p_S}(q)-1} \binom{k}{j} (|\mathcal{V}| - 1)^j$ elements each and their total number of elements must exceed $|\mathcal{V}|^k$, giving the lower bound on r_{max} . \square

These upper and lower bounds are exemplified in Figure 1 for an election with $k = 5$ candidates and $|\mathcal{V}| = 2$ (like the student election example in next section). We have $g_{p_S}(q) = k - m + 1$ when g is applied to a uniform distribution over m -element subsets (m openings) evaluated at $q = 1$ (perfect distinguishability).

4.4 Quantifying the effect of masking on individual verifiability

We would like to quantify the effect of a particular masking strategy, specified by the probability distribution p_S , on individual verifiability. We propose the following quantity:

$$IV(p_S) = \inf_{x \neq y \in \mathcal{V}^k} d_{p_S}(x, y)$$

This quantity takes values between 0 and 1, where $IV(p_S) = 1$ means that the masking strategy p_S leaves the individual verifiability of the underlying voting

protocol invariant, while $IV(p_S) = 0$ means that the masking strategy p_S destroys any individual verifiability that was present in the underlying voting protocol.

The motivation for choosing the quantity $IV(p_S)$ is the following: a voter who has voted x obtains a pair (s, α) where $s \subset \{1, \dots, k\}$ and $\alpha \in \mathcal{V}^{|s|}$ and must decide whether this revealed vote was obtained from his submitted vote x or from some $y \neq x$. Taking the infimum over $x \neq y$ corresponds to considering the worst case over voter choices x and modifications of the voter choice y .

One attractive feature of this setup is that an individual voter does not need to know the distribution p_S or the modification y in order to apply the optimal verification strategy; indeed the optimal strategy for a voter who has chosen x is to apply the distinguisher D_x considered in Proposition 3.

For distributions p_S that satisfy $\exists(r(0), \dots, r(k)) \forall s, p_S(s) = \frac{r(|s|)}{\binom{k}{|s|}}$, Theorem 2 gives a simple formula for $IV(p_S)$:

$$IV(p_S) = \sum_{j=0}^{k-1} \frac{\binom{k-1}{j} r(j+1)}{\binom{k}{j+1}} = \sum_{l=1}^k \frac{l}{k} r(l)$$

where we have used the fact that the transformation from Hamming to distinguishing distance is non-decreasing (Lemma 1), and so the smallest distinguishing distance is between x, y such that $d_H(x, y) = 1$.

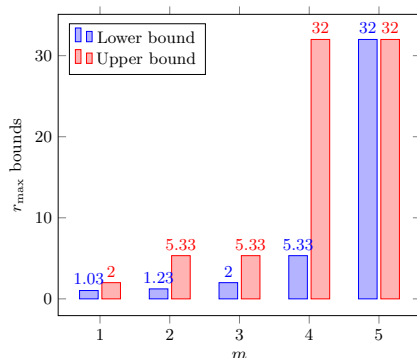


Fig. 1. Example for $|\mathcal{V}| = 2$ and $k = 5$

$m \setminus p$	p_{col}	$(1 - p_{col})^n$
1	0.16	$1.46 \cdot 10^{-79}$
2	0.018	$8.3 \cdot 10^{-9}$
3	0.0005	0.60
4	$9.7 \cdot 10^{-6}$	0.99
5	$1.6 \cdot 10^{-7}$	0.9998

Fig. 2. The probability, p_{col} that a single (resp. no) honest voter casts a ballot which after masking equals the mask of $v_0^O = (0, 1, 1, 1, 1)$ for the student election.

5 Quantitative Privacy-Type Properties

We now want to measure and compare privacy-properties for different masked tally methods. When computing concrete values we will consider approval voting with k candidates only 0 or 1 is allowed for each candidate, without any overall constraint, $(v_1, \dots, v_k) \in \{0, 1\}^k$. For the n honest voters we assume for simplicity that the probability to vote $v_i = 1$ is p_i and these probabilities are independent. As a special concrete case we consider a student election with $n = 1001$ voters (one voter is

under observation), $k = 5$ candidates with probabilities $(0.6, 0.4, 0.01, 0.01, 0.01)$, i.e. two popular candidates and three unpopular.

5.1 Privacy

In order to compare the different approaches we first consider the quantitative δ -privacy definition from [5]. The main other quantitative privacy definition is [3], but it is less suited considering signature attacks. The parties are an observer O , who can use public data, n_h honest voters and an additional voter under observation V_{obs} , whose vote the observer tries to guess.

Definition 1 (δ -privacy). *Let P be a voting protocol and V_{obs} be the voter under observation. We say that P achieves δ -privacy if*

$$\Pr[(\pi_O || \pi_{V_{obs}}(v_0^O) || \pi_v)^{(\ell)} \rightarrow 1] - \Pr[(\pi_O || \pi_{V_{obs}}(v_1^O) || \pi_v)^{(\ell)} \rightarrow 1]$$

is δ -bounded as a function of the security parameter ℓ for all vote choices v_0^O and v_1^O of the observed voter. Here π_O , $\pi_{V_{obs}}$ and π_v are respectively the programs run by the observer O , the voter under observation V_{obs} and all the honest voters.

The value δ will depend on the chosen vote distribution, and we see that it is especially relevant to penalize signature attacks: if we assume that there is a vote choice $v^* = (v_1^*, \dots, v_k^*)$ which rarely gets selected and has a probability close to zero, then an unmasked tally which reveals all cast plaintext ballots, even in anonymised form, will have $\delta = 1$ – the adversary simply checks if v^* appears.

Full ballot disclosure When we reveal all ballots, we can consider the case where the observer tries to distinguish a voter casting the most unpopular vote vs the most popular vote, as in a signature attack. That is, in the definition we let $v_0^O = (v_1, \dots, v_k)$ with $v_i = 1$ if $p_i \leq 1/2$ and $v_i = 0$ if $p_i > 1/2$, and we have $v_1^O = (1 - v_1, \dots, 1 - v_k)$. Denote the corresponding probability p_{min} . Now a good strategy is simply to check if at least one (v_1, \dots, v_k) appears in the disclosed ballots, and the algorithm then outputs “1”. This means $\Pr[(\pi_O || \pi_{V_{obs}}(v_0^O) || \pi_v)^{(\ell)} \rightarrow 1] = 1$ but $(\pi_O || \pi_{V_{obs}}(v_1^O) || \pi_v)$ will also output “1” if another voter chooses v_0^O . This happens with probability $1 - (1 - p_{min})^{n_h}$. We conclude that $\delta \geq (1 - p_{min})^{n_h}$. For the case of the student election we have that $v_0^O = (0, 1, 1, 1, 1)$ with $p_{min} = 0.4^2 \cdot 0.01^3 = 1.6 \cdot 10^{-7}$. Thus for $n_h = 1000$ we have $\delta \geq (1 - p_{min})^{n_h} \approx 0.99984$, i.e. close to 1.

Result Only We now consider the case where we only reveal the overall result $r = (r_1, \dots, r_k)$. In this case we can follow an analysis close to [5,7] for calculating δ . For every possible result r we calculate the probability that the result happened if the observed voter cast v_0^O or v_1^O . The algorithm will then output one if the former probability is larger. We get $\delta = \sum_{r \in M_{v_0^O, v_1^O}^*} (A_r^{v_0^O} - A_r^{v_1^O})$ where $M_{v_0^O, v_1^O}^* = \{r \in \mathbb{R} : A_r^{v_1^O} \leq A_r^{v_0^O}\}$, \mathbb{R} is the set of all possible results of the election and A_r^v denotes the probability that the choices of the

honest voters yield the result r given that V_{obs} 's choice is v . These probabilities can explicitly be calculated since each candidate count from the honest voters, X_i , is binomially distributed, $X_i \sim BD(n_h, p_i)$. We thus have $A_r^v = \mathbb{P}(X_1 = r_1 - v_1) \cdots \mathbb{P}(X_k = r_k - v_k) = \prod_{i=1}^k \binom{n-1}{r_i - v_i} p_i^{r_i - v_i} (1 - p_i)^{n - r_i + v_i - 1}$.

RLT In the original RLT method we keep a certain fraction, f_{blind} , of the ballots hidden, that is $(1 - f_{blind})n$ ballots are published. If we consider the optimal algorithm from the full ballot disclosure and the corresponding δ_{full} we see that $\delta = (1 - f_{blind})\delta_{full}$ since the probability that observed voter's ballot is hidden is $(1 - f_{blind})$.

Masked RLT We now consider the case of masked RLTs where the we release all ballots but with only m out of k components unmasked. A good strategy to lower bound δ is to count the number N_b of colliding ballots v which satisfy $\text{mask}_v v = \text{mask}_b v_b^O$ for $b = 0, 1$. We choose v_0^O as the most unlikely ballot, as above and take v_1^O as the opposite ballot to discriminate optimally between the two counts. The main distinguishing power comes from N_0 , and we let the distinguishing algorithm output "1" if the probability of the honest voters casting $N_0 - 1$ colliding votes is higher than getting N_0 collisions. The probability for each honest voter to have a collision is $p_{col} = 1 / \binom{k}{m} \cdot \sum_{1 \leq i_1 < i_2 < \dots < i_m \leq k} p_{i_1} \cdots p_{i_m}$ and $N_0 \sim BD(n_h, p)$, where p_i is the probability of a match in the i th candidate. In Fig. 2 we have displayed the probabilities for the student election example. The algorithm above will then simply give the probability at the mode of the binomial distribution with p_{col} . For $m = 3$ we find $\delta \geq 0.6$ for the student election.

5.2 Coercion-Resistance

In [6] the authors present a definition of quantitative coercion-resistance following similar ideas as in Definition 1. We will here use their strategy version and not go into all details. We let S denote the election system with specified number candidates, honest (n_h) and dishonest voters (mostly neglected here) and a ballot distribution, and attacker, C_S , and voter, V_S , interactive Turing machine models. We let γ denote a property defining the goal of the coerced voter, e.g. to vote for a specified candidate.

Definition 2. S achieves δ^{cr} -coercion-resistance if for all dictated coerced strategies $\pi_{V_{co}} \in V_S$ there exists a counter-strategy $\tilde{\pi}_{V_{co}} \in V_S$ s.t. for all coercer programs $\pi_c \in C_S$:

- $\Pr[(\pi_c || \tilde{\pi}_{V_{co}} || \pi_v)^{(l)} \mapsto \gamma]$ is overwhelming.
- $\Pr[(\pi_c || \tilde{\pi}_{V_{co}} || \pi_v)^{(l)} \mapsto 1] - \Pr[(\pi_c || \tilde{\pi}_{V_{co}} || \pi_v)^{(l)} \mapsto 1]$ is δ^{cr} -bounded.

With bounded and overwhelming defined in the security parameter. The first part says that the voter is able to achieve her goal (e.g. vote for a specific candidate) and the second part says that the coercer's distinguishing power is bounded by δ^{cr} . This level of coercion-resistance depends on several parameters especially the probability distribution on the candidates.

Whereas this definition gives a level of coercion-resistance, it does not tell the full story. To see this let us consider two different election systems. System A outputs voter names and corresponding votes with probability $1/2$, completely breaking privacy, and otherwise it only outputs the election result. Neglecting the information from the election result we get $\delta^A = 1/2$. In system B the voter secretly gets a signed receipt of her vote with probability $1/2$ and otherwise the protocol works ideally. In this case a coerced voter can always cast her own choice and claim that no receipt was received. A voter following the coercer's instruction will with probability $1/2$ give the corresponding receipt, i.e. we again have $\delta^B = 1/2$. However, the two systems are very different from the point of view of the voter: in system A the coerced voter gets caught cheating with probability $1/2$, whereas in system B, the voter always has plausible deniability.

Since plausible deniability is an essential factor for the usability of coercion-resistance mechanisms, we need a new definition to be able to measure this aspect.

5.3 No Deniability

The level of plausibility of a voter claiming to have followed the coercer, while actually following the counter strategy, relates to the probability of false positives when the coercer tries to determine if the voter disregarded the instructions. In the following we assume without loss of generality that the coercer outputs 1 when blaming the voter. We now want to define the maximal probability of getting caught without any deniability, i.e. we consider the case where $\Pr[(\pi_c || \pi_{V_{co}} || \pi_v)^{(l)} \mapsto 1] = 0$ or negligible, i.e. the coercer only uses strategies where he never blames an honest voter.

Definition 3. *S achieves $\delta^{cr, no-d}$ -coercion-resistance if for all dictated coerced strategies $\pi_{V_{co}} \in V_S$ there exists a counter-strategy $\tilde{\pi}_{V_{co}} \in V_S$ s.t. for all coercer programs $\pi_c \in C_S$:*

- $\Pr[(\pi_c || \tilde{\pi}_{V_{co}} || \pi_v)^{(l)} \mapsto \gamma]$ is overwhelming.
- $\Pr[(\pi_c || \tilde{\pi}_{V_{co}} || \pi_v)^{(l)} \mapsto 1]$ is $\delta^{cr, no-d}$ -bounded and $\Pr[(\pi_c || \pi_{V_{co}} || \pi_v)^{(l)} \mapsto 1]$ is negligible.

Note that the coercer's optimal strategy to obtain this $\delta^{cr, no-d}$ and the voter's strategy might be different from the ones in Definition 2 but $\delta^{cr, no-d} \leq \delta^{cr}$.

The no deniability probability clearly separates the RLT approaches. The original RLT always has plausible deniability if we choose to keep some ratio of ballots shrouded and the voter can claim her ballot was not revealed. This is e.g. important for RLV giving deniability against an attack where the coercer provides a ciphertext to cast and asks for its decrypted vote.

In the case of masked ballots, there can be a chance of getting caught undeniably. This will depend strongly on the number of revealed ballot components m , the vote distribution and the voter's goal. For the student election analysed above, the worst case when the goal of the voter is to cast $(1, 0, 0, 0, 0)$. The coercer's optimal strategy is then to demand a vote for $(0, 1, 1, 1, 1)$. The coercer

will blame the voter if there is no matching masked ballot, i.e. if no honest voters produce a collision which happens with probability $(1 - p_{col})^{n_h+1}$ computed Fig. 2. The probability of no deniability is then $p = 8 \cdot 10^{-9}$ for $m = 2$ but jumps abruptly to $p = 0.6$ for $m = 3$.

An interesting case is when the voter has a relaxed goal allowing to cast a signature part or not, and when the vote distribution has some ballots strictly zero probability. Let us consider a three candidate 0/1 election with 1-vote probabilities $(1/2, 1/2, 0)$. The voter's goal is to cast a 1 for the first candidate. The coercer's optimal strategy is to demand a signature ballot $(0, 0, 1)$. The voter has two counter-strategies: 1) cast a vote $(1, 0, 0)$ without the 0 probability signature part or 2) casting a vote $(1, 0, 1)$ with the signature part. For 1) there is no deniability if no other voter casts a matching ballot and the coerced voter's ballot does not match either. For $m = 1$ this happens with $p = (2/3)^{n_h+1}$ and for $m = 2$ with $p = (11/12)^{n_h}$, both are small if we have many voters. For 2) there will always be a matching vote if the first part of the coerced voter's ballot is masked. However, if the last part is revealed the coercer can deduce this ballot comes from the coerced voter since this candidate had probability 0, and if the 1 vote in the first part is revealed as well then the voter is caught with no deniability. Thus is no deniability with probability $1/3(2/3)^{n_h}$ for $m = 1$ and $1/3 + 1/3(11/12)^{n_h}$ for $m = 2$. Thus for $m = 1$ strategy 2) is always better, but for $m = 2$ strategy 1) is better when we have more than 13 voters. In some cases the voter strategy thus depends on m , which might not be know beforehand.

Finally, it is also natural to define the level of plausability we can provide. The average plausability that a voter has e.g. in Definition 2 is a useful quantity for the voter, but it would be more useful to guarantee that the voter always has a certain level for coercion-resistance. We leave a precise definition for future work.

5.4 Receipt-Freeness

Following [6], definition 2 also covers receipt-freeness. However, we again argue that modelling some variants is useful. The following definition is based on a swap of $\pi_{V_{co}}$ and $\tilde{\pi}_{V_{co}}$ in Definition 3, and models vote buyers who do not want to pay a "free lunch" to vote sellers who follow their own goal. The voter goal γ can here be to cast a specified vote or set of votes.

Definition 4 (Weak Vote Buying Resistance). *For a given small p_H , S achieves δ^{wvb} -coercion-resistance if for all dictated coerced strategies $\pi_{V_{co}} \in V_S$ there exists a counter-strategy $\tilde{\pi}_{V_{co}} \in V_S$ s.t. for all coercer programs $\pi_c \in C_S$:*

- $\Pr[(\pi_c | \tilde{\pi}_{V_{co}} | \pi_v)^{(l)} \mapsto \gamma]$ is overwhelming.
- $\Pr[(\pi_c | \pi_{V_{co}} | \pi_v)^{(l)} \mapsto 1] - \Pr[(\pi_c | \tilde{\pi}_{V_{co}} | \pi_v)^{(l)} \mapsto 1]$ is δ^{wvb} -bounded and $\Pr[(\pi_c | \tilde{\pi}_{V_{co}} | \pi_v)^{(l)} \mapsto 1]$ is p_H -bounded.

We here interpret outputting "1" as paying the vote seller and this definition bounds how often an instruction-following vote seller gets paid by a vote-buyer (by $\delta^{wvb} + p_H$), but under the condition that a voter who casts another vote

is only paid with a (very) small probability p_{H} . This is a weakened vote-buyer model but interesting since a vote buyer should avoid vote sellers going for a “free lunch”. If the probability of an honest vote seller getting paid is low, it would help curb vote selling (even though the vote buyer could increase the price and create a “vote selling lottery”). In this definition, it also makes sense to drop the quantification over the coercer’s strategies to see the resistance to vote buying for different vote choices.

RLT In the original RLT a signature ballot will get revealed with probability $1 - f_{\text{blind}}$. If the vote buyer sees this he can pay the vote seller and will only pay the voter seller wrongly with a small probability p_{H} equal to the probability that one of the honest voters cast the signature ballot, i.e. $\delta^{vb} \simeq 1 - f_{\text{blind}}$ which can be rather high and protects badly against vote buying.

Masked RLT For the masked ballots we can however choose m such that several ballots will have the same masking as the signature ballot and makes it hard for the vote buyer to assess if the signature ballot was cast. For the student election we see from Fig. 2 that the number of matches with the optimal signature ballot $(0, 1, 1, 1, 1)$ is binomially distributed with an expectation value of 18.4 colliding ballots and a standard deviation of around 4.

For a more precise example, we can consider the three candidate election with probabilities $(1/2, 1/2, 0)$ as above and assume that the goal of the voter is to cast 0 for candidate 1 and $p_{\text{H}} = 0$. For $m = 1$ we will have $\delta^{vb} = 0$, but for $m = 2$ the vote-buyer can demand a vote for candidate 1 and 3 and pay out if he sees $(1, *, 1)$. Any counter-strategy with 0 for candidate 1 gives $\delta^{vb} = 1/3$.

We note that the new quantitative definitions for no deniability coercion-resistance (Def. 3), the weak vote buying resistance (Def. 4) and the original δ^{cr} -coercion-resistance (Def. 2) are considering different aspects of coercion-resistance and stating the three different δ -values gives a more nuanced description of the security of a given voting protocol. Also note that the δ values are calculated using potentially different strategies for the coercer and voter, and finding unified strategies optimising the parameters is an interesting line of future work. Finally, there are natural, more fine-grained, definitions extending these which should be also considered in the future.

6 Conclusion

We have shown that the idea of risk-limiting tallies and risk-limiting verification can be applied effectively to complex ballots. By partially masking each ballot rather than simply masking a subset of the ballots as in the original RLT and RLV we gain far greater flexibility in terms of masking strategies. This will be explored further in order to optimise the trade-offs between the various measures defined here in future work.

The approach is more robust against any claims of being undemocratic: all ballots are counted, and indeed in the full tally/partial verification option, all are counted fully. The only compromise then is some reduction in the level of verifiability, but this can be adjusted and is probably acceptable. If we compare this with ThreeBallot, there the chance of detecting a manipulated ballot is $1/3$, assuming that the attacker does not learn which ballot was retained by the voter. In our case we can achieve a good level of coercion mitigation with say a shrouding of $\pm 1/2$ of each ballot. Finally, we did a preliminary analysis of the quantitative privacy for the different tally methods, and the coercion-resistance, in particular, the probability a coerced voter gets undeniably caught. The new masked tallies however, are more appropriate for receipt-freeness, in particular with upper bounds on the number of vote sellers, whereas the old RLT provides good plausible deniability to coerced voters. This suggests combining both methods when possible, but future work is needed to define the precise level of vote-buying resistance.

Acknowledgements

This research was funded in part by the Luxembourg National Research Fund (FNR) grant references STV C18/IS/12685695, Q-CoDe CORE17/IS/11689058 and PRIDE15/10621687/ SPsquared.

References

1. Benaloh, J., Jones, D.W., Lazarus, E., Lindeman, M., Stark, P.B.: SOBA: Secrecy-preserving observable ballot-level audit. *EVT/WOTE* **11** (2011)
2. Benaloh, J., Stark, P.B., Teague, V.: VAULT: Verifiable audits using limited transparency. *E-Vote-ID 2019* p. 69 (2019)
3. Bernhard, D., Cortier, V., Pereira, O., Warinschi, B.: Measuring vote privacy, revisited. In: *Proceedings of the 2012 ACM conference on Computer and communications security*. pp. 941–952 (2012)
4. Jamroga, W., Roenne, P.B., Ryan, P.Y., Stark, P.B.: Risk-limiting tallies. In: *International Joint Conference on Electronic Voting*. pp. 183–199. Springer (2019)
5. Küsters, R., Truderung, T., Vogt, A.: Verifiability, privacy, and coercion-resistance: New insights from a case study. In: *2011 IEEE Symposium on Security and Privacy*. pp. 538–553. IEEE (2011)
6. Küsters, R., Truderung, T., Vogt, A.: A game-based definition of coercion resistance and its applications. *J. Comput. Secur.* **20**(6), 709–764 (2012). <https://doi.org/10.3233/JCS-2012-0444>
7. Liedtke, J., Küsters, R., Müller, J., Rausch, D., Vogt, A.: Ordinos: A verifiable tally-hiding electronic voting protocol. In: *IEEE 5th European Symposium on Security and Privacy (EuroS&P 2020)* (2020)
8. Ryan, P.Y.A., Rønne, P.B., Iovino, V.: Selene: Voting with transparent verifiability and coercion-mitigation. In: *International Conference on Financial Cryptography and Data Security*. pp. 176–192. Springer (2016)

Extending the Tally-Hiding Ordinos System: Implementations for Borda, Hare-Niemeyer, Condorcet, and Instant-Runoff Voting*

Fabian Hertel¹, Nicolas Huber², Jonas Kittelberger³, Ralf Küsters², Julian Liedtke², and Daniel Rausch²

¹ University of Stuttgart `st151599@stud.uni-stuttgart.de`

² University of Stuttgart `{firstname.lastname}@sec.uni-stuttgart.de`

³ University of Stuttgart `jonas.kittelberger@gmail.com`

Abstract. Modern electronic voting systems (e-voting systems) are designed to achieve a variety of security properties, such as verifiability, accountability, and vote privacy. Some of these systems aim at so-called *tally-hiding*: they compute the election result, according to some result function, like the winner of the election, without revealing any other information to any party. In particular, if desired, they neither reveal the full tally consisting of all (aggregated or even individual) votes nor parts of it, except for the election result, according to the result function. Tally-hiding systems offer many attractive features, such as strong privacy guarantees both for voters and for candidates, and protection against Italian attacks. The Ordinos system is a recent provably secure framework for accountable tally-hiding e-voting that extends Helios and can be instantiated for various election methods and election result functions. So far, practical instantiations and implementations for only rather simple result functions (e.g., computing the k best candidates) and single/multi-vote elections have been developed for Ordinos.

In this paper, we propose and implement several new Ordinos instantiations in order to support Borda voting, the Hare-Niemeyer method for proportional representation, multiple Condorcet methods, and Instant-Runoff Voting. Our instantiations, which are based on suitable secure multi-party computation (MPC) components, offer the first tally-hiding implementations for these voting methods. To evaluate the practicality of our MPC components and the resulting e-voting systems, we provide extensive benchmarks for all our implementations.

Keywords: E-Voting · Tally-Hiding · MPC · Accountability · Privacy · Implementations · Benchmarks.

1 Introduction

There is a multitude of different voting methods ranging from relatively simple ones, such as plurality/single-choice voting, to more complex ones, such as

* This work was in part funded by the Deutsche Forschungsgemeinschaft (DFG) KU 1434/11-1 and the Center for Integrated Quantum Science and Technology (IQST).

cumulative voting with multiple votes as well as preferential elections and multi-round votings. Also, there are many different result functions used in elections. For example, one might be interested only in the winner of the election (e.g., for presidential elections), the number of seats of parties in a parliament, or the k best or worst candidates (ranked or not ranked), e.g., to fill positions or to decide who moves on to a runoff election.

Tally-Hiding. A desirable and strong security property that several e-voting systems try to achieve is *tally-hiding* [1,2,3,4,5,6,7]. A tally-hiding system computes and publishes the election result, according to some result function, e.g., the winner of an election, without revealing any other information to any party. In particular, if desired, except for the election result itself, they neither reveal the full tally consisting of all (aggregated or even individual) votes nor parts of it, such as the winner of an election round or the number of votes of a candidate. Even internal parties, like trustees, should not learn anything besides the result. In essence, tally-hiding is a strong form of privacy that not just avoids leaking the content of individual ballots but rather avoids leaking any unnecessary information altogether. As discussed, e.g., in [6], tally-hiding is an attractive feature in many situations: it prevents introducing biases in voters during multi-round elections, losing candidates are not unnecessarily embarrassed due to a (potentially very low) number of votes, mandates of winning candidates remain strong even if they won only by a small margin, tally-hiding helps prevent gerrymandering since the exact vote distributions remain hidden, and it also prevents Italian attacks. To retain trust in the overall result, tally-hiding elections, like other elections, have to provide *verifiability*: Each voter must be able to verify that her vote was counted correctly and that the overall result is correct. Moreover, it should not only be possible to verify the result, but, if verification fails, it should be possible to identify misbehaving parties and hold them accountable for the failure. This stronger form of verifiability is called *accountability* [8].

There are also several systems that achieve what we call *partial tally-hiding*, e.g., [9,10,11,12,13,14]. These systems generally focus on solving specific issues, most notably Italian attacks, and achieve this by hiding only those parts of the tally that are critical for the issue at hand, e.g., the individual votes. However, they still reveal certain information besides the election result, e.g., the losers of intermediate election rounds. In this work, we focus on (full) tally-hiding where nothing but the final result is revealed.

Current State. As mentioned, several e-voting systems have been designed to be tally-hiding. These systems generally follow the same underlying idea, namely, using a publicly verifiable secure multi-party computation (MPC) protocol to compute the election result from an encrypted tally. From a theoretical point of view, it is clear that essentially arbitrary functions, and thus election results, can be computed in this way. The main challenge lies in constructing an efficient MPC tallying component. For example, in recent work Cortier et al. [7] tackles, among others, this challenge by proposing tally-hiding MPC components (for single-vote elections, majority judgement, Condorcet-Schulze, and STV) and studying their asymptotic complexity.

So far, there are only very few (fully) tally-hiding protocols that have been implemented, benchmarked, and shown to be viable. Specifically, Canard et. al. [5] proposed and implemented a tally-hiding protocol for majority judgement that is shown to achieve practical performance. In [6], Küsters et. al. proposed the general Ordinos framework for provably secure accountable tally-hiding e-voting. They also designed and implemented several Ordinos instantiations and demonstrated their practicality. Specifically, they considered the following highly relevant but relatively simple result functions for single/multi-vote elections: computing the k candidates with the highest/lowest number of votes, computing all candidates that pass a certain threshold of votes, a combination of both, with or without revealing the ranking among the winners, and with or without revealing the number of votes the candidates in question have obtained.

Our Goal. In this work, we want to extend the state-of-the-art by implementing and benchmarking MPC components for tally-hiding elections also for many other voting methods. To this end, we build on the Ordinos system, since, as mentioned, Ordinos provides a general provably secure framework for accountable (and hence, verifiable) tally-hiding elections, and because we can base our work on the practical instantiations of Ordinos that have been proposed before.

Our Contributions. We propose and implement several new instantiations of Ordinos for complex election types and result functions. Specifically, we propose MPC components for Borda voting, the Hare-Niemeyer method for proportional representation, Instant-Runoff Voting, and multiple versions of Condorcet (plain Condorcet, weak Condorcet, Copeland evaluation, Minimax evaluation, Smith set, and Schulze evaluation). As we explain, our MPC components for tallying satisfy the requirements of the Ordinos framework and therefore yield provably secure e-voting systems, i.e., they inherit the accountability, privacy, and tally-hiding properties of the Ordinos framework.

Our implementations of the MPC components are available at [15]. We accurately assess the performance and scalability of our MPC components for practical applications. While our algorithms do not asymptotically improve over naturally expected baselines (e.g., IRV performs exponentially in the number of candidates), which was not the main goal of this work anyways, we are indeed able to show that the concrete performance is practical for real world elections (in the case of IRV and Schulze only for relatively small numbers of candidates).

Structure. In Section 2 we recall the Ordinos framework. We then, in Section 3, present and construct important building blocks used in subsequent sections. In Sections 4 to 7, we present our instantiations, implementations, and evaluations for the various voting methods we consider. We conclude in Section 8.

2 The Ordinos Framework

We need the following notation and terminology. We write $[n]$ to denote the set $\{0, \dots, n-1\}$. Let n_c be the number of candidates/choices on a ballot and let n_v be the (maximal) number of voters. The format of a plain ballot is defined via a

finite *choice space* $\mathbb{C} \subseteq \mathbb{N}^{n_c}$, i.e., a ballot assigns each candidate/choice a number subject to constraints defined by \mathbb{C} . For example, a single vote election where a plain ballot contains one vote for a single candidate/choice can be modeled via the choice space $\mathbb{C}_{\text{single}} := \{(b_0, \dots, b_{n_c-1}) \in \{0, 1\}^{n_c} \mid \sum_i b_i = 1\}$. For voter j we denote her plain ballot by $v^j := (v_i^j)_{i \in [n_c]} \in \mathbb{C}$. Ordinos uses an additively homomorphic t -out-of- n_t threshold⁴ public key encryption scheme $\mathcal{E} = (E, D)$ with $E_{\text{pk}}(a)$ denoting a ciphertext obtained as an encryption of plaintext a under the public key pk of the election.

Given this terminology, Ordinos [6] works roughly as follows. The protocol is run among a voting authority, the voters, n_t trustees, an authentication server, and an append-only bulletin board (BB). In the *setup phase*, parameters of the election are generated, including a public key and corresponding secret key shares for \mathcal{E} , one for each trustee, along with a NIZKP $\pi^{\text{KeyShareGen}}$ from each trustee to prove knowledge of their key share. Additionally, \mathbb{C} and the result function f_{res} of the election (see below) are fixed and published. In the *voting phase*, the voters first encrypt their ballots and then publish them on the BB, authenticating themselves as eligible voters with the help of the authentication server. An encrypted ballot of voter j has the form $(E_{\text{pk}}(v_i^j))_{i \in [n_c]}$, i.e., each component of the plain ballot is encrypted separately. The encrypted ballot also contains a NIZKP π^{Enc} that proves validity of the plain ballot, i.e., $v^j = (v_i^j)_{i \in [n_c]} \in \mathbb{C}$. The published encrypted ballots can then be (publicly) homomorphically aggregated to obtain the encrypted and aggregated full tally, i.e., one obtains ciphertexts for $v_i := \sum_{j \in [n_v]} v_i^j$ where v_i is the total number of votes/points that candidate/choice i obtained in the election. In the *tallying phase*, the trustees run a publicly accountable MPC protocol \mathbb{P}_{MPC} to compute f_{res} . This protocol takes as (secret) inputs the secret key shares of the trustees and the (public) encrypted aggregated tally and outputs $f_{\text{res}}(v_0, \dots, v_{n_c-1})$. This result, along with any material that is needed to verify the MPC computation, is published by the trustees on the BB. Finally, in the *verification phase*, voters can check that their ballots appear on the BB and everyone can verify the result by checking all NIZKPs as well as the (accountable) MPC computation.

Security of Ordinos (privacy and accountability) was shown independently of specific instantiations of the mentioned primitives, and hence, security is guaranteed by any instantiation fulfilling the necessary requirements. In what follows, we briefly recall the two generic security results of Ordinos (including the requirements for the underlying primitives), which have been formalized and proven in [6]. The first result states accountability of Ordinos, where accountability was formalized using the *KTV framework* [8].

Theorem 1 (Accountability [6], informal). *Let \mathcal{E} be a correct additively homomorphic threshold public-key encryption scheme \mathcal{E} , $\pi^{\text{KeyShareGen}}$ and π^{Enc} be secure NIZKPs for \mathcal{E} , and \mathbb{P}_{MPC} be a publicly accountable MPC protocol, i.e., if the result does not correspond to the input, then this can be detected and at least one misbehaving trustee can be identified; this must hold true even if all*

⁴ I.e., there are n_t secret key shares with $t \leq n_t$ secret shares being necessary for successful decryption.

trustees running the MPC protocol are malicious. Then (the resulting instance of) Ordinos is accountable.⁵

Importantly, Ordinos provides accountability (and hence, by results in [8] also verifiability) even if all trustees are malicious.

The following theorem (that was formalized and proven in [6]) states privacy of Ordinos, i.e., the tally-hiding property that no information besides the final result, according to the result function, is revealed to anyone, including the trustees. It was proven using the privacy definition given in [16].

Theorem 2 (Privacy/Tally Hiding [6], informal). *Let \mathcal{E} be an additively homomorphic IND-CPA-secure t -out-of- n_t threshold public-key encryption scheme, $\pi^{\text{KeyShareGen}}$ and π^{Enc} be secure NIZKPs for \mathcal{E} , and let P_{MPC} be an MPC protocol that securely realizes (in the sense of UC [17,18]) an ideal MPC functionality which essentially takes as input a vector of ciphertexts and returns f_{tally} evaluated on the corresponding plaintexts without leaking any other information if at most $t - 1$ trustees are malicious. Then (the resulting instance of) Ordinos provides privacy/is tally-hiding in presence of up to $t - 1$ malicious trustees.*

Instantiations of Ordinos. As mentioned in the introduction, for practical instantiations of Ordinos the main challenge lies in finding efficient and suitable instantiations of the primitives, including the MPC component, that work well and efficiently together. For certain kinds of elections and result functions this has been achieved by Küsters et al. in [6]. These instantiations use a threshold variant of the Paillier encryption scheme [19] to implement \mathcal{E} . To design their MPC protocols P_{MPC} for their result functions, Küsters et al. make use of and combine NIZKPs and publicly accountable MPC protocols from the literature that implement the following basic operations:

- $E_{\text{pk}}(c) = f_{\text{add}}(E_{\text{pk}}(a), E_{\text{pk}}(b))$ s.t. $c = a + b$, directly from the additive homomorphic property of Paillier encryption; for brevity we write $E_{\text{pk}}(a) + E_{\text{pk}}(b)$. Similarly, $E_{\text{pk}}(c) = f_{\text{mul}}(E_{\text{pk}}(a), b)$ s.t. $c = a \cdot b$; for brevity we write $E_{\text{pk}}(a) \cdot b$.
- $E_{\text{pk}}(c) = f_{\text{mul}}(E_{\text{pk}}(a), E_{\text{pk}}(b))$ s.t. $c = a \cdot b$, using a publicly accountable MPC protocol for multiplication [19]; for brevity we write $E_{\text{pk}}(a) \cdot E_{\text{pk}}(b)$.
- $E_{\text{pk}}(c) = f_{\text{gt}}(E_{\text{pk}}(a), E_{\text{pk}}(b))$ s.t. $c = 1$ iff $a \geq b$ and 0 otherwise, using a publicly accountable MPC protocol for the greater-than test [20].
- $E_{\text{pk}}(c) = f_{\text{eq}}(E_{\text{pk}}(a), E_{\text{pk}}(b))$ s.t. $c = 1$ iff $a = b$ and 0 otherwise, using a publicly accountable MPC protocol for equality tests from [20].
- $c = f_{\text{dec}}(E_{\text{pk}}(a))$ s.t. $E_{\text{pk}}(a)$ is an encryption of c , using publicly accountable distributed Paillier decryption [19].

The above components have been chosen not only because they meet the necessary security requirements but also due to their efficiency, which facilitates constructing practical instantiations. That is, f_{add} and multiplication with a

⁵ We note that the security proof for accountability (and also for privacy) makes certain standard assumptions, such as honesty of the BB. We refer interested readers to [6] for full details. We also note that if P_{MPC} provides only public verifiability, instead of public accountability, then Ordinos provides verifiability.

publicly known value can be computed locally for the Paillier scheme. Furthermore, both f_{gt} and f_{eq} as proposed by [20] run in sublinear time independently of the actual plaintext space of the encryption scheme if plaintexts contained within the ciphertexts are upper bounded by some bound b_{ct} . Ordinos indeed has this property, where the bound generally depends on n_v and C . Furthermore, both f_{gt} and f_{eq} and Paillier synergize rather well. As discussed in [6], while f_{gt} and f_{eq} can in principle also be used with exponential ElGamal, both functions use decryption for a (upper-bounded but still) relatively large plaintext space, and hence, would perform poorly with exponential ElGamal.

We note that the above components have a useful property, namely, they can be combined to compute more complex functions such that the resulting protocol is still a secure publicly accountable MPC protocol. In other words, they allow for building protocols P_{MPC} for Ordinos that meet the requirements of Theorems 1 and 2.

Our Instantiations and Parameters. In this work, we use Paillier encryption and the above basic building blocks. The main challenge and indeed a core contribution of our paper is to show and empirically demonstrate that these components are not just suitable for constructing protocols P_{MPC} for simple result functions (e.g., revealing the candidate with the most votes in a single-vote election), but also for much more complex voting methods and result functions. To benchmark our implementations, we use the parameters as [6]. That is, we use a Paillier key of size 2048 bits and for the greater-than and equality protocols we use the range $[2^{16}]$, i.e., $b_{\text{ct}} = 2^{16}$, for the (encrypted) plaintext inputs. This range can be increased if needed, i.e., to account for cases where aggregated ciphertexts might contain plaintexts outside of $[2^{16}]$. Note that, except for requiring a suitable upper bound b_{ct} , the performance of our MPC protocols is otherwise independent of the exact number of voters n_v due to aggregation of the ballots. The setup for our benchmarks consists of three trustees communicating over a local network. Each trustee ran on an ESPRIMO Q957 (64bit, i5-7500T CPU @ 2.70GHz, 16 GB RAM). As in [6], the benchmarks of our MPC protocols start with an already aggregated tally. Küsters et al. [6] showed for their MPC protocols that the number of trustees does not influence the benchmarks in a noticeable way and that, due to the sublinear communication complexity of the comparison protocols, there is no significant difference between a local network and the Internet. Both results also hold for our MPC constructions which are based on the same primitives. Hence, our benchmarks focus on the number of candidates which is the main factor for the performance of our protocols.

3 Building Blocks

In this section, we describe three MPC building blocks that can be obtained using the basic operations described in Section 2 and which we use to construct P_{MPC} for our Ordinos instances, where the first building block is from [6].

Minimum k and Maximum k Values. Often, we have a vector $(E_{\text{pk}}(a_i))_{i \in [n]}$ and want to compute ciphertexts $(E_{\text{pk}}(b_i))_{i \in [n]}$ of a vector $(b_i)_{i \in [n]}$ such that

Floor Division	
Input:	$E_{\text{pk}}(a), b, n$
Result:	$E_{\text{pk}}(i)$ with $i \in [n]$ such that $i \cdot b \leq a$ and $(i + 1) \cdot b > a$
1	for $j \in [n + 1]$ do
2	$E_{\text{pk}}(r_j) = f_{\text{gt}}(E_{\text{pk}}(a), E_{\text{pk}}(j \cdot b))$
3	for $j \in [n]$ do
4	$E_{\text{pk}}(\hat{r}_j) = E_{\text{pk}}(r_j) - E_{\text{pk}}(r_{j+1})$
5	$E_{\text{pk}}(i) = \sum_{j \in [n]} E_{\text{pk}}(j) \cdot E_{\text{pk}}(\hat{r}_j)$
6	return $E_{\text{pk}}(i)$

Fig. 1: Algorithm for Floor Division.

$b_i = 1$ if a_i is one of the k largest (resp. smallest) values in $(a_i)_{i \in [n]}$ and $b_i = 0$ otherwise. We do so as described in [6]. That is, we first construct the lower halve of the comparison matrix M such that $M_{i,j < i} := f_{\text{gt}}(E_{\text{pk}}(a_i), E_{\text{pk}}(a_j))$. From this matrix, which consists of ciphertexts containing 0 or 1, one can compute a ciphertext for each a_i that contains the number of comparisons that i has won, i.e., where $a_i \geq a_j$ for some $j \neq i$. We can then use f_{gt} to compare this ciphertext (containing the results for a_i) with a ciphertext on the number $n - k - 1$ and obtain $E_{\text{pk}}(b_i)$.⁶ One can proceed similarly in order to find the smallest k values. Note that this algorithm can also be applied if k is not publicly known but rather only available as a ciphertext; in this situation, k is also not revealed by the algorithm. We make use of this property in the context of the Hare-Niemeyer method, see Section 4. We denote these algorithms for computing the vectors $E_{\text{pk}}(b_i)$ by `GetBest()`, resp. `GetWorst()`. These algorithms have runtime $\mathcal{O}(n^2)$.

Maximum. If we are just interested in obtaining a ciphertext $E_{\text{pk}}(a_i)$ of the maximum value a_i in the vector $(E_{\text{pk}}(a_i))_{i \in [n]}$, we can do so more efficiently in linear runtime. That is, we start with the possible maximum $m = E_{\text{pk}}(a_0)$ and iterate through all a_i 's. For each a_i we test whether it is greater than the current maximum with $g = f_{\text{gt}}(E_{\text{pk}}(a_i), m)$ and adapt the maximum accordingly with $m = g \cdot E_{\text{pk}}(a_i) + (E_{\text{pk}}(1) - g) \cdot m$. The minimum can be computed accordingly. We denote these algorithms by `GetMax()` resp. `GetMin()`. If we are interested in the indices of the values that are the maximum resp. minimum, we can first compute the encrypted maximum m and then compute for each index the encrypted indicator $E_{\text{pk}}(b_i) := f_{\text{eq}}(E_{\text{pk}}(a_i), m)$, $b_i \in \{0, 1\}$. We denote these algorithms for obtaining the tuple of encrypted indicators with `GetMaxIdx()` and `GetMinIdx()`.

Floor Division. Given a ciphertext $E_{\text{pk}}(a)$ of some $a \in \mathbb{N}$ and a plain value $b \in \mathbb{N}_{>1}$, this algorithm, described in Figure 1, is used to compute a ciphertext $E_{\text{pk}}(i)$ with $i = \lfloor \frac{a}{b} \rfloor$. The algorithm also requires a value $n \in \mathbb{N}$, s.t. $n \cdot b$ does not exceed the plaintext space size and $i \in [n]$. The algorithm compares all possible values. The sequence $(r_j)_{j \in [n+1]}$ consists of a sequence of zeros followed by a sequence of ones, where $r_j = 0$ if $a < j \cdot b$ and $r_j = 1$ otherwise. We are interested in the index i such that $r_i = 1$ and $r_{i+1} = 0$. We obtain this index by computing for each j the value $\hat{r}_j := r_j - r_{j+1}$. Then, we can use these \hat{r}_j as indicators to obtain the correct division result.

⁶ If there are multiple a_i with the same value, there might be more than k b_i that are 1. In cases where always exactly k such values are required, one can use a tie breaker mechanism such as the one described in [7].

4 Hare-Niemeyer Method

The Hare-Niemeyer method is an evaluation method for proportional allocation of seats that is used for example in Ukraine and Italy, but has also been used for German federal elections until 2005. The Hare-Niemeyer method is used for situations where a fixed number of seats needs to be assigned to candidates from different parties, where a voter typically votes only for the party and not the candidates themselves. Often, this type of proportional voting is also combined with some form of plurality or majority voting, such as first-pass-the-post-voting for electing single representatives for electoral districts, in so-called *mixed electoral systems*. Such mixed systems are also used for elections in many state parliaments in Germany, elections for the Scottish and Welsh parliaments and elections for the New Zealand House of Representatives. More specifically, the Hare-Niemeyer method for proportional voting works as follows: Assume that there are n_s seats to be assigned among n_c parties. Then, if there are a total of n_v valid votes and party c_i has received v_i votes, the number of seats that c_i is awarded is computed using the “ideal quota” given by $q_i := \frac{v_i \cdot n_s}{n_v}$. Initially, the number of seats awarded to each c_i is set to be $s'_i := \lfloor q_i \rfloor$. However, since these s'_i usually do not add up to n_s , the remaining $n_r \in [n_c]$ seats are distributed in the order of the highest remainders of $\frac{v_i \cdot n_s}{n_v}$. That is, the n_r parties c_i with the highest remainders $d_i = q_i - s'_i$ receive one additional seat each. Note that it could happen that multiple parties have the same remainders d_i , and thus, more than n_r additional seats are assigned. If this is not desired, then one would use a tie-breaking algorithm (cf. Section 5 and Footnote 6). There are many possible ways to vote in proportional elections. Our algorithm can handle every possible ballot format, as long as the ballots can be aggregated such that we obtain one ciphertext per party containing the total number of votes for the party. In the simplest case, one can use $\mathcal{C}_{\text{single}}$ as choice space with ballot format NIZKPs π^{Enc} from, for example, [21] and [19].

Our MPC algorithm for computing the Hare-Niemeyer method is presented in Figure 2. On a high-level, the algorithm follows the above description, i.e., it first computes the seat distribution without taking the remainder seats into account. Next, for each party, the remainder of the division (see above) is computed and the remainder seats are distributed among the parties with the highest remainder values. Importantly, this is achieved without revealing the total number of remainder seats or the set of parties that have received an additional seat.

We present benchmarks for our MPC tallying protocol in Figure 3. The runtime of the algorithm is linear in $n_c \cdot n_s$. As the figure shows, evaluating the Hare-Niemeyer method is highly efficient for a practical number of seats (1000) and (up to) 4 parties. Due to the linear growth, this should still be the case even if there are more parties than the maximum of 4 that we benchmarked. Also, recall from Section 2 that these benchmarks are essentially independent of the number of voters and trustees. In terms of security for our Ordinos instantiation, we obtain the following.

Theorem 3 (Security of Hare-Niemeyer method with Ordinos). *Let \mathcal{E} be an additively homomorphic IND-CPA-secure t -out-of- n_t threshold public-key*

Tally-Hiding Hare-Niemeyer Evaluation	
Input:	Encrypted aggregated votes per party: $\{E_{\text{pk}}(v_i)\}_{i \in [n_c]}$ Number of seats in total n_s and number of total votes n_v
Result:	Vector s such that s_i is the number of seats of party i .
1	for $i \in [n_c]$ do
2	$m_i = E_{\text{pk}}(v_i) \cdot n_s$
3	$E_{\text{pk}}(s'_i) = \text{FloorDivision}(m_i, n_v, n_s)$
4	$E_{\text{pk}}(n_r) = E_{\text{pk}}(n_s) - \sum_{i \in [n_c]} E_{\text{pk}}(s'_i)$
5	for $i \in [n_c]$ do
6	$E_{\text{pk}}(d_i) = E_{\text{pk}}(v_i) \cdot n_s - n_v \cdot E_{\text{pk}}(s'_i)$.
7	$(E_{\text{pk}}(d_i^{\text{best}}))_{i \in [n_c]} = \text{GetBest}((E_{\text{pk}}(d_0), \dots, E_{\text{pk}}(d_{n_c-1})), E_{\text{pk}}(n_r))$
8	for $i \in [n_c]$ do
9	$E_{\text{pk}}(s_i) = E_{\text{pk}}(s'_i) + E_{\text{pk}}(d_i^{\text{best}})$
10	$s_i = f_{\text{dec}}(E_{\text{pk}}(s_i))$
11	return s

Fig. 2: Tally-Hiding Hare-Niemeyer Evaluation

encryption scheme and $\pi^{\text{KeyShareGen}}$ be a secure NIZKP for \mathcal{E} such as, e.g., the primitives used in [6]. Let π^{Enc} be the ballot format NIZKP from above, and let P_{MPC} be our MPC component for the Hare-Niemeyer method as defined above. Then, the Ordinos instance using these primitives is an accountable and private (and hence tally-hiding) voting system for the Hare-Niemeyer method.

Proof Sketch. This theorem is a direct corollary of Theorems 1 and 2 which were proven in [6]. Observe that the primitives \mathcal{E} , $\pi^{\text{KeyShareGen}}$, and π^{Enc} already fulfill the requirements of Theorems 1 and 2. The only thing left to show for Theorems 1 and 2 is that our new tallying protocol P_{MPC} is secure. That is, we have to show that P_{MPC} is a private and publicly accountable implementation of the Hare-Niemeyer method.

Both properties follow because our MPC protocol is built from combinations of the basic components presented in Section 2. As mentioned in that section, each of these basic components guarantees privacy and public accountability. As for the connections of these components, the respective inputs and outputs are all encrypted (except for the final decryption of the election result) and published on the BB. Due to the encryption, these intermediate results do not leak any additional information, neither to internal parties nor to external observers. Also, since the intermediate results are published, external observers can check that the output of one step is used correctly as the input to the next step. Thus, if some trustee tries to use a different input, she can be held accountable. \square

5 Instant Runoff Voting (IRV)

Instant-runoff-voting (IRV) is a ranked voting method which can be used in single-seat elections. It is often used, e.g., in Australia, India, the UK and the US. In IRV, if a candidate has been ranked first by an absolute majority of voters, this candidate is the winner of the election. Otherwise, the candidate ranked first least often is eliminated, i.e., removed from the pool of candidates. Then, all ballots are adjusted accordingly, i.e., the eliminated candidate is removed and other (lower-ranked) candidates are moved up a rank. This process

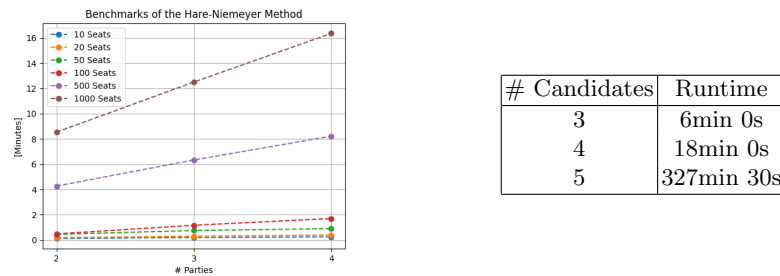


Fig. 3: Benchmarks for the Hare-Niemeyer method (left) and IRV (right).

is repeated until one of the remaining candidates has received the absolute majority of votes and thus wins the election. An algorithm for evaluating IRV in a fully tally-hiding way has already been proposed in [3]. However, this algorithm does not support aggregation and therefore scales with the number of ballots/voters. Hence, instead of building on and providing the first implementations and benchmarks of this algorithm, we rather follow a different approach: we propose an algorithm that is compatible with the aggregation approach of Ordinos. By supporting aggregation, the performance of our solution remains essentially independent of the number of voters. For our instantiation, we use C_{single} but interpret each choice as a ranking of candidates. For example, for $n_{\text{cand}} = 5$, we have $n_c = n_{\text{cand}}! = 120$ choices, where each choice represents a permutation of the set of candidates. Observe that this encoding indeed allows for aggregating IRV ballots to obtain the full (encrypted) tally as usually done in Ordinos. NIZKPs π^{Enc} for showing the well-formedness of such a ballot are given in [21] and [19]. Note that the size of this choice space (and thus the runtime of our algorithm) scales exponentially in the number of candidates. However, we are able to show that this approach is still practical for a small amount of candidates (≤ 5) as they have occurred in practice (see benchmarks presented in Figure 3 and the discussion below).

We present our algorithm to evaluate an IRV election with Ordinos in Figure 4. The idea of our algorithm is that in round i , i.e. after i candidates have been eliminated, we have to consider the first $k = i + 1$ candidates of each ballot to find a candidate that has not been eliminated. We can then look at each possible ordering r_i of k candidates and check how many votes every permutation that starts with r_i received. These votes are then assigned to the respective first non-eliminated candidate in that permutation and the candidate with the least votes is eliminated. Note that it can happen that two candidates are assigned the same (lowest) number of votes in a round. Typically, IRV does not eliminate multiple candidates in the same round, hence in these situations some kind of tie-breaking algorithm is required. Often, this is done by lot - for example, this is the default method for IRV elections in Maine [22]. We address this issue, by letting $\text{GetMinIdx}()$ output only the first candidate (i.e., the lower index) with the least amount of votes. To obtain randomized tie-breaking, one starts with

Tally-Hiding IRV Evaluation	
<pre> 1 X = (E_{pk}(0))_{i ∈ [n_{cand}]}</pre>	// Encrypted indicator bits.
Input: $n_{cand}, (v_j)_{j \in n_c}$, the aggregated single-vote ballots for the choices. Result: An indicator vector $(b_i)_{i \in n_{cand}}$ such that $b_i = 1$ iff i -th candidate is eliminated.	
<pre> 2 for i ∈ [n_{cand} - 1] do 3 (v_j^s = E_{pk}(0))_{j ∈ [n_{cand}]}</pre>	<pre> // perform n_{cand} - 1 elimination rounds // Votes received in this round.</pre>
<pre> 4 k = i + 1 5 for (ordered) k-tuple r_i with entries in [n_{cand}] do</pre>	<pre> // go over ranking prefixes</pre>
<pre> 6 c_f = E_{pk}(0), d = E_{pk}(0)</pre>	<pre> // c_f will be the winner of prefix, d is a helper bit</pre>
<pre> 7 for c in r_i do</pre>	<pre> // find winner in prefix</pre>
<pre> 8 c_f = d · c_f + (1 - d) · c, d = d + (1 - d) · (1 - X_c)</pre>	
<pre> 9 for c in r_i do</pre>	<pre> // add points from ballots for current prefix to the winner</pre>
<pre> 10 b = f_{eq}(E_{pk}(c), c_f)</pre>	
<pre> 11 for j ∈ [n_c] s.t. j represents a ranking where the top k candidates are r_i do</pre>	
<pre> 12 v_c^s = v_c^s + b · v_j</pre>	
<pre> 13 (e_j)_{j ∈ [n_{cand}]} ← GetMinIdx((v₀^s, ..., v_{n_{cand}-1}^s))</pre>	
<pre> 14 for r ∈ [n_{cand}] do</pre>	<pre> // Update/add one eliminated candidate</pre>
<pre> 15 X_r = X_r + (1 - X_r) · e_r</pre>	
<pre> 16 return f_{dec}(X)</pre>	

Fig. 4: Tally-Hiding IRV Evaluation.

a uniformly randomly ordered list of candidates. It is interesting future work to explore implementations of more sophisticated tie-breaking algorithms.

We provide benchmarks for our IRV algorithm in Figure 3. Due to the encoding of IRV ballots as permutations of $[n_{cand}]$, the algorithm has runtime $\mathcal{O}(n_{cand}!)$. But as can be seen in Figure 3, for small numbers of candidates the evaluation is still feasible. Indeed, 5 candidates is already a realistic scenario for real world IRV elections. E.g., in the 2015 New South Wales state election [23], which, however, uses a different IRV instance than we consider here, most electoral districts had 5 or less candidates. Using the properties of our basic building blocks described in Section 2, one can check that our IRV algorithm does not leak information. By the same reasoning as for Theorem 3 we obtain:

Theorem 4 (Security of Instant-Runoff voting with Ordinos). *Let \mathcal{E} and $\pi^{\text{KeyShareGen}}$ be as for Theorem 3. Let π^{Enc} be the NIZKP from above, and let P_{MPC} be our MPC component for the Instant-Runoff voting as defined in this section. Then, the Ordinos instance using these primitives is an accountable and private (and hence tally-hiding) voting system for Instant-Runoff voting.*

6 Condorcet methods

Condorcet is a ranked voting method that aims to determine a so-called *Condorcet winner*, i.e., a candidate that would beat all other candidates in a pairwise runoff election (we will call these pairwise runoff elections *comparisons*). It might happen that no candidate exists that wins all comparisons. There are several variants of (plain) Condorcet that deal with this, i.e., they output the Condorcet winner if it exists but additionally define mechanisms for obtaining a winner (or a set of winning candidates) also in some cases where no Condorcet winner exists. We discuss certain variants and their applications in practice below. We represent Condorcet ballots (which specify a full ranking of n_{cand} candidates

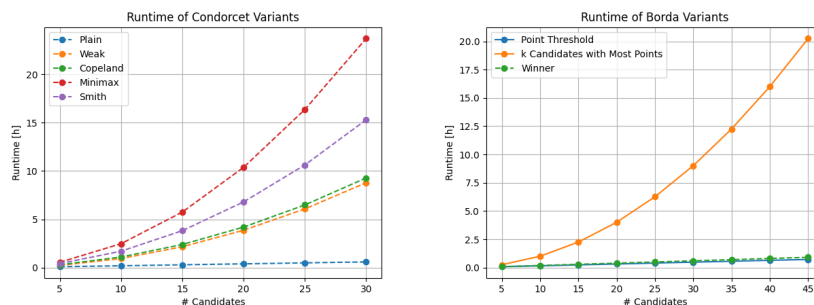


Fig. 5: Benchmarks for Condorcet voting (left) and benchmarks for Borda voting (right). The evaluation of the Schulze method for Condorcet took 135 minutes for 5 candidates and 9 days, 10 hours and 27 minutes for 20 candidates (not included in the figure).

without ties⁷) in Ordinos by interpreting them as a *comparison matrix*, i.e., an $(n_{cand} \times n_{cand})$ -matrix M , where $M_{ij} \in \{0, 1\}$ and $M_{ij} = 1$ means that a voter V prefers c_i over c_j . In order to obtain a choice space in the sense of definition of Section 2, we encode a comparison matrix as a vector of length $n_c = n_{cand}^2$ as expected way. Combined with some checks that ensure such a matrix indeed encodes a ranking (e.g., comparisons must be transitive), we obtain the choice space:

$$\mathcal{C}_{\text{Condorcet}} = \left\{ M \in \{0, 1\}^{n_{cand} \times n_{cand}} \mid \forall i, j, k \in [n_{cand}] : \right. \\ \left. i \neq j \implies M_{ij} + M_{ji} = 1 \wedge M_{ij} = M_{jk} = 1 \implies M_{ik} = 1 \right\}$$

We can use the NIZKP π^{Enc} presented in [9] for showing the well-formedness of such ballots. As usual, Ordinos aggregates all the comparison matrices of all voters, yielding (encryptions of) a matrix containing at entry (i, j) the total number of comparisons that c_i wins versus c_j . This is then used as input for the various Condorcet variants that (try to) compute a winner in different ways.

We have implemented MPC tallying protocols for several such Condorcet variants, with details provided below. The benchmarks of these algorithms are presented in Figure 5. Apart from the Schulze method, the runtime of the MPC components of all Condorcet versions grow quadratically in n_{cand} , as expected due to the nature of pairwise comparisons, but remain practical for reasonable numbers of candidates. (We note that the verification of the NIZKPs given in [9] requires runtime that is asymptotically cubic in the number of candidates but is not included/shown in the benchmarks.) Plain Condorcet in particular exhibits runtime that suggests practicality even for very large numbers of candidates. Also, recall that our benchmarks are essentially independent of n_v and n_t . With the same reasoning as for Theorem 3 we obtain:

Theorem 5 (Security of Condorcet voting with Ordinos). *Let \mathcal{E} and $\pi^{\text{KeyShareGen}}$ be as for Theorem 3. Let π^{Enc} be the NIZKP from above, and let*

⁷ Often, one allows for ties in Condorcet voting. However, in this work we do not consider this case.

P_{MPC} be one of our MPC components for a Condorcet voting method as defined below. Then, the Ordinos instance using these primitives is an accountable and private (and hence tally-hiding) voting system for that Condorcet method.

Next, we give details of the individual Condorcet variants and our corresponding MPC algorithms.

Plain Condorcet. We denote the vanilla Condorcet method, that outputs the unique Condorcet winner if and only if such a candidate exists, as *Plain Condorcet*. In Figure 6 (for bit $b = 1$), we present an algorithm for Plain Condorcet that is based on the building blocks described in Section 3. Note that, by choosing the bit $b = 0$ in Figure 6, the algorithm instead returns (encrypted) intermediate values, namely N , s^g and s'^g , which can be used for computing other Condorcet methods. Here, N denotes the *strict comparison matrix* that denotes in each entry $N_{i,j} \in \{0, 1\}$ whether c_i has won the majority of comparisons against c_j ($N_{i,j} = 1$) or won the same or less comparisons ($N_{i,j} = 0$). Additionally, for each candidate c_i , s_i^g denotes the number of comparisons that she has won or tied, while $s_i'^g$ only counts the winning comparisons.

Weak Condorcet. In this method all candidates that did not lose any comparisons (but that might be tied with other candidates and thus no Condorcet winners), i.e. all *weak Condorcet winners* are output. This method can be obtained via a straightforward extension of Figure 6 for $b = 0$. That is, for each c_i , compute and check whether $f_{\text{dec}}(f_{\text{eq}}(s_i^g, E_{\text{pk}}(n_{\text{cand}} - 1))) = 1$.

Copeland. This method, as opposed to the previous two methods, is guaranteed to output some winning candidate(s). To do so, it considers the wins and losses of each candidate in their comparisons and outputs all candidates with the most *Copeland points*, that is the highest difference between wins and losses. For $b = 0$, Figure 6 can be extended to first obtain the Copeland points of a candidate c_i via $E_{\text{pk}}(p_i) := E_{\text{pk}}(s_i'^g + s_i^g)$. We then compute the candidate with the most Copeland points with the `GetMaxIdx()` discussed in Section 3 and applying $f_{\text{dec}}()$.

Schulze Method. This method is more complicated than the previous ones and is very commonly used in practice (e.g., [24]). This method defines the score of candidate c_i 's comparison versus c_j to be the difference of the number of comparisons that c_i wins versus c_j minus the number of comparisons that c_j wins versus c_i . The candidates and the comparisons between them are considered as a directed weighted graph Γ , where the nodes of Γ represent the candidates and an arrow $c_i \rightarrow c_j$ is weighted with the score of c_i 's comparison versus c_j . Now, for any path p in Γ , we define the *value* of p as the lowest weight among the arrows involved in p . We then consider the *path value matrix* `PathMatrix`, an $(n_{\text{cand}} \times n_{\text{cand}})$ -matrix with entry `PathMatrixij` being the highest path value among paths from c_i to c_j . The Schulze method then outputs all candidates c_i such that `PathMatrixij` \geq `PathMatrixji` for each $j \in [n_{\text{cand}}]$. Note that the Schulze method is guaranteed to output some candidate(s). And if a unique Condorcet winner exists, then it will be returned by the Schulze method. The intuitive and probably most natural way to implement the Schulze method is to simply compute the standard algorithm while using MPC building blocks to implement all operations, which, for example, is also done in [7]. The main challenge lies

Condorcet Evaluation	
<pre> Input: Encrypted aggregated comparison matrix: $A := E_{pk}(M)$ $b \in \{0, 1\}$: indicator whether plain Condorcet should be evaluated. 1 $N = 0_{n_{cand} \times n_{cand}}, s^g = 0_{n_{cand}}, s'^g = 0_{n_{cand}}$ 2 for $i \in [n_{cand}]$ do 3 for $j \in [i + 1, n_{cand}]$ do 4 $g = f_{gt}(A_{i,j}, A_{j,i}), e = f_{eq}(A_{i,j}, A_{j,i}), g' = g - e$ 5 $N_{i,j} = g', N_{j,i} = E_{pk}(1) - g$ 6 $s_i^g = s_i^g + g, s_j^g = s_j^g + E_{pk}(1) - g', s_i'^g = s_i'^g + g', s_j'^g = s_j'^g + E_{pk}(1) - g$ 7 if $b = 1$ then 8 if $f_{dec}(f_{eq}(s_i'^g, E_{pk}(n_{cand}) - 1))$ then 9 return i 10 return N, s^g, s'^g </pre>	

Fig. 6: Condorcet Evaluation.

Condorcet: Schulze Evaluation	
<pre> Input: Encrypted aggregated comparison matrix: M Result: Vector $(b_i)_{i \in [n_{cand}]}$ such that $b_i = 1$ if c_i is a Schulze winner and $b_i = 0$ otherwise. 1 PathMatrix = $(E_{pk}(0))_{n_{cand} \times n_{cand}}$ 2 for $i \in [n_{cand}], j \in [n_{cand}] \setminus \{i\}$ do 3 PathMatrix$_{i,j} = M_{i,j} - M_{j,i}$ 4 for $i \in [n_{cand}], j \in [n_{cand}] \setminus \{i\}, k \in [n_{cand}] \setminus \{i, j\}$ do 5 $m = \text{GetMin}(\text{PathMatrix}_{j,i}, \text{PathMatrix}_{i,k})$ 6 PathMatrix$_{j,k} = \text{GetMax}(M_{j,k}, m)$ 7 MSchulze = $(E_{pk}(0))_{[n_{cand}] \times [n_{cand}]}$ 8 for $i \in [n_{cand}], j \in [i]$ do 9 $g = f_{gt}(\text{PathMatrix}_{i,j}, \text{PathMatrix}_{j,i})$ 10 $e = f_{eq}(\text{PathMatrix}_{i,j}, \text{PathMatrix}_{j,i})$ 11 $M_{i,j}^{\text{Schulze}} = g, M_{j,i}^{\text{Schulze}} = E_{pk}(1) - g + e$ 12 $b = (E_{pk}(0))_{n_{cand}}$ 13 for $i \in [n_{cand}]$ do 14 $w = \sum_{j \in [n_{cand}] \setminus \{i\}} M_{i,j}^{\text{Schulze}}$ 15 $b_i = f_{dec}(f_{eq}(w, E_{pk}(n_{cand}) - 1))$ 16 return $(b_i)_{i \in [n_{cand}]}$ </pre>	

Fig. 7: Condorcet: Schulze Evaluation.

in choosing suitable MPC building blocks such that the resulting tally-hiding Schulze algorithm performs well. Here we use the sublinear comparison protocols from Section 2, with the resulting algorithm presented in Figure 7.

Further Condorcet methods: We have also implemented and benchmarked the so-called *Smith set* and *Minmax* Condorcet methods. Intuitively, the smith set outputs a set of candidates such that each candidate from this set wins the comparisons against every candidate outside of the set. Minmax intuitively considers the “worst” comparison of each candidate and then output all candidates that have the “best” of these worst comparisons. Our algorithms for these Condorcet methods are constructed using the same techniques and building blocks as for the previous methods. Due to space constraints, we do not present our algorithms in detail her but rather refer the reader to our implementation [15].

7 Borda

Borda count is a ranked voting method where each assignable rank is associated with a pre-defined number of points that the corresponding candidate receives. The winner typically is the candidate who has received the most points in total (summed over all ballots). A famous application of Borda count is the election

of the winner of the grand final in the Eurovision Song Contest, but it is also used for national elections, for example in the Republic of Nauru.

The following choice space can be used to capture Borda, where we interpret \mathcal{P} both as a list and a set: $\mathbf{C}_{\text{Borda}}(\mathcal{P}) = \{(x_1, \dots, x_{n_c}) \mid \forall i : x_i \in \mathcal{P} \wedge \forall i \in \mathcal{P} \exists j : x_j = i\}$. A NIZKP π^{Enc} for the well-formedness of ballots for this choice space is presented in [21]. By definition of Ordinos, the encrypted aggregated tally $(E_{\text{pk}}(p_i))_{i \in [n_c]}$ then consists of encryptions of the sum of points p_i that candidate c_i received. In principle, one can now use the same MPC tallying protocols presented in [6] for single-/multi-vote to (i) output the candidate with the highest points, (ii) output the k candidates with the most points, or (iii) output all candidates that cleared a certain threshold of points. However, for the standard case (i) we propose a more efficient way that is not quadratic but linear in the number of candidates: We use the algorithm `GetMaxIdx()` (cf. Section 3) and then apply f_{dec} ; the winner is the candidate for whom decryption yields 1.⁸

The benchmarks of these algorithms are presented in Figure 5, where the result functions (ii) and (iii) are implemented using the algorithms by [6]. As the benchmarks show, our algorithm for (i) and the algorithm for (iii) can be computed highly efficiently. Due to the linear growth, this should still be the case even if there are much more candidates than the maximum of 40 that we benchmarked. Result function (ii) shows, as expected, a quadratic growth in the number of candidates. However, the runtime for ≤ 40 candidates remains in a range that is often still reasonable for practical elections. Also, recall that our benchmarks are essentially independent from n_v and n_t . With the same reasoning as for Theorem 3 we obtain:

Theorem 6 (Security of Borda voting with Ordinos). *Let \mathcal{E} and $\pi^{\text{KeyShareGen}}$ be as for Theorem 3. Let π^{Enc} be the NIZKP from above, and let \mathbf{P}_{MPC} be one of our MPC components for (one of the result functions for) Borda voting as defined in this section. Then, the Ordinos instance using these primitives is an accountable and private (and hence tally-hiding) voting system for Borda (using that result function).*

8 Conclusion

We have proposed, implemented, and benchmarked several new accountable tally-hiding MPC components for Ordinos. These are the first tally-hiding implementations for the Hare-Niemeyer method, IRV, multiple variants of Condorcet, and Borda. The performance of our MPC components is determined by the number of candidates while being essentially independent of the number of trustees and the number of voters, as long as the aggregated ballots still meet the bound b_{ct} . Analogously to [6], due to the comparison protocols with sub-linear communication cost, our runtimes are almost independent of the network (local vs. Internet). Our instantiations achieve reasonable runtimes that allow

⁸ If always a single winner should be determined, one can use a tie-breaking algorithm after `GetMaxIdx()`, similarly to what we describe in Section 5 for `GetMinIdx()`. Note that this adds only a small linear overhead.

for deployment in real-world applications. In future work, it would be interesting to investigate optimizations for our algorithms and to implement further voting methods.

References

1. J. D. Cohen, *Improving Privacy in Cryptographic Elections*. Citeseer, 1986.
2. A. Hevia and M. A. Kiwi, “Electronic jury voting protocols,” *TCS*, 2004.
3. R. Wen and R. Buckland, “Minimum Disclosure Counting for the Alternative Vote,” in *VoteID, Luxembourg.*, 2009.
4. A. Szepieniec and B. Preneel, “New Techniques for Electronic Voting,” ePrint Report 2015/809.
5. S. Canard, D. Pointcheval, Q. Santos, and J. Traoré, “Practical Strategy-Resistant Privacy-Preserving Elections,” in *ESORICS 2018*, vol. 11099. Springer, 2018.
6. R. Küsters, J. Liedtke, J. Müller, D. Rausch, and A. Vogt, “Ordinos: A Verifiable Tally-Hiding E-Voting System,” in *EuroS&P*. IEEE, 2020, pp. 216–235.
7. V. Cortier, P. Gaudry, and Q. Yang, “A toolbox for verifiable tally-hiding e-voting systems,” ePrint Report 2021/491.
8. R. Küsters, T. Truderung, and A. Vogt, “Accountability: Definition and Relationship to Verifiability,” in *CCS*, 2010.
9. T. Haines, D. Pattinson, and M. Tiwari, “Verifiable Homomorphic Tallying for the Schulze Vote Counting Scheme,” in *VSTTE 2019*, 2019.
10. K. Ramchen, C. Culnane, O. Pereira, and V. Teague, “Universally Verifiable MPC and IRV Ballot Counting,” in *FC 2019*, ser. LNCS. Springer, 2019.
11. W. Jamroga, P. B. Rønne, P. Y. A. Ryan, and P. B. Stark, “Risk-Limiting Tallies,” in *E-Vote-ID 2019*, 2019.
12. A. Juels, D. Catalano, and M. Jakobsson, “Coercion-Resistant Electronic Elections,” ePrint Report 2002/165.
13. J. Heather, “Implementing STV securely in Prêt à Voter,” in *CSF*, 2007.
14. J. Benaloh, T. Moran, L. Naish, K. Ramchen, and V. Teague, “Shuffle-sum: coercion-resistant verifiable tallying for STV voting,” *TIFS*, 2009.
15. F. Hertel, N. Huber, J. Kittelberger, R. Küsters, J. Liedtke, and D. Rausch, “Ordinos Code Repository,” <https://github.com/JulianLiedtke/ordinos>.
16. R. Küsters, T. Truderung, and A. Vogt, “Verifiability, Privacy, and Coercion-Resistance: New Insights from a Case Study,” in *S&P 2011*, 2011.
17. R. Canetti, “Universally Composable Security: A New Paradigm for Cryptographic Protocols,” in *FOCS 2001*. IEEE Computer Society, 2001.
18. R. Küsters, “Simulation-Based Security with Inexhaustible Interactive Turing Machines,” in *CSFW-19*, 2006, see [25] for a full and revised version.
19. I. Damgård, M. Jurik, and J. B. Nielsen, “A Generalization of Paillier’s Public-Key System with Applications to Electronic Voting,” *Int. J. Inf. Sec.*, 2010.
20. H. Lipmaa and T. Toft, “Secure Equality and Greater-Than Tests with Sublinear Online Complexity,” in *ICALP 2013*, vol. 7966. Springer, 2013, pp. 645–656.
21. J. Groth, “Non-interactive Zero-Knowledge Arguments for Voting,” in *ACNS 2005*.
22. Maine State Legislature, “Ranked Choice Voting in Maine,” <http://legislature.maine.gov/lawlibrary/ranked-choice-voting-in-maine/9509>, 2020.
23. Electoral Commission NSW, “NSW State Election Results 2015,” <https://pastvtr.elections.nsw.gov.au/SGE2015/la-home.htm>, 2021.
24. M. Schulze, “The Schulze Method of Voting,” *CoRR*, 2018.
25. R. Küsters, M. Tuengerthal, and D. Rausch, “The IITM Model: A Simple and Expressive Model for Universal Composability,” *Journal of Cryptology*, 2020.

Hyperion: An Enhanced Version of the Selene End-to-End Verifiable Voting Scheme

Peter Y. A. Ryan¹, Simon Rastikian², and Peter B. Rønne¹

¹ University of Luxembourg, Esch-sur-Alzette, Luxembourg

`peter.ryan@uni.lu`, `peter.roenne@gmail.com`

² École Normale Supérieure, Paris, France

`simon.rastikian@ens.fr`

“Everything should be made as simple as possible, but no simpler.”

— A. Einstein

Introduction We present a novel, end-to-end verifiable scheme, Hyperion, inspired by the Selene scheme [1], which similarly provides highly transparent verification: voters check their vote directly in plaintext in the tally. It has a number of advantages including eliminating the tracker collision threats in Selene, indeed our construction does not need trackers for verification. The new scheme should give voters a greater sense of privacy.

In the original Selene, vote/tracker pairs are revealed in the tally on the Bulletin Board. Voters are later notified of their tracker: by providing them with a private “alpha” term which along with their private, trapdoor key, opens their commitment to reveal their tracker. As long as the trackers remain private and deniable ballot privacy is preserved. However, some voters, understandably, find the public posting of the trackers alongside the votes troubling. Furthermore, Selene suffers from the possibility of a coercer claiming that the alternate tracker proffered by a coerced voter is their own.

Hyperion, by contrast, does not publicly reveal trackers, indeed, we can do away entirely with trackers. Instead, the voter identifies her vote in the tally by identifying the commitment which, along with her “alpha” term and her trapdoor key opens to a constant, e.g. the identity 1. This is rather like identifying your house by finding the door that opens to your key. This is still deniable, but the mechanism is now different: a coerced voter identifies a commitment paired with the coercer’s required vote and, if necessary, computes using her trapdoor secret key, the fake alpha term that opens this to 1.

At first glance this seems to counter the tracker collision threat, but we have just shifted the problem: now the coercer might claim that the commitment the voter points to is theirs. To counter this we propose a further innovation: each voter gets an individual view of Bulletin Board BB . Each view is verifiably derived from the BB with its own, independent shuffling. Thus the rows and betas appear in a different form and order for each voter, so even a shoulder-surfing coercer cannot identify his beta in the voter’s view.

The Setup Signing keys for the voters, PK_i are published on the master bulletin board, one row per voter. For voter V_i trapdoor keys, x_i and $h_i := g^{x_i}$, are generated by the voter's app and h_i is registered along with the casting of the vote, along with suitable ZK proofs of knowledge of x_i .

Voting Voting is much as in Selene: V_i sends an encryption of her vote with the associated plaintext awareness and well-formedness proofs, and her public trapdoor key h_i along with ZK proofs of knowledge of x_i . We denote the concatenation of these proofs by Π_i . This is signed and posted to the master BB against PK_i :

$$PK_i, \text{Sign}_i(\{\text{Vote}_i\}, h_i, \Pi_i)$$

Tellers now generate the analogues of the alpha and beta terms of Selene: each trapdoor public key h_i is raised to a fresh, secret random r_i . The corresponding (pre-)alpha term g^{r_i} is kept secret for the moment by the tellers:

$$PK_i, \text{Sign}_i(\{\text{Vote}_i\}, h_i, \Pi_i), h_i^{r_i}$$

Tallying On the BB , ballots with valid signatures and proofs are identified and for these we extract the encrypted votes and beta commitments:

$$(\{\text{Vote}_j\}, h_j^{r_j})$$

These are now put through verifiable, hybrid, parallel mixes: the vote terms are subjected to a conventional re-encryption mix, but the commitment terms are subjected to an “exponentiation” mix: all raised to a common, secret exponent s . Such mixes can be implemented using Verificatum in suitable modes, [2]. Finally, the votes are verifiable decrypted outputting:

$$(\text{Vote}_j, h_j^{r_j \cdot s})$$

The rows are now sorted to group votes for the same candidate together. For V_i we now create an individual view by applying a further parallel mix, but here we just permute and re-randomise within the candidate groups. For each voter there will be a different, independent common exponent s_i for the exponentiation mix of the commitment terms:

$$(\text{Vote}_j, (h_j^{r_j})^{s \cdot s_i})$$

The mix tellers keep secret for now the “alpha” terms: $\alpha_i = g^{r_i \cdot s \cdot s_i}$.

Note that the commitment terms are not opened or decrypted, thus the plaintext votes appear paired with cryptographic blobs.

At notification time, α_i is sent to V_i who raises this to x_i and finds the match among the beta terms in her view, so identifying her vote. In the event of coercion, V_i identifies a row containing the coercer's required vote and computes the alpha which, when raised to x_i , matches the beta in this row.

Discussion We note that the individual views are only necessary if we want to fully counter the tracker/commitment collision problem and render the scheme fully coercion resistant. Even without the individual views the scheme provides the same guarantees as Selene (e.g. receipt-freeness and coercion mitigation). It is significantly simpler and provides a greater sense of privacy than Selene.

We can in fact retain trackers in our construction, and this may inspire a greater sense of assurance in the verification. In this case, we assign trackers in the setup phase, as in Selene. Now a coerced voter computes a fake alpha that opens the alternative beta to **her own** tracker, i.e., no need to identify a fake tracker. Furthermore, the association of trackers with voters can now be made public! This allows universal verification that all the trackers are distinct. Note that in this construction each voter only see their own tracker in their own view. Nowhere are all the trackers displayed alongside the votes, so the privacy concerns of Selene do not arise.

Conclusions We have outlined Hyperion, a new scheme, with three variants, that provides voters with a similarly direct, intuitive way to verify that their votes are correctly included in the tally. It is conceptually much simpler than Selene and avoids the tracker collision threats, indeed we can do away with trackers altogether. Furthermore, voters should feel more comfortable with this scheme as it does not involve the public posting of tracker/vote pairs.

Achieving full coercion resistance comes at the cost of introducing individual views for each voter, but this should give voters a greater sense of privacy. The individual boards may be better suited for smaller elections, where the collision threat is more troublesome. The construction without individual boards, may be useful for large elections where the collision problem is anyway less important, or contexts in which coercion threats are deemed mild, e.g. boardroom voting. We have also described a variant that retains the trackers, but having the remarkable property that now voters do not need to identify a fake tracker, and indeed the voter/tracker association can be made public (inter alia demonstrating that each voter gets a unique tracker).

We stress though that the integrity of the voters' verification checks relies on their secret trapdoor key not being compromised. We do not therefore recommend that Hyperion be used for critical, binding elections.

Full details of the constructions and proofs will appear in the full version of this paper. The authors acknowledge support of the Luxembourg National Research Fund and the Research Council of Norway for the joint project SURCVS.

References

1. Peter Y A Ryan, Peter B Rønne, and Vincenzo Iovino. Selene: Voting with transparent verifiability and coercion-mitigation. In *International Conference on Financial Cryptography and Data Security*, pages 176–192. Springer, 2016.
2. Douglas Wikström. Blackbox constructions from mix-nets. *Cryptology ePrint Archive*, Report 2019/995, 2019. <https://eprint.iacr.org/2019/995>.

The Road to Deploying e-Voting

The challenges of enabling public scrutiny

Xavier Monnat¹ and Simon Oswald¹

¹ Post CH Ltd, Wankdorfallee 4, 3030 Bern, Switzerland
e-voting@post.ch

Abstract. Transparency and public scrutiny are necessary in order to gain the trust of voters and the authorities when it comes to electronic voting. However, public scrutiny is a methodical challenge for a company to implement. Swiss Post disclosed an earlier version of the source code for its e-voting system in 2019 and conducted a public intrusion test. Multiple researchers published attacks, highlighted vulnerabilities, and suggested improvements. But the disclosure was also met with criticism in the expert community for various reasons.

This paper presents the insights Swiss Post gained from its experiences regarding transparency in 2019 – from the point of view of a vendor and developer. Based on those learnings, we explain our new approach to enabling public scrutiny with our new system. This approach includes expert, community-friendly terms and conditions, closer collaboration with the community, improvement of the system’s documentation and auditability, easier opportunities to compile and run the system and a permanent bug bounty programme. Public scrutiny of this new system with complete verifiability started on a step-by-step basis at the beginning of 2021.

Keywords: Online Voting, E-voting, Public Scrutiny, Transparency, Complete Verifiability, Bug Bounty, Community.

1 Introduction

1.1 Outline of this paper

This article has two main aims. Firstly, we present from the point of view of a vendor and developer of an e-voting solution the conclusions from the transparency experiences we gained in 2019. Secondly, we describe our new public scrutiny approach, which we are applying since then.

The new approach involves amongst other things a step-by-step disclosure, actively engaging with academic experts and the hacker community, friendly participation conditions and a permanent bug bounty programme.

1.2 E-Voting in Switzerland

Switzerland has a longstanding tradition of direct democracy, allowing Swiss citizens to vote approximately four times a year on elections and referendums. In recent years, voter turnout hovered below 40 percent¹.

The vast majority of voters in Switzerland fill out their paper ballots at home and send them back to the municipality by postal mail, usually days or weeks ahead of the actual election date. Remote online voting (referred to as e-voting in this document) is the digitalization of the postal voting and would provide voters with additional advantages. Firstly, it would guarantee the timely arrival of return envelopes at the municipality (especially for Swiss citizens living abroad). Secondly, it would improve accessibility for people with disabilities. Thirdly, it would eliminate invalid ballots when inadvertently filling out the ballot incorrectly or doing a formal error with the ballot.

In Switzerland, the cantons are responsible for the organisation of the elections (also at federal level) and for tallying the results at cantonal level. The Confederation defines the legal conditions for a valid election that the cantons have to fulfil. This construct defined in the constitution is also applied for e-voting. In that sense Swiss Post is a provider of the cantons.

In the past, multiple cantons offered e-voting to a part of their electorate. They have been testing e-voting since the beginning of 2000². Over 300 trials in real elections and votations have been conducted with different systems since then.

Many voters would welcome the option to vote online - provided the e-voting system protects the integrity and privacy of their vote [2]. State-of-the-art e-voting systems alleviate the practical concerns of mail-in voting and, at the same time, provide a high level of security. Above all, they must display three properties [6]:

- Individual verifiability: allow a voter to convince herself that the system correctly registered her vote
- Universal verifiability: allow an auditor to check that the election outcome corresponds to the registered votes
- Vote secrecy: do not reveal a voter's vote to anyone

Following these principles, the Federal Chancellery defined stringent requirements for e-voting systems. The current draft of the Ordinance on Electronic Voting (VEleS - Verordnung über die elektronische Stimmabgabe) and its technical annex (VEleS annex)[1] describes these requirements.

1. Eidgenössisches Departement für auswärtige Angelegenheiten EDA: Swiss Political System - Direct Democracy. <https://www.eda.admin.ch/aboutswitzerland/en/home/dossiers/overview.html/content/aboutswitzerland/en/meta/news/politik>. Retrieved on 2021-07-10.

2. For the milestones from the point of view of the Federal Government, see <https://www.bk.admin.ch/bk/en/home/politische-rechte/e-voting/chronik.html>

1.3 New legal basis in CH

In June 2019³, the Federal Council mandated the Federal Chancellery together with the cantons on restructuring the trial operation. The redesign is based on the following objectives:

- Further development of the systems towards a complete verifiable e-voting system
- effective control and oversight
- Increasing transparency and trust
- stronger connection with the scientific community

In December 2020⁴, the Federal Council mandated the Federal Chancellery to implement the measures required to achieve the goals in stages. In doing so, it is relying on the final report of the Federal Chancellery and the cantons⁵. This was prepared in close cooperation with experts from science and industry.

In April 2021⁶, the Federal Council began the public consultation procedure⁷ on the new legal basis for the trial operation of e-voting. Groups, companies and individuals can express their views on the proposals during the consultation procedure.

1.4 Swiss Post as a developer

Swiss Post has been developing an e-voting system for several years. Its former system without universal verifiability was used by the cantons Thurgau, Fribourg, Neuchâtel and Basel-Stadt from 2016 to 2019.

By developing an e-voting system, Swiss Post is digitizing what it does best as a trustworthy state-owned company: the secure transport of confidential information. Swiss Post aims to guarantee the established principle of mail secrecy in the digital world. It is building on its extensive experience as a trustworthy carrier of sensitive information, providing new digital solutions for companies, authorities and private citizens and enabling them to exchange confidential data via a reliable Swiss provider.

In 2019 we decided to publish an earlier version of the documentation and the source code of the e-voting system to allow public scrutiny. In response to this release, multiple

³ e-Voting: Federal Council to reframe trial phase and delay introduction as a regular voting channel, 27.06.2019 <https://www.bk.admin.ch/bk/en/home/dokumentation/medienmitteilungen.msg-id-75615.html>

⁴ e-voting: Federal Council launches redesign of trials, 21.12.2020 <https://www.bk.admin.ch/bk/en/home/dokumentation/medienmitteilungen.msg-id-81772.html>

⁵ Redesign and relaunch of trials, Final report of the Steering Committee Vote électronique (SC VE) https://www.bk.admin.ch/dam/bk/en/dokumente/pore/Final%20report%20SC%20VE_November%202020.pdf.download.pdf/Final%20report%20SC%20VE_November%202020.pdf

⁶ Redesign of e-voting trials: consultation procedure opened, 28.04.2021 <https://www.bk.admin.ch/bk/en/home/dokumentation/medienmitteilungen.msg-id-83257.html>

⁷ Reports and studies <https://www.bk.admin.ch/bk/en/home/politische-rechte/e-voting/berichte-und-studien.html>

researchers published attacks, highlighted vulnerabilities, and suggested improvements [3, 4, 5, 7]. Swiss Post is thankful to all security researchers for their contributions and the opportunity to improve the system's security guarantees. This experience with transparency is the focus of this paper. We discuss our experience and what we have learned from it in depth from chapter 2 onwards.

In the autumn of 2019, Swiss Post decided to continue developing an e-voting system for Switzerland independently and acquired all rights that are necessary for the independent development of its e-voting system from its former technology partner Scytl in spring 2020⁸. This step enables Swiss Post to provide the cantons with a solution developed in Switzerland for Switzerland.

In-house development provides numerous advantages: we can better meet the demanding security objectives, adapt our solution to Swiss-specific requirements, and provide a solution "made in Switzerland.". A system developed in Switzerland was also demanded by some politicians.

Swiss Post believes that transparency is necessary in order to gain the confidence of the voters and cantons when it comes to electronic voting. This is why we have started in the beginning of 2021 to disclose our new e-voting system with universal verifiability in order to enable public scrutiny.

On a more abstract level, transparency can also be seen as a digital ethics concept. Swiss Post is engaged in the field of digital ethics and sustainability for public service companies. For example, we are collaborating with academics in order to develop an integrated framework for ethical and sustainable digitalization[11].

1.5 Swiss Post new System with complete verifiability

Swiss Post's new e-voting system enables universal verification of the votes that are cast. When counting votes, the cantonal electoral authorities can verify whether votes casted in the electronic ballot box have been manipulated. Such a system with universal verifiability has never been used in Switzerland before.

Various requirements set out in the recently published ordinances have been incorporated into Swiss Post's system and approach. The new system largely meets the requirements of the recently published legal framework and any required modifications to the already published elements will be made over the coming months, in line with the project roadmap.

⁸ <https://www.evoting-blog.ch/en/pages/2020/an-e-voting-system-for-switzerland-and-by-switzerland>

2 Public Scrutiny

2.1 Public scrutiny as a method

Public scrutiny of an IT system has in our view two aims:

- Independent experts can critically examine the system and report potential vulnerabilities to the system developer.
- Establishing trust

Public Scrutiny requires transparency. At least in Switzerland, it seems that transparency is gaining in importance, especially in the field of e-government services. There are also increasing with political demands. A recent example is the SwissCovid app⁹.

There has recently been discussions among experts how to implement transparency methodically in order to reach the goals of public examination. Haines and Roenne[10] propose a close collaboration between election management bodies and/or vendors and the scientific community: *“To ensure e-voting systems are secure it is important the vendors and election management bodies engage with researchers in an open, transparent and collaborative process”*.

They draw upon their own experiences scrutinizing e-voting systems worldwide and give recommendations to vendors for a better scrutiny:

1. *Clear claims: Make clear claims about the security of the system*
2. *Thorough documentation: Provide comprehensive, clear, correct and consistent documentation*
3. *Minimality: The source code should be minimal and only contain code relevant to the system under review*
4. *Buildable: The source code should be easily buildable*
5. *Executable: The built system should be executable*
6. *Exportable: It should be possible to export test vectors for independent verification*
7. *Consistent documentation and source: The documentation should correspond to the source code*
8. *Regularly Updated: The open source¹⁰ system should be updated regularly to reflect the fixes of previously found bugs*
9. *Minimal restrictions on disclosure: Avoid long vulnerability disclosure times*

In international procedures such as the Recommendation CM/Rec(2017)5 of the Council of Europe¹¹ (particularly VI. Transparency and observation), the importance of public scrutiny is highlighted: *“Conscious, therefore, that only those e-voting systems*

⁹ <https://github.com/SwissCovid/swisscovid-app-android>

¹⁰ Haines and Roenne uses the term ‘open source’ here. We interpret it as “openly available”. We do not think that an “open source system” in a legal sense is a necessary condition to be able to follow these principles.

¹¹ https://search.coe.int/cm/Pages/result_details.aspx?ObjectId=0900001680726f6f

which are secure, reliable, efficient, technically robust, open to independent verification and easily accessible to voters will build public confidence, which is a prerequisite for holding e-elections”.

Public scrutiny is a methodical challenge to implement for a vendor and developer. The first disclosure in 2019 was indeed met with criticism in the expert community for various reasons. This experience helped us to define a new approach.

We will in the following explain what mistakes we, as a vendor, made in 2019 and what we learned from then. Then we explain in depth what we changed in the approach and how we proceed for this new system scrutiny.

2.2 Public scrutiny of the system in 2019

In 2019 took Swiss Post two actions for public scrutiny. The source code of an earlier version of the system was disclosed and a public intrusion test was conducted as required by the law¹². This disclosed system was never used in real elections. However, parts of the system were identical to the system in use.

During the four-week intrusion test, around 3,200 international IT experts tried to inflict targeted attacks on this e-voting system. After the completion of the intrusion test, there were no manipulated votes in the electronic ballot box. The hackers did not manage to infiltrate the e-voting system. Attempts at overloading the system through a DDoS attack was unsuccessful. The hackers submitted a total of 173 findings. The Federal Chancellery, Cantons and Swiss Post confirmed 16 of them. They all fall under the lowest classification level, “Best Practice”, and are thus considered non-critical. The entire assessment process for the findings was overseen by representatives of the Confederation and the cantons^{13, 14, 15}.

However, multiple researchers published attacks, highlighted vulnerabilities, and suggested improvements in the source code and the documentation [3, 4, 5, 7]. One of them concerned the individual verifiability, which was also present in the former system in use. As a consequence, Swiss Post decided to suspend its former system immediately¹⁶ and later to abandon it completely in order to focus on the new system¹⁷.

¹² For a summary of these tests, see chapter. 3 “Public intrusion test and publication of the source code” in [9]

¹³ For the Federal Chancellery’s final report of the intrusion test see https://www.bk.admin.ch/bk/en/home/politische-rechte/e-voting/oeffentlicher_intrusionstest.html

¹⁴ For Swiss Post’s final report see https://www.post.ch/-/media/post/evoting/dokumente/abschlussbericht-oeffentlicher-intrusionstest-post.pdf?vs=1&sc_lang=en&hash=49EE456EE4D0B4EBA14BD3F6767E051E

¹⁵ For an analysis from the point of view of the developer Scytl, see Puggiali, 2019, Implementing a public security scrutiny of an online voting system: the Swiss experience [8]

¹⁶ Ballot box not hacked, errors in the source code – Swiss Post temporarily suspends its e-voting system - Swiss Post <https://www.post.ch/en/about-us/media/press-releases/2019/swiss-post-temporarily-suspends-its-e-voting-system>

¹⁷ Swiss Post to focus solely on new system with universal verifiability <https://www.post.ch/en/about-us/media/press-releases/2019/swiss-post-to-focus-solely-on-new-system-with-universal-verifiability>

Of course, this can be seen as a failure of the transparency exercise. But from a methodical point of view, our conclusion is that the transparent system disclosure - with the involvement of experts from all over the world for hacking tests – proved to be an effective method for quickly identifying and rectifying vulnerabilities.

2.2.1 Criticism to the public scrutiny 2019

Participants in the disclosure programme criticized the way in which we published the code. The sources of this criticism were academics, IT practitioners, critical tech journalists and activists affiliated with an anti-e-voting initiative. These criticisms can be roughly divided into three categories: scrutiny limitations, code quality, participation conditions.

Scrutiny limitations

There was criticisms about what could be tested. The source code lacked building instructions and failed to compile easily. There was no possibility to simulate an event

Code quality

An important aspect of the code quality is how easily it is for external experts to read and scrutinize the code. It does not refer to the code's function. We received criticism for poorly readable code and large amounts of dead code, which are obstacles to thorough scrutiny.

Participation conditions

The conditions refer to the rules a researcher has to follow in order to get access to the published code. Several points were criticized, e.g.:

- There was an impression that Swiss Post could forbid the publication of findings
- The delay for responsible disclosure was set on 45 days after the last communication of Swiss Post (and not after the report). There was a fear that Swiss Post could indefinitely extend this period.
- It was not clear whether researchers may work together and whether a registered researcher may work with someone who is not registered.
- There was the impression that the conditions prevent the publication of studies that lack security-critical issues.

This challenge concerning participation conditions was also pointed out by Puiggali (2019) [8]. He identifies in the expert community a “lack of a common consensus of which mechanism must be used for reporting issues to the election managers”.

2.2.2 What have we learned

We drew various conclusions from the experiences we gained and condensed them into six recommendations. These are recommendations from the point of view of a developer and vendor of a complex IT system. There are valuable learnings from other point of views which we do not focus on¹⁸.

1. Do not minimize the possible impact of an identified weakness

There is always room for a rational and scientific argument that a weakness is not important. However, such an attitude does not help you building up trust in the expert community and can backfire when more problems are found later.

2. Define participation conditions which are accepted by the expert community

The disclosure procedures must be widely accepted. Otherwise, the community will not participate or will criticize the approach. This includes especially the ease of participation modalities, the rules for reporting and publishing issues (responsible disclosure).

3. Think of transparency as a process instead of a one-off measure

Transparency is not a one-off measure, but instead requires constant dialogue with the expert community. In other words: transparency is not an event, but a process. Interaction and an open dialogue with the community are important. Transparency about what happens after a weakness is submitted is equally important. In 2019 we did not enough focus on that and prepared primarily the initial disclosure.

4. Be prepared for conflicting expectations to transparency in the public

Public scrutiny is risky. In our opinion, this is because of conflicting expectations of transparency: On one hand, the aim of transparency is to identify problems and bugs in order to improve the system. On the other hand, bugs can be seen in the eye of the public or contradictors considered as a proof for a system to be irremediably insecure. This attitude does not take into account the reality of software engineering because any defects and weaknesses identified do not mean the end of the system.

As a vendor and developer, it is important to be able to analyse the consequences of the findings, especially in terms of the impact for cantons and voters.

The publication of a system is a process that is still not widely known by the Swiss population and companies. We think that the more the method of public scrutiny is established in the future, the less this area of conflict will be a problem.

¹⁸ As e.g. Maurer (2019)[9] points out the experience raises also legal and policy questions about the certification process for state authorities and regulators

5. *Choose the starting point wisely*

On the one hand it is useful to start public scrutiny early, on the other side the system must have a certain maturity level. E-voting is politically controversial, at least in Switzerland. A group of politicians tried to introduce a ban of e-voting. They stopped their attempt after failing collecting enough signatures for a popular vote in 2020¹⁹. Political opponents of e-voting might try to use findings as a proof for their political stance that e-voting cannot be trusted. This risk is higher if the starting point of such initiatives is not chosen wisely.

6. *Treat code quality as equal important as code functionality*

By quality, we do not mean the quality of the functionalities, but aspects such as auditability, readability, maintainability, publishability (e.g. publication format) of the code. Missing auditability and readability does not necessarily make a system vulnerable; however, it can make it difficult for experts to scrutinize and understand the code and it can increase the likelihood of the system being, or becoming vulnerable.

Based on these learnings, we defined a new approach for the future system.

3 Our new approach to public scrutiny

Since the beginning of 2021, Swiss Post is disclosing its new e-voting system in stages, as part of a community programme. The source code and key documents are continuously, iteratively and openly published. This iterative approach is central: we regularly update the published artefacts and continuously improve our publication, based on feedback and learning from the e-voting community and IT experts. We deliberately started at an early stage to give experts enough time to test the system and for us to implement the improvements that were reported.

As mentioned above in chapter [2.1], Haines and Roenne[10] gave nine recommendations to vendors for a better scrutiny. We agree on these nine principles and hope that our new publication approach meets them.

In the following, we deepen and explain the most important aspects of our community programme:

- a) Definition and implementation of the new disclosure with the inclusion of cantons and experts
- b) Simplified access to the code (no registration is required) and a streamlining of the terms and conditions of use (Code of Conduct)
- c) Improvement of the documentation and auditability of the system
- d) Key parts of the code under an open source licence
- e) Transparent development
- f) Publication of a compilable system

¹⁹ <https://www.swissinfo.ch/ger/schweiz-demokratie-abstimmung-e-voting-moratorium-volksinitiative-zurueckgezogen/45855362>

- g) Opportunities to simulate contests
- h) Permanent bug bounty programme
- i) Communication

3.1.1 Definition and implementation of the new disclosure with the inclusion of cantons and experts (a)

We started early enough the first activities for the new disclosure, involving experts from different horizons as well as the cantons regularly, to be able to define, validate and adapt the new disclosure on a step by step approach.

3.1.2 Simplified access to the code and a streamlining of the terms and conditions of use (b)

For the new disclosure, no registration with acceptance of terms and conditions is required anymore.

We took a collaborative approach of defining a Code of Conduct with experts from the community. The Code of Conduct governs access to the e-voting material within the framework of the Swiss Post e-voting community programme.

3.1.3 Improvement of the documentation and auditability of the system (c)

As already introduced, an important aspect of the quality is how easily it is for external experts to scrutinize the system. It is on one side addressed by the concept of auditability, on the other side with a comprehensive accompanying documentation. We have also rewritten various parts of the code to simplify external verification of the specifications and source code.

Auditability

The auditability is answering to the following question: which characteristics of the codebase lead to increased effort to execute a review or constraints on the pool of possible reviewers? Leading experts of the industry defined a model to address this need.

The source code, test artefacts and documents to be published are analysed and checked according to this model. The results of this audit are listed in an auditability report, which is published²⁰.

This process proves to be an efficient measure and ensuring a holistic approach to a system with resulting measures and improvements to be applied in order to be ready to be assessed by external auditors.

Key published items

The published items aims to be a comprehensive set of elements to ensure a clear, correct a comprehensive description of the system, to sustain public scrutiny and help the

²⁰ <https://gitlab.com/swisspost-evoting/e-voting/e-voting-documentation/-/tree/master/Reports>

expert understand the system, from the definition of the system to the run itself, explaining the constraints and the quality principles.

We are committed to providing transparent information on the operating principles and security precautions of the system. We want to foster cooperation with independent experts and thus facilitate the continuous improvement of the system.

3.1.4 Open source (d)

We will not be making the entire system available on an open-source basis. The system includes certain patents, which prevent us from releasing it for free commercial use worldwide. However, we decided to publish key parts of the code under an open source licence: key cryptographic algorithms, known as crypto-primitives, are available in a library provided by Swiss Post. The Swiss Post verifier, an open verifier software to verify the Swiss Post Voting Protocol, will be as well available under an open source licence. This means that we can offer free use and development of relevant parts of the source code.

We have chosen the permissive open-source licence (Apache 2) to facilitate and encourage further development and reuse by third parties. This allows Swiss Post to gain on the experience of the reuse of this code.

There is sometimes a confusion between open source and transparency. In our view, these two concepts should strictly be separated. Software can be source-available and meet all necessary transparency and scrutiny criteria without being open source. Open Source is a licence model for the software. In fact, with our approach we try to fulfil the demand formulated by Haines and Roenne²¹.

3.1.5 Transparent Development (e)

Transparent development is Swiss Post's approach to software development for its e-voting service. The transparent development approach supports disclosure on the following points:

- Swiss Post permanently discloses software releases.
- Software increments are published between releases and are made available on a designated branch. An increment is a defined feature/task designed to make it easier to understand the changes in the source code.
- The commits history for the software increments and releases are published. For all files, the commit history can be examined either per release or per increment.
- The contributions from the community (e.g. pull requests) are reviewed, and if accepted, integrated into the e-voting source code. This means that changes or improvements from the community can be included in the source code.

²¹ “We highlight that any solution used for significant elections should be well designed, carefully analysed, deftly built, accurately documented and expertly maintained. Until e-voting system implementations are clear, comprehensible, and open to public scrutiny security standards are unlikely to improve.” [10]

3.1.6 Publication of a compilable system (f)

The building of the whole e-voting system has been made much easier and uses standard tools only. The maven build can be called in a standard environment with only one command.

3.1.7 Opportunities to simulate contests (g)

We provide researchers with a docker environment to run the whole e-voting system and simulate election events on their own infrastructure – i.e. they can run a ballot with productive data and also carry out attacks (including adjustments to the source code).

3.1.8 Bug Bounty programme (h)

In order to continuously improve the security of its digital products, Information Security at Swiss Post operates a Group-wide bug bounty programme²² and E-Voting is also part of it. As with all products, it runs first in a private scope with a limited number of hackers. As soon as the programme reach a maturity stage, it will be converted into a public bug bounty. The private bug bounty for e-voting started in December 2020.

For the private bug bounty, we defined a grid combining for the different scopes an evaluation based on the industry standard common vulnerability scoring system (CVSS), plus specific scenarios taking in consideration the e-voting's specificities.

The public programme will cover three aspects:

- Static tests (Search for errors in the disclosed documentation or source code)
- Dynamic tests (Search for errors by analysing the executable system in a private infrastructure)
- Internet tests (Attacks on the provider's infrastructure, fixed term annual test)

We see the bug bounty programme as a programme within the community programme. This means that one does not have to register for the bug bounty programme in order to be able to scrutinize the system if one does not wish to. All system artefacts and documents which are disclosed can be accessed on GitLab without signing up to the bug bounty.

3.1.9 Communication (i)

Communication tailored to the target audience is important and needs to be addressed on an ongoing basis. For the expert audience, we publish a blog²³ for each milestone (disclosed objects) and send an email with information. In order to explain in depth our

²² The bug bounty programme makes an important contribution to digital transformation at Swiss Post on the aspects of: Cyber security, Digital trust, Culture of tomorrow and Digital transformation: <https://www.post.ch/en/about-us/responsibility/swiss-post-bug-bounty>

²³ <https://www.evoting-blog.ch/en>

system and approach, we conduct expert webinars²⁴ on a regular basis. The aim of these communication activities is to engage dialog with the expert community. Transparency does not reach its goals if there is no one scrutinizing the system.

We have built a dedicated community website²⁵ where the access to all information and to all disclosed system items can be found. The platform is updated step by step and is written in a language that also non experts can understand. However, the dialogue between Swiss Post's e-voting team and experts takes place on GitLab where all findings are published.

When communicating the public scrutiny programme, it must be emphasized that “if bugs are found, it is a success”. For big and established companies, this can be a challenge as it involves to admit publicly that a system does not yet live up to the company's quality standards.

It is important to prepare communicatively for all kind of scenarios. Obviously, weaknesses and bugs can be found in the code or an attack during an intrusion test can be successful. Such things might have to be communicated. But there are many other scenarios which can happen in the public discussion on social or traditional media and which you have to prepare for:

- Responsible disclosure is not respected
- An expert expresses his disagreement with your analyzis of your finding.
- The methodical approach is criticized (e.g. not enough items are disclosed, participation conditions)
- The code quality is criticized as it makes it difficult to scrutinize the system
- Fake news about submitted or not submitted findings

Such scenarios should already be anticipated when you define the modalities of public scrutiny. By defining the modalities according to the expectations and needs of the expert communities, you mitigate the risks of such scenarios. If such scenarios happen, it is a huge communicative challenge. As a developer, you are automatically in a position of weakness compared to an independent expert when it comes to public trust.

Finally, companies often communicate a security issue after it is solved. In the e-voting context, we communicate as soon as we have analyzed it. This makes the communication more complicated regarding to the messages.

3.2 First learnings from the new approach

It is obviously too early for an evaluation of the new approach, as we are still in the process of disclosing the system. Since January 2021, we have disclosed system artefacts such as the cryptographic protocol, the library of cryptographic primitives, the system specification, architecture documentation, information about the development process and a test concept²⁶.

²⁴ <https://www.evoting-blog.ch/en/pages/2021/security-by-design-in-swiss-post-s-new-e-voting-system>

²⁵ <https://evoting-community.post.ch/en/>

²⁶ For an up-to-date overview over all disclosed system parts see the community website (<https://evoting-community.post.ch/en/>)

At the time of writing this article, the bug bounty programme has not yet been launched, and the complete code base and open-source verification software have not yet been disclosed. This will happen shortly.

Our first experience is encouraging. We have received findings that we have already been able to implement²⁷, and the reactions are mainly positive. So far, we have not received any fundamental objections to our approach or to aspects of it.

We are, however, observing an ongoing discussion about the necessity of an open-source licence for entire e-voting systems. It seems that our far-reaching system transparency has not yet convinced all experts, even though we try to follow open-source principles regarding system transparency and system development, and even though part of the system is actually open source.

Thanks to our step-by-step procedure, we will be able to modify our approach if we do not achieve our goals or if we receive well-founded suggestions for improvement from the community.

4 Conclusion

In early 2021, we began the disclosure of the beta version of the new system as part of a community programme. The aim of the programme is that independent experts can critically examine the system and report potential vulnerabilities.

Based on the learnings of the disclosure of an earlier system, we improved our approach. We have for instance modified the participation conditions so that they are more accepted within the expert community. We have put more focus on code quality in order to make it easier for external experts to scrutinize it. And we think of transparency as a process instead of a one-off measure. As presented, we are using well-established methods for the public scrutiny such as continual and complete disclosure of the system's source code, transparent development or an optional ongoing bug bounty programme.

In fact, transparency does not reach its goals if there is no one scrutinizing the system. E-voting is a very specialised and still comparatively young field. The existing systems on the market/countries are relatively different (comparisons are difficult), with a limited amount of productive deployment and therefore few active people being able to use such systems. The effort to find and encourage experts to contribute is a delicate task. Motivations are diverse, and even with significant rewards, the effort to contribute remains high. The different scopes defined allow the system to be analysed from different angles, but require on the other side more explanation for the community. To address these issues, we try to engage more actively with the expert community.

In fact, this paper will also serve as a good base for starting a discussion with the expert community, with the aim of obtaining feedback and suggestions for improvement.

Finally, we can say that the publication of a white-box system is still a relatively recent topic, at least in Switzerland. We are proud to contribute to it and hope that some

²⁷ All findings are published on GitLab (<https://gitlab.com/groups/swisspost-evoting/-/issues>)

of our reflection may not only apply to the special e-voting context, but also be useful for similar projects of IT companies in other industries.

References

1. Die Schweizerische Bundeskanzlei (BK): Federal Chancellery Ordinance on Electronic Voting (OEV), Draft of 28 April 2021.
2. gfs.bern: Vorsichtige Offenheit im Bereich digitale Partizipation - Schlussbericht. Mar. 2020
3. R. Haenni: Swiss Post Public Intrusion Test: Undetectable attack against vote integrity and secrecy. 2019.
4. T. Haines, S. J. Lewis, O. Pereira, and V. Teague: "How not to prove your election outcome". In: 2020 IEEE Symposium on Security and Privacy (SP). IEEE. 2020, pp. 644–660.
5. P. Locher, R. Haenni, and R. E. Koenig: Analysis of the cryptographic implementation of the swiss post voting protocol. 2019.
6. B. Smyth: "A foundation for secret, verifiable elections." In: IACR Cryptology ePrint Archive 2018 (2018), p. 225.
7. V. Teague and O. Pereira: Report on the SwissPost-Scytl e-voting system, trusted-server version. 2019
8. J. Puiggalí Implementing a public security scrutiny of an online voting system: the Swiss experience, 2019
9. Ardita Driza Maurer; The Swiss Post/Scytl transparency exercise and its possible impact on internet voting regulation, 2019
10. T. Haines, P. Roenne : New Standards for E-Voting Systems: Reflections on Source Code Examinations, 2021. <https://eprint.iacr.org/2021/391.pdf>
11. Wallimann-Helmer, Ivo & Terán, Luis & Portmann, Edy & Schübel, Hanna & Pincay Nieves, Jhonny. (2021). An Integrated Framework for Ethical and Sustainable Digitalization. In book: 14th International Conference on Theory and Practice of Electronic Governance (ICEGOV 2021)

Use of Electronic Voting in the Albanian Parliamentary Elections in 2021

Jurlind Budurushi^[0000-0002-6732-4400]

<https://jurlindbudurushi.com>
e-voting@jurlindbudurushi.com

Abstract. This paper describes the introduction of electronic voter identification, voting and tallying in the parliamentary elections on April 25, 2021 in Albania. First, we outline the legal framework regarding the use of technology in elections. Further, we describe in a chronological order the most relevant steps that paved the way towards the use of electronic voter identification, voting and tallying. We report on pre-election preparation activities, election day and election results. Next, we highlight challenges of legal, procedural, social, organisational and technical nature. We conclude our work by suggesting future work directions.

Keywords: Electronic Voter Identification · Electronic Voting & Tallying · Parliamentary Elections · Albania · SMARTMATIC.

1 Introduction

The political agreement between the two major parties in Albania in 2017, and the consensus of the political council in 2020 led to the use of information technology in the forthcoming elections. Consequently, in this paper we describe the introduction of electronic voter identification, voting and tallying in the parliamentary elections on April 25, 2021 in Albania. The extremely short time frame regarding relevant activities, such as policy writing, procurement, implementation, testing and independent auditing, not only diminished public discourse substantively, but also posed various challenges to the electoral process. Note that while electronic voter identification was deployed in all polling stations across the country, electronic voting and tallying was deployed only in a limited number of polling stations.

This paper is structured as follows: In section 2 we outline the relevant parts of the Albanian Electoral Code, in particular regarding election principles and voter's rights, as well as regarding the use of voting technology such as electronic voter identification, electronic voting, and electronic tallying systems. Section 3 describes in a chronological order the most relevant steps that paved the way towards the use of technology in elections, including recommendations by well-known institutions like OSCE/ODIHR, political agreements, electoral commission's acts, regulatory decisions, and vendor (system) selection. Section 4 reports on used devices, pre-election preparation activities, election day and election results. In section 5 we highlight challenges and classify these in legal, procedural,

social, organisational and technical challenges. Section 6 concludes the paper by suggesting future work directions.

2 Electoral Code

In this section we outline the relevant parts of the Albanian Electoral Code (AEC)¹ in particular regarding election principles and voter's rights, as well as regarding the use of voting technology such as electronic voter identification, electronic voting, and electronic tallying systems. Note that the current AEC has been published with the support of the Organisation for Security and Cooperation in Europe (OSCE).²

2.1 Election Principles and Voter's Rights

According to *Article 3* of AEC, elections shall be *free, secret, equal, and direct*. Further, each voter has the right to cast only one vote. In addition, every Albanian citizen who has reached the age of 18, including election day, has the right to vote and to be elected independent of their race, ethnicity, gender, language, political conviction, religion, physical ability or economic status. Though not exhaustive, the voter's rights are also to be found on the website³ of the Central Electoral Commission (CEC).

2.2 Technology in Elections

Article 22 is one of the articles in the AEC that mainly addresses the use of information technology in elections.⁴ It defines the CEC competences concerning information technology in elections, while focusing on different aspects:

- Enabling CEC to explore, experiment and decide on the use of information technology systems⁵ in elections for addressing specific aspects and/or procedures of the electoral process.
- Restricting CEC to implement the use of information technology systems in elections via pilot projects with at least 10 percent of the number of voters in each implementation stage.
- Specifying that the use of information technology systems must first ensure electronic identification of voters and their post-election verification, voting with and through electronic equipment, and machine assisted counting of cast ballots. Further, the used systems must be applicable for at least five consecutive electoral processes.

¹ <http://kqz.gov.al/wp-content/uploads/2021/04/Electoral-Code-of-Albania-english-2.pdf>, last accessed 18.06.2021

² <https://www.osce.org/presence-in-albania/477547>, last accessed 18.06.2021

³ <http://kqz.gov.al/te-drejtat-e-votuesit/?lang=en>, last accessed 18.06.2021

⁴ The use of information technology in elections was added to the AEC with the changes on July 23, 2020.

⁵ Includes both hardware and software equipment.

- Defining that the identification, selection, and use of information technology systems must be subject to the principles of legality, transparency, inclusiveness, security, efficiency, and sustainability.
- Describing the operational testing procedures for information technology systems prior to elections. Operational testing must take place in public sessions by randomly selecting from each electoral administration zone at least three percent of therein used systems, and with no less than 50 participants.
- Verifying the correctness of information technology systems used to vote or count ballots by manually counting ballots and comparing the results in at least 10 percent of the voting centres. The verification process must take place in public sessions and the voting centres must be selected randomly. Note that mismatches may become subject of criminal investigation.

Other articles of the AEC do also touch on the use of technology in elections, but mostly on the sideline. *Article 14* appoints the Deputy Commissioner as responsible regarding voter electronic identification technology. *Article 20* defines competences of the regulator regarding the analysis and examining of draft acts of normative nature on information technology systems. *Article 23* outlines the procedure for introducing technology in elections and decision-making. *Article 25* discusses the use of technology in elections in the context of out-of-country voting procedures. *Article 81* describes the location and preparation of ballot counting centres, allowing CEC to decide regarding the use of recording cameras and screens for displaying the ballot papers before their evaluation. *Article 179* introduces the procedures with respect to the procurement of information technology systems and equipment. In addition, *Article 179* states that the use of information technology in elections must be implemented through pilot projects, where at most 20 percent of the voters shall be included, however electronic voter identification can include 100 percent of the voters.

3 The Journey towards Electronic Voter Identification, Voting, and Tallying

In this section we describe in a chronological order the most relevant steps, including recommendations by well-known institutions like OSCE/ODIHR, political agreements, electoral commission's acts, regulatory decisions, and vendor (system) selection for electronic voter identification, voting, and tallying in the parliamentary elections on April 25, 2021.

3.1 OSCE/ODIHR Recommendations

As usual, after being invited by the government of the Republic of Albania, OSCE/ODIHR deployed an *Election Observation* mission in the previous, parliamentary elections on June 25, 2017. The final report⁶ of this mission does not contain any explicit recommendation regarding the introduction of electronic

⁶ <https://www.osce.org/files/f/documents/4/d/346661.pdf>, last accessed 18.06.2021

voting, i.e. the use of information technology in elections as means for addressing any electoral challenges identified in this and prior OSCE/ODIHR reports, e.g. vote-buying or double-voting. Rather, many of the OSCE/ODIHR EOM interlocutors emphasised the necessity to involve experts beyond the largest parliamentary parties to study policy options, including electronic voting, before future reform. The report only outlines that the introduction of electronic voting for the next elections stems by a request of the Democratic Party and a consequent agreement with the Socialist Party reached on May 18, 2017. This agreement was the first step towards the use of information technology in elections.

3.2 Commissioner's Acts

In this section we introduce four relevant Central Electoral Commission (CEC) Commissioner's Acts regarding the use of information technology in elections.

Commissioner's Act No. 82 from October 12, 2020⁷

After the political agreement on May 18, 2017, and motivated by the consensus of the political council for using information technology in the forthcoming elections, reached on June 5, 2020⁸, CEC released on October 12, 2020 an internal order for creating a working group. The working group, monitored by the CEC deputy commissioner, consisted of seven participants, four CEC members, two external IT experts, and the director of the finance directorate. The goal of the working group was to evaluate information technology systems, and deliver a report on October 25, 2020. To achieve this goal the following tasks were defined:

- Research and identify information technology systems that can be used in the forthcoming elections for electronic voter identification and voting.
- Evaluate the use of such technology in the corresponding context.
- Make an overall assessment of financial costs.
- Present, if possible, a number of entities that have developed such technology.
- Prepare a list of legal acts and technical protocols that will enable the implementation and use of such technology.

Commissioner's Act No. 111 from March 2, 2021⁹

With the decision No. 111 on March 2, 2021, CEC determined the electoral administration zone in the municipality of Tirana for the implementation of the electronic voting and tallying pilot project. As specified by the Regulatory Commission the selected zone should not consist of more than 55 polling stations. Only three zones met this requirement in the municipality of Tirana:

⁷ <http://kqz.gov.al/wp-content/uploads/2020/11/Urdher-nr.-82-per-nritjen-e-grupit-te-punes-per-pajisjet-e-teknologjise-se-informacionit.pdf>, last accessed 18.06.2021

⁸ <http://ata.gov.al/2020/06/05/keshilli-politik-arrin-konsensusin-per-reformen-zgjedhore/>, last accessed 18.06.2021

⁹ http://kqz.gov.al/wp-content/uploads/2021/03/dadsit_210301_pv_caktimi_zaz-40_votimelektronik.pdf, last accessed 18.06.2021

- Electoral administration zone No. 33, consisting of 52 polling stations.
- Electoral administration zone No. 38, consisting of 50 polling stations.
- Electoral administration zone No. 40, consisting of 32 polling stations.

Based on consultations with the subjects participating in the elections, nominated parties and independent candidates, CEC decided in favour of the electoral administration zone No. 40.

Commissioner’s Act No. 239 from April 15, 2021 ¹⁰

With the decision No. 239 on April 15, 2021, CEC approved the procedural and technical rules with respect to the management and conduction of the electronic voting, and tallying processes. CEC defined 35 rules for the electronic voting process, and 24 rules for the electronic tallying process. In addition, this document defines the steps of the voting process. Thereby, it is important to emphasise that voters are allowed to cast an invalid¹¹ vote intentionally, and verify the corresponding Paper Audit Trail before casting their vote. However, a voting session, i.e. the time after a poll worker activates the electronic voting device until voters confirm their cast vote, can last only six minutes.

Commissioner’s Act No. 206 from April 23, 2021 ¹²

With the internal order from April 23, 2021, CEC approved the operational procedures for preserving and protecting voters’ personal data stored and processed in the electronic identification devices. This document consists of the following sections:

- Description of the voters’ personal data gathered by the CEC.
- Description of the infrastructure of CEC, where voters’ personal data is stored and processed.
- Definition of the authorised personnel having access to the electronic identification devices, and the corresponding access policies.
- Listing of the devices containing voters’ personal data, and definition of the organisational and technical requirements to be fulfilled by these devices during their usage.
- Definition of the operational procedures for the management in the field of the electronic identification devices.

3.3 Regulatory Commission Decisions

In this section we introduce four relevant decisions of the Regulatory Commission (RC)¹³ of CEC regarding the use of information technology in elections. In

¹⁰ http://kqz.gov.al/wp-content/uploads/2021/04/Vendim_-239_210414_Pv_rregulla_procedurale_votim_nr_elektronik-3_Komisoneri.pdf, last accessed 18.06.2021

¹¹ Note that a blank vote is also considered as invalid.

¹² http://kqz.gov.al/wp-content/uploads/2021/04/Urdher_206_210425_ruajtja-e-te-dhenave.pdf, last accessed 18.06.2021

¹³ According to AEC it is the competent body for approval of acts of normative nature.

addition to where and how the pilot project shall be conducted, the RC defines in these decisions technical characteristics that must be met by the electronic identification, voting, and tallying devices.

Decision No. 02 from October 31, 2020 ¹⁴

The objective of this decision is: *The type and technical characteristics that must fulfil the systems and equipment for **electronic identification of voters** procured and used in elections in the Republic of Albania.* Thereby, RC decided that voters shall be identified through electronic identification devices in all polling stations. Further, RC defined a number of technical criteria to be met by the electronic identification devices. Below we list the most relevant.

- Voter identification shall be based on the voters' electronic register stored on the device.
- The device shall contain all eligible voters in all polling stations, but show only the voters of the polling station for which it is configured.
- It shall be possible to cross-check all used devices whether any voter has voted twice, including in other polling stations or for other voters.
- The device shall work autonomously without being connected to external networks.
- The device shall read the Machine Readable Zone of the biometric identity document to identify voters, but also support manual voter identification.
- The device shall store voters' biometric fingerprint and confirm the identification process by providing a physical trail that contains voters' information, including their picture.

Decision No. 04 from November 13, 2020 ¹⁵

The objective of this decision is: *The type and technical characteristics that must fulfil the systems and equipment **for electronic voting and electronic tallying** procured and used in elections in the Republic of Albania.* Thereby, RC decided that the process of voting and tallying shall be conducted electronically, for at least 10% and at most 20% of the voters. Further, RC defined a number of technical criteria to be met by the electronic voting and tallying devices. Below we list the most relevant.

- The devices shall be operated in a closed network, not externally accessible.
- The devices shall be separated and not communicate or exchange information with the electronic identification devices, in order to preserve voters' vote secrecy.
- The voting device shall store cast votes, but without corresponding times-tamps.

¹⁴ http://kqz.gov.al/wp-content/uploads/2020/12/VendimI-002_DJ_Miratimi-Karakteristikat-teknike-te-Sistemeve-ne-Zgjedhje.pdf, last accessed 18.06.2021

¹⁵ http://kqz.gov.al/wp-content/uploads/2020/12/Vendimi-004_DJ_Tek_Pajisje-Vot_Num_Elek.pdf, last accessed 18.06.2021

- The voting device shall confirm voters' selection on the screen and print a Paper Audit Trail, which is deposited on a box attached to the device.
- The screen of the voting device shall, as much as possible, prevent taking a picture of the vote.
- The devices shall work autonomously without being connected to external networks.
- The devices shall be activated by the Commissioner for each voter, and automatically disable after voting.
- The voting device shall enable voters with disabilities, e.g. voters with vision loss, to cast their vote.

Decision No. 04 from February 12, 2021 - Changes in the Decision No. 04 from November 13, 2020 ¹⁶

The objective of this decision is: *The type and technical characteristics that must fulfil the systems and equipment **for electronic voting and electronic tallying** procured and used in elections in the Republic of Albania.* Thereby, RC decided on the following changes:

- The electronic voting and tallying process shall be conducted in one of the electoral administration zones¹⁷ of the municipality of Tirana, but not in more than 55 polling stations.
- The support for voters with disabilities, e.g. voters with vision loss, is repealed.

Decision No. 17 from April 14, 2021 ¹⁸

The objective of this decision is: *Procedural rules of the electronic voting and tallying process in the electoral administration zone No. 40 in Tirana.* This decision outlines a framework around the electronic voting and tallying process. It is important to emphasise that through this decision, in accordance to the concrete situation, CEC can decide to proceed the voting and tallying process with ballot papers.

3.4 Evaluation and Procurement Process

Note that at the time of writing this article CEC has not published the report by the respective working group regarding the evaluation of potential information technology (e-voting) systems. Though the report should have been delivered on October 25, 2020 to CEC. Instead, CEC has published on their website only

¹⁶ http://kqz.gov.al/wp-content/uploads/2021/02/Vendim_Nr.004_210212_Per-disa-ndryshime-ne-vendimin-nr.04-date-13.11.20....pdf, last accessed 18.06.2021

¹⁷ The electoral administration zone was specified by the Commissioner's Act from March 2, 2021.

¹⁸ http://kqz.gov.al/wp-content/uploads/2021/04/vendim_17_krr_rregulla_procedurale_votim_nr_elektronik_zaz_40.pdf, last accessed 18.06.2021

corresponding information about the steps and progress of the evaluation and procurement process.

On November 13, 2020¹⁹ CEC notified about their request to 20 companies with international experience in information technology and/or in election processes. The companies were requested to estimate the costs for the electronic voter identification system and devices, based on the type and technical characteristics as specified in the decision No. 02 from October 31, 2020 by the Regulatory Commission Decisions.

On November 18, 2020²⁰ CEC notified about the progress of the market testing process for the implementation of technology in elections. In order to improve the price indicators CEC had send their request to two additional companies, however not specifically named. Six out of the 22 contacted companies provided a price estimation, ranging from approximately 15 million to 28 million USD. Furthermore, CEC asked the responding companies to estimate the additional cost for implementing the Electronic Voters Identification and Results Management System and the Voting and Counting Electronic Devices, based on the type and technical characteristics as specified in decision No. 04 from November 11, 2020 by the Regulatory Commission Decisions.

On January 6, 2021²¹ CEC published an update regarding the procurement procedure for electronic identification of voters. From the six companies, only SMARTMATIC²² submitted the required documentation and offer (approximately 20 million USD without VAT) and delivered the required sample equipment.

On January 14, 2021²³ CEC notified to have completed the evaluation of the bid and technical solution submitted by SMARTMATIC regarding the information technology systems/procedures for electronic identification of voters and their post election verification. CEC announced that after reviewing the documentation, the financial bid and technical solution decided to proceed with the stage of *Terms and Contact Negotiation* with SMARTMATIC. No further information was published by CEC regarding the negotiation or contract signing process. However, the Albanian Public Procurement Agency (APPA)²⁴ announced²⁵ SMARTMATIC as a winner with negotiation without publication with an offer of 19.975.77,23 USD (without VAT), on page 229 of the bulletin No. 11²⁶

¹⁹ <http://kqz.gov.al/2020/11/13/njoftim-per-media-tirane-me-13-nentor-2020/?lang=en>, last accessed 18.06.2021

²⁰ <http://kqz.gov.al/2020/11/18/njoftim-per-media/?lang=en>, last accessed 18.06.2021

²¹ <http://kqz.gov.al/2021/01/06/perditesim-informacioni-lidhur-me-proceduren-prokuruere-per-identifikimin-biometrik-te-zgjedhesve/>, last accessed 18.06.2021

²² <https://www.smartmatic.com/>, last accessed 18.06.2021

²³ <http://kqz.gov.al/2021/01/15/update-information-regarding-the-procurement-procedure-for-electronic-identification-of-voters/?lang=en>, last accessed 18.06.2021

²⁴ <http://www.appa.gov.al>, last accessed 18.06.2021

²⁵ APPA was notified already on January 15, 2021 about the decision by CEC.

²⁶ <http://www.appa.gov.al/GetData/DownloadDoc?documentId=870a8751-2cf7-46a2-be64-4ad2b28c1967>, last accessed 18.06.2021

from January 25, 2021. In addition, APPA announced SMARTMATIC also as a winner with negotiation without publication with an offer of 960.180 USD (without VAT) regarding the electronic voting and tallying devices, on page 262 of the bulletin No. 42²⁷ from March 23, 2021. According to the internal APPA rules, SMARTMATIC had five days after the corresponding announcement to accept the offer and sign the contract. SMARTMATIC confirmed the reached agreements implicitly by announcing on their website²⁸ about being selected by CEC.

4 E-voting supported Elections

In this section we describe the electronically supported parliamentary elections of April 25, 2021. We introduce the used devices, describe the pre-election preparation activities, the election day and also the election results.

4.1 Electronic Devices: Voter Identification, Voting and Tallying

SMARTMATIC provided²⁹ their voter verification device *VIU-818*³⁰ for electronic voter identification, and their premium voting device *A4-517*³¹ for electronic voting and tallying, shown in Figure 1a and 1b respectively.

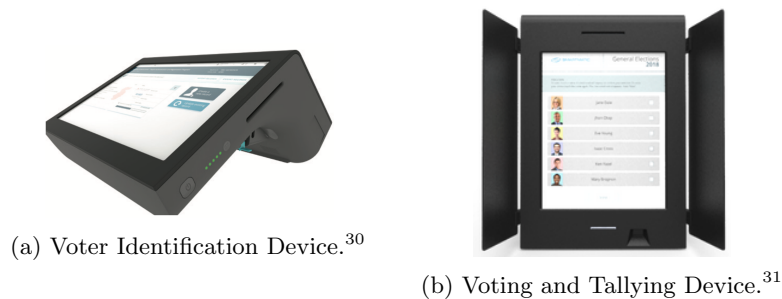


Fig. 1: SMARTMATIC Devices.

²⁷ <http://www.app.gov.al/GetData/DownloadDoc?documentId=0fa380e7-f0e7-4c29-b4bf-91c2a118f7a6>, last accessed 18.06.2021

²⁸ <https://www.smartmatic.com/case-studies/article/smartmatic-supports-seven-elections-in-six-countries-over-five-months/>, last accessed 18.06.2021

²⁹ <https://www.smartmatic.com/media/article/albania-strengthens-confidence-in-the-election-process-through-technology/>, last accessed 18.06.2021.

³⁰ https://www.smartmatic.com/fileadmin/user_upload/VIU_Desktop_ProductSheet_2020.pdf, last accessed 18.06.2021

³¹ https://www.smartmatic.com/fileadmin/user_upload/PremiumVotingMachine.pdf, last accessed 18.06.2021

4.2 Pre-Election Phase

Recruiting and Training of Electoral Employees

In this section we outline the procedures and steps taken by Central Electoral Commission (CEC) for recruiting and training electoral employees.

Recruiting Process Recruiting started with a public announcement³² for part-time employment for a large number of positions by CEC on December 10, 2020. Further, CEC expressed their interest for hiring trainers to train the electoral administration on January 7³³, 2021, on March 5³⁴, 2021, and on March 30³⁵, 2021, publicly. It is to be noted that since the first announcement for hiring trainers for the electoral administration CEC required the same amount of trainers, namely 54 for training the commission of the electoral administration zones, 265 for training the polling stations commissions, and 92 for training members of the ballot tallying teams. In addition, the application deadline for trainers in the last announcement (March 30, 2021) was on March 31, 2021. Last, but not least, CEC announced on March 30³⁶, 2021 *a one day employment opportunity* as operator of voters' electronic identification device on election day, publicly. In addition to a number of strict criteria to be met by the applicants, the application deadline was on April 5, 2021, with an incentive of approximately 75€. Note that neither a public nor an explicit announcement was made by CEC for recruiting operators for the electronic voting and tallying devices.

Training Process There is a lack of public information regarding the training process for both electronic voter identification, and electronic voting and tallying devices. The only publicly available information is 1) a report by *abcnews.al*³⁷ about training operators of the electronic voter identification device in the Municipality of Shkodra, refer to Figure 2; and 2) a statement³⁸ by SMARTMATIC about having trained 460 field support technicians who provided support to poll workers, and 32 poll workers who operated the electronic voting devices for the pilot.

³² <http://kqz.gov.al/2020/12/10/njoftim-per-punesimin-e-punonjesve-me-kohe-jo-te-plote-pune-per-zgjedhjet-per-kuvend-te-dates-25-prill-2021/?lang=en>, last accessed 18.06.2021

³³ <http://kqz.gov.al/2021/01/07/announcement-for-expression-of-interest-for-trainers-to-train-the-electoral-administration/?lang=en>, last accessed 18.06.2021

³⁴ <http://kqz.gov.al/2021/03/05/announcement-for-expression-of-interest-for-trainers-to-train-the-electoral-administration/?lang=en>, last accessed 18.06.2021

³⁵ <http://kqz.gov.al/2021/03/30/cec-is-hiring-trainers-to-train-the-electoral-administration/?lang=en>, last accessed 18.06.2021

³⁶ <http://kqz.gov.al/2021/03/03/employment-opportunities-at-the-cec-operator-for-electronic-voter-identification-devices-on-election-day/?lang=en>, last accessed 18.06.2021

³⁷ <https://abcnews.al/operatoret-ne-shkoder-trajnohen-per-votimin-elektronik-kqz-ve-ne-dispozicion-5500-pajisje/>, last accessed 18.06.2021

³⁸ <https://www.smartmatic.com/media/article/albania-strengthens-confidence-in-the-election-process-through-technology/>, last accessed 18.06.2021

³⁹ <https://www.youtube.com/watch?v=F8Kah9xVDE4>, last accessed 18.06.2021



Fig. 2: Operator Training for Electronic Identification of Voters.³⁹

Voters' Education

In order to educate voters about elections in general, CEC provides on their website⁴⁰ a voter's corner, where topics like 1) Where is my polling station?; 2) How can I vote?; 3) Voter's Rights; 4) Voters' List; and 5) FAQs about filling in the ballot, are addressed. Further, CEC provides also a voter's education section⁴¹ on their website, where posters⁴² and videos⁴³ describing the procedures of the electronic voters' identification are published. With respect to educating voters about the voting procedures through the electronic voting device, CEC has published a video⁴⁴, and has also organised a simulated election from April 3 to April 11, 2021.⁴⁵ Thereby, all the voters of the Election Administration Zone No. 40 had the opportunity to participate in the simulation. Except for the number of participants, 547, and that the ballot didn't match the one used on election day⁴⁶, no information about potential challenges and lessons learned is available. Last, but not least, CEC organised an awareness campaign⁴⁷ with first time voters, where students from the Faculty of Medicine in Tirana were introduced to CEC, its leading bodies and competences. Thereby, the participants' interest was focused on topics such as electronic voter identification, the electronic voting and tallying pilot project, general voting procedures, and the correct marking of the ballot paper.

Demonstration and Operational Testing

In this section we describe the CEC activities for demonstrating and testing the functional operation of the electronic identification, voting and tallying devices.

⁴⁰ <http://kqz.gov.al/?lang=en>, last accessed 18.06.2021

⁴¹ <http://kqz.gov.al/strategjia-e-edukimit/?lang=en>, last accessed 18.06.2021

⁴² <http://kqz.gov.al/wp-content/uploads/2021/04/Electronic-identification-1-scaled.jpg>, last accessed 18.06.2021

⁴³ http://kqz.gov.al/wp-content/uploads/2021/04/3_IDENTIFIKIMILELEKTRONIK.webm, last accessed 18.06.2021

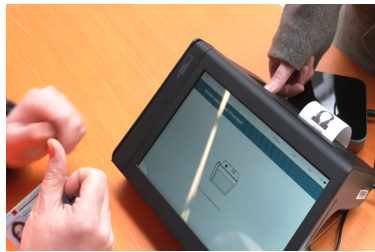
⁴⁴ <https://www.youtube.com/watch?v=1UHNYSxLOY>, last accessed 18.06.2021

⁴⁵ <http://kqz.gov.al/2021/04/02/cec-simulation-of-the-electronic-voting-in-municipality-unit-no-10/?lang=en>, last accessed 18.06.2021

⁴⁶ <https://www.osce.org/files/f/documents/2/7/484688.pdf>, last accessed 18.06.2021

⁴⁷ <http://kqz.gov.al/2021/04/07/cec-awareness-campaign-with-first-time-voters/?lang=en>, last accessed 18.06.2021

Demonstration The functionality of the electronic voters' identification devices was demonstrated in three municipalities, in Tirana⁴⁸, Kukes⁴⁹ and Fier⁵⁰, as shown in Figures 3a, 3b and 3c respectively. During the demonstrations in Kukes and Fier the *Deputy Commissioner* made the following statements respectively: "...identification is simple for the citizens and is completed in a short time. Electronic identification avoids multiple voting and does not allow the voting in a centre where the citizen is not registered...", and "...one of the primary functions of the device is to identify cases when a voter attempts multiple voting. Since the device has stored only the voters' list of that centre where the device is installed, it becomes impossible to vote for those voters who are not registered in that voting centre...". While in Tirana participated more than 200 subjects selected by the political parties, no information is published about Kukes and Fier. However, according to CEC the demonstrations showed an efficient functioning of the devices. Note that except for the "demonstration" in the scope of voters' education, aiming to familiarise voters with the electronic voting process, no further demonstration of the electronic voting and tallying devices was conducted.



(a) Tirana on February 22, 2021.⁴⁸



(b) Kukes on March 4, 2021.⁴⁹



(c) Fier on March 8, 2021.⁵⁰

Fig. 3: Demonstration of Electronic Voter Identification Devices.

⁴⁸ <https://www.youtube.com/watch?v=uF7gHP6APG4>, last accessed 18.06.2021

⁴⁹ <http://kqz.gov.al/2021/03/04/electronic-voters-identification-device-demonstrated-in-kukes/?lang=en>, last accessed 18.06.2021

⁵⁰ <http://kqz.gov.al/2021/03/08/electronic-voters-identification-device-demonstrated-in-fier/?lang=en>, last accessed 18.06.2021

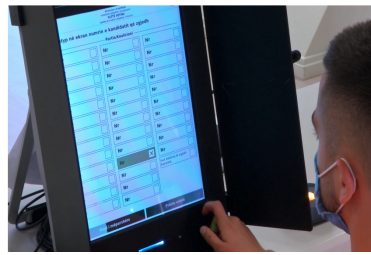
Operational Testing In accordance to *Article 22, sentence 6*, of the Albanian Electoral Code (AEC), CEC organised and conducted operational tests for the electronic voter identification devices, and for the electronic voting and tallying devices. The operational tests for the electronic voter identification, shown in Figure 4a, took place on April, 16 2021⁵¹. Thereby, 3% of the devices were randomly selected and tested against the following test cases:

- Presented voter is registered in the according voting centre.
- Presented voter is registered in another voting centre.
- Multiple voting attempts.
- Acoustic signals released by the devices.

At the end of this operational test a draft report shall confirm the functionality of the devices, and whether these fulfil the criteria specified by the Regulatory commission with *Decision No. 02 from October 31, 2020*. It is to be emphasised that neither this report was published, nor any information about participants. Further, during the testing the *State Election Commissioner* stated that “...the process of electronic voter identification is easy to use, and that CEC has taken all the necessary measures in order for the project to be successful...”. The operational tests for the electronic voting and tallying devices, shown in Figure 4b, took place on April 23, 2021⁵². According to CEC the operational tests were conducted in compliance to the AEC, i.e. to identify whether the devices fulfil the criteria specified by the Regulatory commission with *Decision No. 04 from November 11, 2020*. No further information about the test cases or participants was published.



(a) Electronic Voter Identification.⁵¹



(b) Voting and Tallying.⁵²

Fig. 4: Devices' Operational Testing.

⁵¹ <http://kqz.gov.al/2021/04/16/cec-tests-electronic-identification-devices-pei/?lang=en>, last accessed 18.06.2021

⁵² <http://kqz.gov.al/2021/04/23/the-central-election-commission-performed-the-operational-test-of-electronic-voting-and-counting-devices/?lang=en>, last accessed 18.06.2021

4.3 Election Day

In accordance with the *Article 11* of the Albanian Electoral Code (AEC), on election day polling stations were open between 7AM and 7PM. While the electronic voter identification was deployed in all, 5199 polling stations across the country, electronic voting and tallying was deployed only in 32 polling stations.⁴⁶

Electronic Voter Identification

Before opening the voter identification procedure, poll workers configured the electronic identification device. To configure the device poll workers were required to enter first the configuration PIN, then the corresponding polling station number⁵³, and then activate a smart card. After configuration, to operate the device poll workers were required to enter the operating PIN, activate the smart card, and confirm. After confirmation, the device printed an identification opening protocol, containing the polling station number, the total number of eligible voters for the corresponding polling station, and the number of identified voters (zero in the beginning). For identification voters used their national electronic identification card. Poll workers swiped the identification card through the device, which read the corresponding Machine Readable Zone (MRZ). If the MRZ was not readable, for instance due to a damaged card, poll workers could search for voters manually. In both cases the device displayed on its screen voters' personal information and picture. After checking that the displayed information matches that on the card, poll workers required voters to put their finger on the biometric scanner. The device stored voters' biometric fingerprint locally, thus enabling to identify any attempts of double-voting in the corresponding polling station. Note that poll workers also marked voters fingers with indelible ink. After storing the fingerprint successfully the device printed a confirmation, containing the voters' personal information and picture. Poll workers collected the printed confirmation within an envelope for later auditing purposes. Next, voters were able to cast their vote either on a paper ballot or electronically. At the end of the election day poll workers closed the identification process by inserting the smart card, entering the operational PIN, and printing the final identification report. Note that according to the preliminary report⁴⁶ by the Office for Democratic Institutions and Human Rights (ODIHR) approximately 4% of the polling stations delayed their opening due to issues with starting-up identification devices. Approximately 3% of eligible voters were registered using paper lists, due to malfunctioning of devices or absence of technical operators. In some polling stations, poll workers used both the devices and paper lists. These challenges were also reflected in various local media, like televisions⁵⁴ and newspapers⁵⁵.

⁵³ This enables the device to identify only voters registered in the corresponding polling station as eligible.

⁵⁴ <https://www.youtube.com/watch?v=7xGLYVRd4ck>, last accessed 07.07.2021

⁵⁵ <https://exit.al/142-qendra-votimi-kane-probleme-me-identifikimin-elektronik/>, last accessed 18.06.2021

Electronic Voting

Electronic voting was implemented in 32 polling stations⁵⁶ in the municipality of Tirana. A total of 23.597 eligible voters could cast their vote electronically. After successfully identifying voters, poll workers activated the stand-alone voting device. Then, voters entered the voting booth and cast their vote by selecting on the touchscreen either a party or an independent candidate or by casting an invalid vote. When selecting a party, voters had also the option to cast one vote to individual candidates of the selected party. After confirming their selection the voting device displayed and also printed on a Paper Audit Trail (PAT) voters' selection. Voters were required to verify that the displayed information and the PAT matched their selection. After their final confirmation, voters' selection was stored on the device locally, the PAT automatically added into a ballot box, and the device automatically disabled. Note that after the PAT was printed, voters could change their selection only one time. Further, voters could not touch the PAT, which was shown behind the glass of the printer, refer to Figure 5. It must be noted that many voters required assistance to cast their vote electronically.⁴⁶



Fig. 5: Setup of the Electronic Voting Device.⁵⁷

4.4 Election Results

While the election results⁵⁸ are still not finally certified by CEC due to ongoing evaluations of manipulation allegations⁵⁹, the preliminary results⁶⁰ from the electronic voting were published already a day after election day. According to CEC, 12.096 out of 23.597 eligible voters participated in the elections and cast their vote electronically, resulting in 11.976 valid and 120 invalid votes. Note that no complaints were made regarding the results from the electronic voting.

⁵⁶ Each polling station was equipped with two electronic voting devices.

⁵⁷ <https://www.youtube.com/watch?v=1UHNyGSxLOY>, last accessed 18.06.2021

⁵⁸ <http://kqz.gov.al/results/results2021.htm>, last accessed 18.06.2021

⁵⁹ A number of active and passive election manipulations are being evaluated by CEC and the Special Anti-corruption Structure (SPAK). Refer to http://kqz.gov.al/wp-content/uploads/2021/05/Regjistri-i-Njoftimeve-KAS_210511-1.pdf and <https://spak.al/2021/04/27/njoftim-24/> respectively.

⁶⁰ <https://www.youtube.com/watch?v=g9S6hSvr0t4>, last accessed 18.06.2021

5 Challenges

Despite being rated as a successfully completed project by CEC⁶¹, a number of challenges accompanied the electronically supported elections. These challenges can be classified as legal, procedural, social, organisational, and technical. *Legal challenges* include contradictions to *Article 22* of the AEC, e.g. 1) Electronic voting was implemented with less than 10% of voters; 2) Voters with disabilities were excluded; 3) No manual ballot verification was conducted; and 4) Lack of analysis regarding systems' applicability for the next five consecutive electoral processes. *Procedural challenges* include contradicting and non best-practice procedures, e.g. 1) Invalid votes printed as QR-Codes on PAT, making it difficult for voters to verify; 2) No manual audit for tallying required; 3) No explicit random selection of USBs storing votes; 4) No security required during electronic transfer of results; 5) No message requiring voters to verify the displayed vote and their PAT; 6) Not clear whether identification devices do or don't store data on their internal, persistent memory; 7) Not specified how keys of electronic devices for data encryption and integrity are generated, stored and accessed; 8) Electronic devices shall not have external components, however they do use external USBs; 9) Not clear how the display and printer glass shall dis- and allow taking pictures; and 10) No independent audit performed. *Social challenges* include, e.g. 1) Use of electronic voting motivated politically rather than objectively; 2) Remaining vote-buying allegations; and 3) Wrong mental model regarding security guarantees. *Organisational challenges* include, e.g. 1) Lack of reports (e.g. exit questionnaires, cross-checking of biometric data); 2) Partially failed recruiting, training and educational processes; and 3) Extremely limited time and unreasonable deadlines. Last, but not least, *technical challenges* include, e.g. 1) Lack of authenticity verification of ID cards; 2) Lack of trust model regarding vote secrecy and privacy for voters' biometric data; and 3) Lack of implementing design principles that increase voters' motivation to verify their PAT.

6 Conclusion

In this paper we described the introduction of electronic voter identification, voting and tallying in the parliamentary elections on April 25, 2021 in Albania. Despite being rated as a successful achievement by the Central Electoral Commission, various challenges of legal, procedural, social, organisational and technical nature were identified. In particular, the absence of independent auditing, lack of transparency, and lack of trust model regarding vote secrecy and integrity potentially compromised the election principles. Therefore, before proceeding further with the use of technology in elections, it is strongly recommended to conduct a thorough analysis, whether electronic voting is the adequate solution to electoral challenges in Albania. In that case, consequently, future work shall focus first on suggesting adequate improvements to the herein highlighted challenges.

⁶¹ <http://kqz.gov.al/2021/04/25/deklarata-e-komisionerit-shteteror-te-zgjedhjeve-zilirjan-celibashi-ora-1900/>, last accessed 18.06.2021

Use of innovative technologies in the electoral process in Armenia

Ardita Driza Maurer,¹ Justin Nettmann,² Rafik Grigoryan³

^{1,3} Independent experts; ² ICT Elections Management Consultant
info@electoralpractice.ch; justinnettmann14@gmail.com;
rafik.grigoryan@sigmalaw.am

Abstract. This paper¹ presents an overview of the use of ICT-backed solutions in the electoral processes in Armenia and their possible future development. It discusses questions that arise in relation to the current use of ICT and to the envisaged developments which may be of general interest.

Keywords: Armenia, Use of ICT in Electoral Processes, Voter Authentication Device (VAD), E-registers, Internet Voting, Elections Automated System, E-identification, E-government.

1 Introduction

1.1 Electoral system

Armenia is a small country (area – 29,8 km², population – 2.9 million, of which 2.5 million have voting rights) geographically located in the South Caucasus and generally considered geopolitically European. The citizens of Armenia vote in elections and referenda, at the state and local self-government level. The Constitution foresees the regular and extraordinary election of the National Assembly and elections of local self-government bodies (election of Head of Community and Member of Community Council of Elders). They are regulated in detail in the Electoral Code (EC). The Constitution further provides for direct participation in the administration of community affairs through the local referendum. At the national level, it is possible to vote in a referendum to modify the Constitution, on a draft law submitted by popular initiative and in a referendum on the membership of the Republic of Armenia in supranational or international organisations as well as on changes of territory.

¹ This paper was prepared with permission from UNDP following the Feasibility Study on Innovative Technologies for Electoral Processes in Armenia, produced in the framework of “Electoral Support Project in Armenia” funded by the Government of Japan and implemented by UNDP. The authors of this paper are the main authors of the feasibility study. The views expressed in the paper are those of the authors and do not necessarily represent the views of UNDP.

The principles of secret ballot and universal, equal, free and direct suffrage apply equally to elections and referenda. So do other relevant principles, namely the mandatory and periodic nature of elections and the publicity of elections (Articles 1-8 EC).

Armenia is a parliamentary republic since the 2015 constitutional amendments and the election of the National Assembly is the single nationwide election. It is also the most sophisticated one. The electoral system, in force from 1 June 2016 until April 2021, was quite complex.² The system was changed on 1st April 2021 when the Parliament approved amendments to the EC introducing a pure, one-tier proportional system, easier to handle also for the voters. The National Assembly that came out of the snap election of 20 June 2021 was elected through the new proportional system, with one multi-mandate constituency covering the entire territory of the Republic. Each party (or alliance of parties) running in the elections nominates one nationwide electoral closed list of candidates. The electoral list of the party (alliance of parties) includes not less than 80 and not more than 300 candidates.

1.2 Election administration

Elections are administered by a three-tier system comprising the Central Electoral Commission (CEC), 38 territorial electoral commissions (TECs) and 2,010 Precinct Election Commissions (PECs). A distinct governmental agency, the Police Passport and Visa Department, manages the State Population Register from which is drawn the electoral register. It updates the register and submits it electronically to the CEC for posting on the commission's website.³ A search engine is available on the website of CEC. The Police Visa and Passport Department is also responsible for eliminating any inaccuracies in the register and notifying the applicant either directly or via a dedicated website,⁴ which, technically, is not part of the elections management system.

² It provided for a minimum of 101 members of parliament (MPs) to be elected through a two-tier proportional system. In that system, each party (or alliance of parties) participating in the elections nominated one electoral list of candidates, which consisted of two parts. The party (or alliance of parties) compiled regional electoral lists from the candidates included in its national electoral list. A candidate could be included in only one regional electoral list. During the National Assembly elections, 13 districts were formed in the Republic of Armenia (4 in Yerevan and 9 in the regions). Half of the seats were assigned through the nationwide list and the other half through territorial lists submitted in each of 13 territorial districts. According to OSCE/ODIHR EOM Report of 2017 Parliamentary elections, voters had difficulties in understanding the then-new voting process which led in some cases to group voting and attempts to influence voters as well as to procedural omissions during the counting.

³ www.elections.am

⁴ www.azdarar.am

1.3 Current use of ICT in elections

Use of ICT is envisaged as a mean for ensuring publicity and transparency of organising and holding elections and should be done in a manner that respects security, smooth operation and proper exercise of powers (Article 8§3 EC). Currently, several solutions based on so-called information and communication technologies (ICT) are used in election administration in Armenia.

Voter Authentication Devices. Use of ICTs during election day started mainly in 2016 with the adoption of the new Electoral Code whose main novelty was the introduction of Voter Authentication Devices (VADs). These devices have been used for the last three parliamentary elections (2017, 2018 and 2021). Even though the parliamentary elections held in April 2017 were generally well-administered, important shortcomings remained, including vote-buying and misuse of administrative resources which contributed to an overall lack of public confidence and trust in the election. Following the Velvet Revolution and democratic changes in 2018, the following two snap elections held respectively in 2018 and 2021 were claimed to be more democratic and transparent.

The introduction of VADs is one mechanism that not only assists in the identification of a voter but also aims to address the issue of trust in elections. International observers of the 2017 election noted that the introduction of the VADs was welcomed by most of their interlocutors as a useful tool for building confidence in the integrity of election day proceedings, while also recognizing a number of issues, such as the late development and delivery of the VADs which led to a limited time for testing of equipment and training of the VAD operators.⁵ In contrast, international observers had noted in previous elections a number of serious violations including multiple and proxy voting, impersonation and issued recommendations to address them.⁶ In all elections held before 2018, voting rights were allegedly infringed, and so public confidence in the outcome was very low. It was in this context that the Government of Armenia decided to implement new ICT-backed solutions to reduce the risks of double voting, namely by preventing undue use of credentials/identities of Armenians who are either living abroad or out of the region where they are registered and where their polling station is located. Additionally, the VADs were intended to also put an end to widespread allegations that the Government included names of dead voters on the voting lists.

VADs are considered a means to help preventing multiple voting, impersonation, and fraud and, according to observers, have helped instil trust in recent elections.⁷

⁵ See the OSCE/ODIHR Election Observation Mission Final Report on parliamentary elections 2 April 2017, p. 10 as well as p. 5, https://res.elections.am/images/doc/osce02.04.18_en.pdf

⁶ See the OSCE/ODIHR Election Observation Mission Final Report on parliamentary elections 18 February 2013, page 21, <https://www.osce.org/files/f/documents/a/d/101314.pdf>

⁷ See OSCE/ODIHR as well as Council of Europe PACE reports of the parliamentary elections of 2 April 2017 and of the 9 Dec. 2018 early parliamentary elections, available at:
 – <https://www.osce.org/files/f/documents/6/7/328226.pdf>
 – <https://www.osce.org/files/f/documents/b/7/413555.pdf>

They have been successfully used since 2017. Despite the identified shortcomings in the 2017 elections, the use of VADs was claimed to be successful: nine cases of attempted multiple voting were identified by the VADs and thus prevented.⁸ The same VADs were also successfully used in the 2018 and the 2021 parliamentary elections, without any changes to the procedures.

The Voter Authentication Devices are used on voting day to authenticate voters and to collect fingerprints for post-election controls in case of alleged multiple voting. VADs verify the identity of the elector by reading the machine-readable zone (MRZ) of the identity document and comparing it with the electronic list of electors. Manual registration is foreseen for certain documents. In a next step, the VAD takes the fingerprint. The VAD then prints a voting pass which enables the voter to register as voting by signing (on paper) next to his or her data in the column envisaged for it on the list of electors.

It is to be noted at this juncture that the VADs are not connected to the internet. No data transfers, neither push nor pull, are done at any point during election day. The only data being sent happens after the election, whereby a scanned copy of the voters' list, including voters' signatures, is published on the CEC website. The possible control of the fingerprints entered in the system is done after the election, at the TEC level, by means of special software, in case of claims or disputes.

From a technical perspective, the VADs access the voters' data in the field by means of accessing a memory card which is prepopulated with the voters list. These memory cards are configured prior to the election and the data residing on the memory cards comes from the Central Voter Information System or commonly known as the CVIS. The CVIS is a web-based application which makes use of several processes to prepare the master data files for each VAD. Once the master data file is prepared making use of a replicator the data files are replicated and distributed to the TECs and finally to the polling stations according to CEC processes. The CVIS is very relevant in this entire cycle as not only does it prepare the master memory cards, but also for the consolidation of data collected from polling stations VADs after the election. This data is used to analyze voter turnouts and to produce reports as required by the CEC. The CVIS is an indispensable component of the voter authentication system. Major developments of both VADs and CVIS depend on the provider whereas the election administration oversees the rest.

Internet voting. Internet voting is used in a very limited capacity. The small size of the country and of its population, and the fact that polling stations are located very near each other, may be a reason why there doesn't seem to be demand for internet voting inside the country. The use of internet voting is however of interest to the Armenian broad diaspora. The law on the election of the President of the Republic of

– <https://assembly.coe.int/nw/xml/XRef/Xref-XML2HTML-en.asp?fileid=23748&lang=en>

– <https://assembly.coe.int/nw/xml/XRef/Xref-XML2HTML-en.asp?fileid=25251&lang=en>

⁸ See OSCE/ODIHR Election Observation Mission Final Report on parliamentary elections of Armenia of 2 april 2017, https://res.elections.am/images/doc/osce02.04.18_en.pdf

1996 provided for the first time the right of expatriates to vote from abroad.⁹ Initially voting from abroad took place in diplomatic and consular representations: first at the 1996 election and then at the 1998 snap presidential election. In both cases it was considered successful.¹⁰

The 2005 constitutional amendments, which allowed dual citizenship, abolished however the voting rights of the large diaspora. The Electoral Code adopted in 2011 allowed voting from abroad only for citizens working in the diplomatic service in representations of Armenia abroad and members of their families residing abroad with them and having the right to vote as well as for a few other groups (the military, students and employees of certain firms registered in Armenia).

An internet voting system was introduced first at the parliamentary election of 6 May 2011 and then at the presidential election of 18 February 2012.¹¹ Noting that remote electronic voting is controversial, Venice Commission and OSCE/ODIHR have recommended that Armenian authorities carefully examine the need for Internet based voting against the alternative of organising polling stations at the consular offices on election day for this small group of voters.¹²

Whereas the Electoral Code foresees the use of internet voting to enable several groups of Armenians living abroad to participate in elections of the National Assembly (diplomatic and consular personal and their families, military, students, employees of certain firms registered in Armenia and located abroad, and members of their family), in practice internet voting has only been offered to diplomats and their families. The organisation and duration of internet voting is regulated by the CEC through a 17 June 2016 decision which establishes internet voting as an alternative channel, complementary to voting in polling station. It is organised between the 9th and the 7th day before election day. According to the EC, the CEC is “obliged to establish such terms for electronic voting that would ensure free expression of the will of voters and secrecy of voting”. The decision clarifies questions related to the lists of electors and the generation, sending and handling of personal codes, in a secure manner. It stipulates that the vote should be secret, and the voter has the possibility to vote multiple times, the last vote cancelling the precedent one. The system notifies the voter that he/she voted. After the vote ends, members of CEC enter their individual codes in alphabetical order of their surnames and the system produces a protocol of voting results which is then signed by the members of CEC. Results are entered in the “Elections” automated system. Conditions for the invalidity of ballots are foreseen. Recounting is foreseen as being a “counting the results of electronic voting for the second time”

⁹ Law on the Election of the President of the Republic of Armenia (adopted on April 30, 1996), Article 2, www.arlis.am.

¹⁰ Some 15'000 Armenians abroad participated which attested of their great interest in the election. See: Hamazasp Danielyan, “Internet voting in Armenia. Invisible Innovation Analysis” (in AM), https://www.osf.am/wp-content/uploads/2015/10/Hamazasp_Danielyan_PP.pdf

¹¹ *Ibid.* Some 195 voters participated at the parliamentary election and 228 did so at the presidential election.

¹² Venice Commission, OSCE/ODIHR, Joint final opinion on the Electoral Code of Armenia, 26 May 2011 (CDL-AD(2011)032), [https://www.venice.coe.int/webforms/documents/default.aspx?pdffile=CDL-AD\(2011\)032-e](https://www.venice.coe.int/webforms/documents/default.aspx?pdffile=CDL-AD(2011)032-e)

with the mandatory participation of CEC members and optional participation of observers, journalists, representatives of parties and the candidates. It should be impossible to link the name of the voter and the vote, at the counting stage. The source code of the e-voting program is required to be published on the website of CEC. No information is available on past controls of the source code and their conclusions.

Other ICT-backed solutions used in elections. An automated system called “Elections” centralises information which feeds into the CEC website. It includes several stand-alone systems which “interact” with the CEC. Other ICT-backed solutions that aim at ensuring transparency include cameras that film participation as well as counting and a live webcast feed on election day. These were successfully used in 2016 and continue to be used. Live feed transmissions are recorded on the central servers for the purpose of being used to assist the judiciary with solving possible claims or objections. Such transparency measures do not apply to some specific groups of voters (e.g. the military).

Additionally, signed voters’ lists are published on the internet after the elections. These measures serve the same purpose as VADs: to prevent and detect fraud, multiple voting and impersonation. However, they also raise questions of compliance. Namely the filming of participation and publication of handwritten signatures question compliance with requirements on confidentiality and protection of personal data.

E-identification and other e-government developments. In addition to ICT solutions specific to elections, Armenia has been creating an ecosystem of web-based solutions for interactions between citizens and the administration.¹³ The Government has an agenda of digitalization of state procedures and public services. A new “Digitalization Strategy”, summarizing previous decisions, was adopted by the Government on 11 February 2021. Several Government decisions (e.g. 31.08.2015 № 1093-N, 19.12.2019 № 1849-N, 25.05.2017 № 572-N and 26.12.2013 № 1521-N) introduce security, interoperability and other standards and requirements that are relevant for electronic information systems and web portals of state agencies.

Solutions for electronic identification have been available to Armenian citizens since 2009. The e-ID solution is foreseen for signing referendum demands, for instance. To gain access to the e-ID, citizens need to have an ID card reader and special software installed on their PC, which are tedious and costly procedures. As a result, digital identification is usually used only if people are obliged to do so. Sometime both e-ID and hand signatures are possible. For example, the E-Request portal¹⁴ allows people to apply to any state body electronically either by introducing a document signed by electronic signature or by providing a scanned copy of a paper document which is hand signed. In 2018, a Mobile ID solution was introduced, which makes digital identification and signing possible with a special USIM card inserted in the mobile phone. Work is ongoing on a smartphone software solution to enable the users of Android and iOS smartphones to sign documents, as well as to login and gain ac-

¹³ Several solutions exist already and can be accessed through the portal www.e-gov.am

¹⁴ www.e-request.am

cess to e-services just by downloading an application and passing a registration procedure. No additional hardware would be needed, and the solution would be compliant with the standards of security of the EU eIDAS regulation.

Different distributed Armenian e-government public services are interconnected and exchange information via the Government Interoperability Platform (GIP) using secure means. A Governing Authority (EKENG) coordinates the GIP, creates and enforces standardized security policies and provides technical support to members of the governmental data exchange. The election management system and other election-related ICT solutions are not part of the GIP though.

1.4 Extending the use of ICT in elections

The Government of the Republic of Armenia has a strong political will to strengthen democratic processes and enhance the procedure and quality of the organization of elections. Both the Government and the Central Electoral Commission look with interest in the possibility of moving more services to a digital technology platform, including online spaces, as part of the Government's strategic planning. The COVID-19 pandemic has strengthened such interest.

The future of ICT use in elections is currently being studied. A big package of amendments to the Electoral Code was adopted in May 2021 (it did not apply to the election held on 20 June 2021). It addresses mainly past recommendations of international organizations and observers over the years. Depending on the decisions to be taken on the future development of ICT solutions in elections, other important amendments of the EC are to be expected.

The Electoral Code puts the use of ICT in elections under CEC responsibility. The EC currently enables the CEC to introduce e-voting and e-counting in small scale pilots during local elections. CEC is expressly enabled to organise and hold pilot projects involving ICT during elections of local self-government bodies (only) and at a small scale with no more than 2000 electors per community and no more than ten communities annually. Any use of e-voting and e-counting at a larger scale or during national elections or referendums, requires a specific legal mandate, which implies changes in the EC, whose modification should be endorsed by a qualified majority of the Parliament.

The international community has been assisting Armenia with the strengthening of democratic institutions, including election administration and use of ICT in electoral processes. For instance, UNDP Armenia has been assisting with the introduction and development of ICT solutions for elections since 2016 when Armenia decided to introduce Voter Authentication Devices (VADs) to assist in instilling more trust in the voting process. UNDP's contribution extends to IT infrastructure, hardware components, development and maintenance processes, professional trainings including an e-Learning platform and a Training and Resource Centre as well as to long term strategic planning at the CEC.

2 Trends and questions

Armenia, like other countries, is in the process of expanding its e-governance footprint. Its electoral authorities continuously look at streamlining and implementing more effective and efficient mechanisms to enhance the various electoral processes for the stakeholders involved. Building on the current state of election technology systems used in the country, the electoral authorities wish to explore possible technology solutions and systems which could be introduced in a structured manner in the future, and which would follow international standards and good practice.

One of the developments envisaged by the authorities is the feasibility of introducing e-voting in polling stations and/or e-counting. In a first step, the authorities ponder the development and transformation of some of the VADs into electronic voting machines (EVMs). Alternatively, they consider the introduction of totally new EVMs and/or of e-counting technology. Additionally, complementing and upgrading the elements of the Elections management system is being envisaged. There are no plans about extending the use of internet voting.¹⁵

When considering the further development of ICT in elections, several questions arise about both the current and the envisaged systems, including their purpose, their integration into the current election management system, their security, maintenance, development, and, finally, the sustainability of such developments. We present an overview of some of these questions which may present a general interest. Any detailed study of the current and future development of ICT in elections requires combined expertise from different fields, including electoral legislation, election administration, IT, socio-political aspects and international cooperation.

2.1 VADs

The Armenian CEC has opted to introduce a voter authentication device at polling stations in an attempt to reduce possible double voting as well as to strengthen the overall confidence in the elective process. From a purely technical perspective, while the idea was very sincere and the approach a step in the right direction, the question of what else could be done with this technology to further enhance and strengthen the process of voter authentication and ensure that it achieves the objective of reducing double voting begs.

The VAD, in its current form, only scans and saves the voter's fingerprint, doing no fingerprint algorithm validation/verification check on presented fingerprints. It can be argued that if biometric data is readily available, it could further be used to verify and validate voter data on the spot. This would not only deter people from taking the chance of defrauding the process but would certainly enhance the transparency aspect

¹⁵ In the context of recent reflections about a possible extension of use of ICT solutions in elections, UNDP Armenia conducted a feasibility study "Innovative technologies for the electoral process in Armenia" to assess and evaluate options of e-voting technologies, to identify international good practice and to discuss cases involving e-voting and other innovations, analyse pros and cons and propose solutions that best suit the needs and existing possibilities of the Republic of Armenia (see fn. 1).

of the elective process. By implementing a fingerprint analysis process brings to the fore the question relating to the possible centralising or de-centralising of voter data verifications and management thereof as well as related accuracy and security considerations, among others.

From a legal perspective, the VAD system is one of those “legacy systems” which despite being trusted and used through several elections, is not satisfactorily regulated. A closer look into the VAD regulation shows that it contains important gaps. It does not envisage the case of VAD malfunctioning or of erroneous answers provided by the voter authentication software. Legal provisions currently refer to a well-functioning system, only. How to handle possible malfunctioning is not clear. Such gaps are problematic especially in countries like Armenia, where regulation is very detailed, and issues are treated in a very formalistic way: if a situation is not foreseen and dealt with in the electoral regulation, it may fall beyond the control of authorities. A better way of handling these situations could be for the regulation to establish objectives that a solution should fulfil (instead of attempting to regulate any possible issue) and for the competent authority (CEC, judge, etc.) to evaluate whether the solution fulfils the objectives, and do so most probably in cooperation with academia, experts, etc.

VAD regulation does not address dispute settlement issues either. For instance, it is not clear what happens when a voter contests the “decision of the VAD”. This is a general preoccupation. Regulation of current and future ICT solutions should pay attention to dispute resolution mechanisms and remedies. These are even more important when the vast majority of people do not understand the technology solutions in place. It should be done in the light of the specific ICT solution and of the possibilities of verification offered to voters and other stakeholders. Criminal sanctions may need to be further developed to address ICT-related violations.

Another legal preoccupation with respect to VADs is the lack of a clear definition of competences of the various state bodies involved in the collection, storage, and use of personal data, including biometric ones like the fingerprints or the voters’ signatures.

Yet another issue is the publication, at the end, of sensitive data on the web. In this case, it’s the lists of voters which includes voters’ hand-written signatures which is made public, for transparency purposes. Although such publication is quite specific to Armenia, it points to a general tension that exists between requirements of transparency and publicity on one side and those of confidentiality of participation and of the signature on the other. In the case of Armenia, international observers have repeatedly noted that the publication of voters’ signatures is at odds with commonly shared principles of confidentiality and secrecy. Yet, in Armenia, such publication is considered to be an efficient measure for discouraging and discovering potential fraud in participation.

2.2 Internet voting

Internet voting in the context of a parliamentary election in Armenia is only offered to diplomats and their families. Technically, the capabilities to make use of this

technology on a larger scale which meets all legal requirements are not available in Armenia. The long list of high-risk technical issues surrounding internet voting often sees EMB's shy away from it. There are many technical and nontechnical issues surrounding its use and most of these issues are so high risk that it is not worth deploying in fear of systems being compromised. The overall sentiment is that internet voting cannot be trusted. The issue of data and system security seems to be the top preoccupation.

If it were that internet voting was being considered, what are some of the more serious technical points that should be addressed as a minimum? Addressing security by adopting a strategy that considers cybersecurity in totality is an important point. Ensuring that the technology meets all legal, operational and system requirements is another one. The adoption and implementation of industry accepted failover technology systems and strategies, the inclusion of intelligent technology systems to monitor integrated internet data, system and services, the inclusion of defined system protocols that ensure good ICT governance and ensuring that ICT in the organisation drives the overall internet data, system and services in accordance with all provisioned directives are other important points to consider.

From a legal perspective, it is to be noted that the Armenian current regulation of the limited use of internet voting remains quite high-level, which is problematic because it lacks detailed requirements on how the general principles can be implemented and respected by the internet voting system. Additionally, internet voting regulation contains no provision on control of compliance of the system with the requirements. The consequences of non-compliance are not discussed either. There is however a detailed provision on multiple voting which is introduced as a mitigation to potential coercion and violation of secrecy as internet voting takes place from an uncontrolled environment.

Another observation is that the development of internet voting in Armenia illustrates the tension between universality and equality on one side and security on the other. By enabling a limited group of electors (diplomats) to vote through the internet voting channel (the limitation to this small group being a risk management/security measure), the competent authorities treat unequally electors who are in the same situation (military, students, etc. living abroad) but who, unlike diplomats, are offered no effective voting channel.

2.3 Electronic voting and counting machines

Technology solutions in Armenia are increasingly being architected, designed, and implemented into current elective processes and procedures as the authorities realise that technology can play a big part in the process, from online training services to electronic voting systems.

The idea of implementing a centralised elections solution with the deployment of electronic voting systems has been analysed and documented as part of a feasibility study that was conducted by UNDP.¹⁶ The CEC and other stakeholders are interested

¹⁶ See fn.1

in looking at the viability of transforming VADs into EVMs and at testing e-voting on EVMs. A possible ICT roadmap to follow would be one that defines a strategy that would see the CEC make use of a redesignated VAD into a EVM, testing the concept and then defining a longer terms strategy which could either see a completely different technology approach being followed or one that makes use of redesignated VADs.

The key technical points that must be reviewed when looking at implementing electronic voting machines include the following. It is always important to define all the technical requirements of the device in line with the legal directives. In the Council of Europe region, the acceptance of EVMs is increasingly linked to the level of verifiability that they offer. One must ensure that the technology stack of the device meets all the technical requirements – including but not limited to system security, device security, system and user interaction, ease of use, technical support, and availability thereof etc. The technology should be sustainable: the EMB must implement a technology which can be internally supported after the initial implementation for a sustained period. The technology must fit in with the defined technology roadmap (if a defined technology roadmap is not available the EMB must define one). The overall technology used and that which governs/enables it must be tested and meet the basic technology best practices and standards that are globally accepted. A defined technological methodology (ITIL as an example) must be used to implement the system or any system related changes. Upskilling maybe necessary if new technologies are implemented. This is important when knowledge transferring takes place.

Additionally, the Armenian case shows that authorities should be clear about the needs that are being addressed by new ICT. For instance, the use of EVMs alone cannot control that a person votes only once, which seems to be one of the main preoccupations in Armenia. More generally, technology alone cannot help to solve trust issues: indeed, trust in the electoral authorities is a precondition to introducing and developing e-voting solutions; technology is by no means a ‘silver bullet’ Technology must be seen in the context of a mechanism which is introduced to solve problems not create them. While the use of technology helps to assist in elections, it still needs to be governed.

2.4 Centralised Elections System

Centralised Elections Systems can range from centralised or clustered systems, services or databases. Centralised systems for the most part help when consolidation of services, data or overall ICT outputs such as reports need to be done at a central level, whether to save costs or to manage resources as an example. In the case of the Armenian CEC, all ICT activities and outputs are done at a central level in Yerevan. No decentralised ICT activities happen outside of Yerevan as the CEC has limited ICT resources available. Should the CEC be looking at centralising systems, what should they maintain as a standard?

When looking at developing or implementing centralised solutions, the EMB must be skilled enough to support such solutions. The ICT maturity level within the EMB must be high, ensuring that they can support the operational environment without compromising any of the processes.

Strategic developments must be aligned to the overall strategy of the EMB. Systems that are developed or implemented must take all aspects of ICT good governance, security, system failover, networking etc into consideration. All associated systems which are integrated or are going to be integrated must be compatible, flexible, and sustainable. Cross platform friendly base must be identified and developed or implemented.

A clean design methodology must part of any solutions being developed or purchased. The last thing one wants is to develop an entire centralised solution/system only to learn afterwards that the database that was chosen is not easily integrated into other databases.

2.5 E-government solutions

Armenia has developed a number of innovative e-gov systems and solutions, which have assisted in many different aspects of life in Armenia. The data produced within the various civils systems can be and is in some instances consumed by other government institutions. One of these institutions is the CEC. The CEC relies on another state institution to provide it with the voters' roll and associated data.

Those EMBs that rely on other institutions to provide information or data such as is the case with CEC must always ensure that they are acting in line with directives specific to the EMB. The EMB must have ICT processes and procedures in place to verify and validate information and data which is provided by other sources.

The EMB should define a sound ICT good governance structure and ensure that institutions that are sources of information or data comply with the structures. Structured processes in the institutions that provide information and data to EMBs must be enforced. The EMB should be entitled to be part of all information or data sharing between the various institutions. The EMB should have the ability to enforce best practices and good ICT governance processes within institutions that deliver services to it. This is important when it comes to ensuring transparency of any elective process.

2.6 June 2021 snap elections under COVID

Use of ICT has been considered in the context of health restrictions related to the COVID19 pandemic as the June 2021 snap elections were held under such restrictions. However, no changes in technology were introduced and social distancing, mask wearing, and hand and surfaces sanitization were privileged.

At the beginning of the COVID pandemic the state of emergency was declared by Decree of the Government on 16 March 2020, initially for the duration of 30 days,¹⁷ subsequently prolonged until 11 September 2020¹⁸ and validated by the National Assembly. In particular, the following restrictions to fundamental rights, relevant for elections, were introduced: restrictions on the freedom of movement, of assembly,

¹⁷ <https://www.venice.coe.int/files/EmergencyPowersObservatory//T06-E.htm>

¹⁸ <https://www.venice.coe.int/files/EmergencyPowersObservatory//T10-E.htm>

and of the press (later repealed). There were also restrictions on economic activities. In August and September 2020, restrictions were softened. The National Assembly introduced new amendments to the laws on Protection of Population in Emergency Situations and Provision of Sanitary-Epidemiological Security of the Population of Republic of Armenia which allowed for appropriate measures against COVID-19 without declaring the state of emergency. On 11 September the state of emergency was not prolonged and thus terminated.

By Decision no. 1514, the Government declared quarantine on 11 September 2020 for four months, until January 2021 and then extended it to 11 July 2021. The quarantine decision is a prerogative of the Government, which does not have to be approved by the National Assembly. Government's decision no. 1514 was completed by a decree of the Ministry of Health. Government's decision no. 1514 prescribes the use of personal protective equipment (PPE). The exact kind of PPE is foreseen by the Decree no. 23 of Minister of Health, adopted on 11 September 2020. Other restrictions introduced by Government's decision no. 1514 relate to travelling to and from Armenia, to some restrictions in penitentiary bodies, schools, army.

The legal regimes of state of emergency and of quarantine have different consequences on elections. Whereas holding elections or referendums is prohibited during the state of emergency or martial law, the quarantine regime allows holding elections or referendums. However, it introduces new rules which may affect electoral rights such as the wearing of masks in closed places or keeping physical distance of at least 1.5 meters and wearing masks during assemblies.

The very detailed nature of the Electoral Code which regulates the tasks of CEC and its room for manoeuvre suggests that the CEC's mandate to ensure the exercise of the right of suffrage does not enable the CEC to introduce new modalities that are not expressly foreseen in the EC. The application of quarantine related measures to restrict electoral rights is subject to a legal basis approved by Parliament. Furthermore, any new usage of ICT (e.g. to mitigate COVID related risks) would require the CEC to obtain a clear legal mandate for that.

June 2021 elections were held under COVID. The restrictions were maintained during the election process but were largely ignored during the pre-election phase, also by the candidates. The situation was different on election day. Entering the polling station was allowed only with wearing masks and if a voter didn't have a mask, the commission was providing it. The commission, observers, party and candidates' agents and journalists were strictly wearing masks inside the polling station. There were sanitizers inside the polling stations. The VAD's were being cleaned with special sanitizers the water content of which was low, to prevent water ingress of the scanner. There was no use of pen or other device to write on the ballot because the voter was given separate ballots for each nominated party and could vote by putting the ballot of the chosen party in the envelope. Eventually, election day activities had no significant influence on the pandemic situation, as showed by the statistics published by the Ministry of Health of Armenia.¹⁹

¹⁹ <https://ncdc.am/coronavirus/confirmed-cases-by-days/>

3 Final remarks

We conclude with a few general suggestions for situations where modernisation of electoral processes through increased use of ICT-backed solutions is considered.

The will of the authorities to modernise the electoral system does not remove the necessary identification of the actual needs of stakeholders, including of specific groups of voters (women, the sight-impaired, the expatriates), of staff, observers, parties, etc. The development of ICT should be needs-oriented and not an end. Identified needs should be carefully considered and addressed already during the design phase of the ICT solutions.

Legal regulation should be the driving force behind the use and development of ICT (and of any other solution) in elections. Legislation should dictate the values that ICT must respect, not the other way around. However, this is a challenge, in any country. Solving it requires, first, clarity about all legal principles that apply to elections, including but not limited to the right to universal, equal, free, direct and secret suffrage, periodic elections and publicity of elections as well as national or local principles. Second, it requires detailed requirements that ensure respect for the principles. Such requirements represent as many objectives that any solution used in elections, including ICT ones, must fulfil. Finally, it requires a good understanding of other requirements, coming from “outside” the traditional electoral legislation field, which are important when ICT is used. These include data protection, information and system security or cybersecurity, or international cooperation in fighting cybercrime requirements, among others.

In order to build a sound regulatory framework that guides the development of ICT in elections, recommendations from international bodies may be useful. These include the Council of Europe Recommendation CM/Rec(2017)5 on standards for e-voting and its accompanying documents, the OSCE/ODIHR Handbook for the observation of new voting technologies or the Council of Europe guidance documents on the application of the Budapest Convention on cybercrime and of the Convention 108+ on data protection to elections as well as European Union guidance documents on the application of the GDPR to elections.

The envisaged extension of use of ICT offers the opportunity to also evaluate and improve the regulation of existing “legacy” systems.

When implementing any new technologies, it is advisable to ensure that the EMB is not vendor locked, that knowledge transfer continually takes place and that support contracts and any other contracts are well defined.

Usability and Voter Perception

Voter Perceptions of Trust in Risk-Limiting Audits

Asmita Dalela¹, Oksana Kulyk¹, and Carsten Schürmann¹

IT University of Copenhagen, Denmark
 {asmd,okku,carsten}@itu.dk

Motivation: A Risk-limiting audit (RLA) [2] is a post-election auditing technique that is gaining increasing popularity, because it can automatically correct a wrong election outcome, while being very efficient especially if the margins are wide. The purpose of RLAs has been to strengthen public confidence in the election, first, because they can be integrated into existing election processes, for example, in Denmark, where the result of the first (rough) count can be verified during the second (fine) count [3]. Second, some of the ceremonies surrounding RLAs can be turned into public events, such as the dice-rolling ceremony used to create entropy to select a random sample.

The question, however, remains open, whether RLAs really strengthen public confidence. While the theory behind the audits is sound, the resulting sample size is often very small (e.g. 523 ballots audited in an RLA given a total of 393,826 cast votes in the 2020 election in Denver, Colorado). Thus it remains an open question whether the voters would find an audit with such a sample size convincing or not.

Method: To answer the research question *if RLAs really strengthen public confidence in the outcome of an election*, we conducted a user study with 105 randomly chosen US residents across all demographics using the Prolific platform. We study several hypothesis, most importantly:¹:

$H_{1,1}$ When asked about their opinions about which number of ballots should be selected for auditing, the participants provide a number higher than the one prescribed by the RLA methodology.

$H_{1,2}$ Participants' confidence in the audit results changes when they are informed about the number of ballots selected for auditing

Results: When asked which number of ballots the participants would prefer to be audited, this number tended to be magnitudes higher than the actual number required by the RLA for most of the participants (see Figure 1). The sign test has confirmed that the difference between preferred and actual number of ballots is significantly different from zero ($p < .001$, 95% CI for median difference between preferred and actual ballots (as percentage of total ballots) is [3.23%, 16.1%]), thus, **$H_{1,1}$ is confirmed.**

While the majority of the participants (70%, 74 out of 105) had a positive attitude towards conducting RLAs, choosing either “maybe yes” or “definitely

¹ For a more detailed description of the methodology and the rest of tested hypotheses and conducted analyses, see the extended version of the paper [1]

yes” as the answer to the question whether their confidence in the election result would increase after an RLA, only 44% provided a positive answer to the same question asked after presenting the number of audited ballots to the participants. Figure 1 shows the distribution of changes of participants’ answers. The Wilcoxon signed-rank test shows a significant difference between the “before” and “after” answers ($p < .001$, $Z = -4.47$, effect size $r = .33$, moderate), thus, $H_{1,2}$ is confirmed.

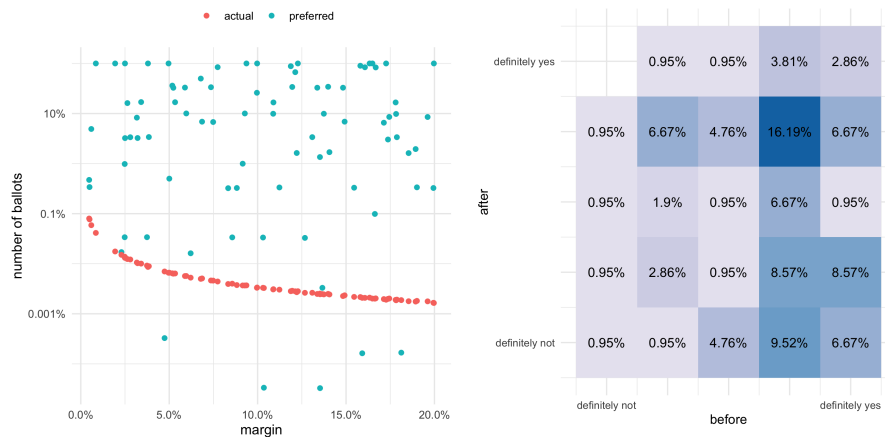


Fig. 1: Left: Preferred vs. actual number audited ballots (as percentage of total ballots) depending on the margin. The scale is logarithmic. Right: Percentage of participants for each case of confidence change.

Conclusion: We recognize the value of RLAs in confirming the integrity of the election result. However, our results show that as a measure to create trust, they are not sufficient by themselves, and additional measures such as voter education need to be considered. While this study is the first one to investigate this issue, follow-up work is needed to better understand the factors influencing voter’s trust and the effectiveness of various ways one can educate voters about RLAs or raise trust via other measures.

References

1. Asmita Dalela, Oksana Kulyk, and Carsten Schürmann. Voter perceptions of trust in risk-limiting audits. <https://arxiv.org/abs/2109.07918>, 2021.
2. Mark Lindeman and Philip B. Stark. A gentle introduction to risk-limiting audits. *IEEE Security & Privacy*, 10(5):42–49, 2012.
3. Carsten Schürmann. A risk-limiting audit in Denmark: A pilot. In *First International Joint Conference on Electronic Voting and Identity (E-Vote-ID’16)*, pages 192 – 202. Springer Verlag LNCS 10141, October 2016.

Usable Verifiable Secrecy-Preserving E-Voting

Oksana Kulyk¹, Jonas Ludwig², Melanie Volkamer², Reto E. Koenig³, and Philipp Locher³

¹ IT University of Copenhagen, Rued Langgaards Vej 7, DK-2300 Copenhagen S

² Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen

³ Bern University of Applied Sciences, Quellgasse 21, 2501 Biel

Abstract. In this paper we propose the usage of QR-Codes to enable usable verifiable e-voting schemes based on code voting. The idea – from a voter’s perspective – is to combine code voting proposed by Chaum with the cast-as-intended verification mechanism used e.g. in Switzerland (using a personal initialization code, return codes per option, a confirmation code and a finalisation code); while all codes to be entered into the e-voting system by voters are available as QR-Code (i.e. one personalised QR voting code per voting option and one personal confirmation QR-Code). We conduct a user study to evaluate the usability and user experience of such an approach: both the code sheets and the election webpage are based on usability research in this area but adopted for our idea. As our proposal performs good wrt. usability, we discuss how such usable front-ends enable more secure e-voting systems in respect to end-to-end verifiability and vote secrecy.

1 Introduction

Developing e-voting systems with high level of security using non trustworthy voting clients (i.e. malware infected voting machine or personal computer equipment in case of remote e-voting) is very delicate and existing proposals come with severe usability challenges. In the most recent past, this challenge – both in academia as well as in conducting e-elections – was addressed with respect to vote integrity. Several cast-as-intended verification methods have been proposed to allow voters themselves to perform checks to verify that their vote has not been manipulated; in particular that it has not been manipulated by their compromised voting component. There are different schemes depending on the accepted trust assumptions: e.g. methods involving return codes and rely on trust in the postal service e.g. as used in Switzerland [1] or methods involving a second hardware device and rely on trust in the independence of this device from the device used to cast the vote, e.g. as used in Estonia [32]. The proposed and applied cast-as-intended verification methods require additional involvement from the voter. This involvement comes with usability challenges addressed in various user centered research investigations and user studies – see e.g. [4, 8]. One well studied cast-as-intended verification method is the one based on *return codes*

as it is used by the Swiss Post voting system in Switzerland. User studies and improved voting materials and interfaces were studied e.g. in [17,22]. In particular with the improvements of [17,22], this method is – both from a security (wrt. to integrity) and a usability point of view – very promising. From a usability point of view the most challenging and error prone task is that voters need to manually enter the *initialisation code* and the *confirmation code* provided on the voting material (the *personal paper code sheet*) they receive via postal service.

This method (like other cast-as-intended verification methods) has one major drawback: untrustworthy voting clients can violate vote secrecy. To address vote secrecy although the voting client is not trustworthy, the *code voting* approach has initially been coined by David Chaum in 2001 [9] and was re-addressed in [28]. The idea of code voting is that voters enter a secret, individual code that corresponds to their desired option on their personal paper code sheet. However, combining both approaches, i.e. the return code approach as discussed in [17,22] with code voting seems not worth considering as the number of error-prone voter tasks increases. Thus, researchers may not consider developing a corresponding voting protocol as it is likely to not be usable.

But what if – and this is our main idea – the codes are included in QR-Codes. Thereby, the task of entering codes becomes less error-prone and the number of interactions can be reduced as the QR-Code can contain more than one of the originally needed codes. We show how voting material could look like for such voting schemes and that it enables usable vote casting and usable verifying. Thereby, we enable new (more robust) types of usable verifiable code voting schemes and invite the community to propose corresponding secure voting protocols using the possibility to provide cryptographic data, even e.g. zero knowledge proofs, in the QR-Codes. Thus, our contributions are as follows:

- An extension of the code sheet and voting webpage of [17] to enable QR-Code based voting codes and confirmation codes.
- A user study to evaluate the *usability* of our code sheet and voting webpage.
- A discussion of opportunities for more *secure* voting protocols with our idea.

The approach that we take differs from how usability concerns are commonly incorporated into the design of secure voting systems: So far, many voting protocols were proposed and then invite the community to design voting material and election webpages being as usable as possible with the given protocol. With this paper, we provide usable front-ends for a verifiable secrecy-preserving e-voting system and invite the community to propose adequate protocols.

2 Background and Related Work

In this section, we first explain the concepts necessary to understand the verifiable code voting ceremony considered in this paper. Afterwards, we discuss related work, i.e. research on the usability of verifiable voting as well as on code voting.

2.1 Background

Verifiable Voting in General. Verifiability in the context of e-voting is no issue, as long as voter privacy is not a requirement. Every voter can verify using public data, if their intention is reflected in the final tally. This can be as simple as checking if the voter's name is alongside the clear-text vote and the sum of all votes match the tally. If, however, voter privacy is a requirement, a more complex process has to be established in order to provide vote secrecy and verifiability at the same time. In this case, verifiability splits in two parts, i.e. individual verifiability and universal verifiability. Universal verifiability can be delegated to any public entity and provides strong proof whether the final tally has been correctly derived from the recorded votes. Individual verifiability on the other hand can only be conducted by the individual voters themselves as only they know their true intention. Individual verifiability provides a strong proof to the verifying voter whether their intention is contained in the cast ballot (*cast-as-intended*) and if the recorded ballot corresponds to the cast ballot (*recorded-as-cast*). This way, the verification is complete, from the voter's intention to the final tally, and hence is called end-to-end verification.

Security Model. For an end-to-end verifiable e-voting system offering vote secrecy, its soundness is based on trust and computational intractability assumptions. The soundness of the end-to-end verifiable voting scheme is strengthened if it operates within minimal trust assumptions. This includes that no unbeknownst manipulation is possible while a minimal defined subset of entities is working honestly, whereas entities can be devices and people. The scheme should also not provide any single entity the capability to break secrecy. However, many proposals rely on the trustworthiness of the voting device when it comes to vote secrecy.

Verifiable Voting using Return/Confirmation/Finalisation Codes. There are various approaches with different trust assumptions. Our paper focuses on the approach used in Switzerland and as required in [1]: Voters receive their individual code sheet via postal service, containing one initialisation code, return codes for each voting option, one confirmation code, and one finalisation code. Voters enter their initialisation code on the election webpage and then select their voting option using the election webpage. Afterwards, they receive a return code which voters are supposed to compare with the one next to their voting option on the code sheet. If the return code is correct, the voter confirms its correctness by entering the confirmation code. If the return code is incorrect, the voter is supposed to vote via an alternative voting channel (postal or in person). Finally, voters receive a finalisation code which should match the one on their code sheet, as an assurance that their ballot has been recorded. The return code would be enough to verify, however, from an organisational perspective it is recommended to have the additional two steps and codes respectively in order to have a chance to react to complaining voters. According to the requirements in [1] and the voting material in [17], the initialisation and the confirmation code are very long (more than 20 digits). The finalisation code should consist of 8 digits (0-9) and the return code of 4 digits.

Usability Considerations. In terms of the ability to cast a vote the most error-prone task is to enter the initialisation code and then the confirmation code.

Security Model. According to the general idea of verifiable voting, no single entity should be able to break vote integrity. In particular, the voting client is considered untrustworthy. Hence, the voting scheme must ensure detection of any malicious vote manipulation. However, it is worth mentioning that a malicious online collusion of the voting client with the printer used to print the voting material or the postal service could break vote integrity. Hence, it is usually recommended to operate the printer offline. This way, the trust assumption remains that the postal service is working highly distributed and thus is more difficult to abuse for large scale attacks.

Code Voting. Code voting has one goal: The human at the far end of the e-voting system shall be enabled to provide a vote in privacy even though its voting client is controlled by malware targeting vote secrecy. The general idea from the voter's perspective is as follows: The voter is provided with a unique code sheet via a trusted secure channel and is supposed to enter the voting code representing their chosen option via the insecure voting client.

Usability Considerations. Very short voting codes may be usable. However, they are problematic from a security and operational perspective. Namely, the minimum length of the unique voting code grows fast as it depends on the size of the electorate times the number of voting options. As such, in many elections with larger electorate or with larger number of voting options, four digits are insufficient and longer codes are required in order to discriminate each eligible vote cast. This problem usually is addressed by more complex system settings, multiplexing different code semantics, e.g. voting card identifier and voting code. From a usability perspective this results in either entering a longer code per voting option or in entering multiple codes (e.g. voting-card number and voting option).

Security Model. Only a minimum subset of entities should be required to work truly honest in order to guarantee vote secrecy. In particular, the voting client is assumed not being trustworthy in any aspect. Again, the voting scheme must ensure vote privacy at this point. This is where Code Voting is at its best. However, as already mentioned in section 2.1 the printer used to print the voting material or the postal service could break vote secrecy together with the voting client.⁴ Furthermore, a malicious printer could manipulate the QR codes leading to a variety of attacks, such as leading the voter to a malicious website. Our security model therefore assumes that the printed material is trustworthy, which can be ensured e.g. via proper audits before the materials are distributed to the voters.

Extending code voting. Note that there exist code voting proposals such as [9, 13] which provide somewhat verifiability. Somewhat, because they only address the (potentially) untrustworthy voting client while the overall soundness

⁴ Even if a malicious printer operates offline, it can insert subliminal messages for malicious voting clients.

relies on too many trust assumptions at the server-side. The proposal by Rui et. al [14] is end-to-end verifiable and uses code voting but the election server must be ultimately trusted for vote secrecy. This trust setting is not acceptable within our security model. The proposal by Neumann et. al [24] lacks robustness as the authorities are oblivious to any malicious voter attacking the system during election phase. By sending different codes to the individual authorities, this rather simple attack results in inconsistent counting results.

2.2 Related Work

In this subsection, we review related research, i.e. user studies on verifiable electronic voting systems: researchers have explored various human factor related dimensions of verifiable electronic voting systems, including research on voters' mental models regarding verifiability in (electronic) voting and usability of verifiable voting systems.

A number of studies have explored voters' *verification-related mental models* [25, 26, 29] and revealed a number of factors that would potentially prevent voters from verifying, such as a lack of verification-related knowledge, effort required to verify, and verification-related misconceptions. These factors need to be addressed when introducing verifiable electronic voting systems.

Other human factor related electronic voting research focused on *usability of verifiable electronic voting systems*. As such, a number of them evaluated user experience and voter satisfaction in these system, e.g. [10, 11, 16, 18, 19, 27, 33]. While some studies reported high users satisfaction scores related to the voting activity, others uncovered usability issues. In particular, a study into usability issues of the Norwegian Internet voting system [11], which relies on a code-based verification, identified a lack of understandability wrt. the different codes used by the system and needed to cast / verify a vote. Distler *et al.* [10] investigated the user experience of the Selene voting system, which uses code-based verification. They reported participants feeling less secure *after* verifying than before.

Other works measure the effectiveness of verification, e.g. for Prêt à Voter and Scantegrity II in [2, 3], for BingoVote in [6], for StarVote in [4], for EasyVote in [8], for Helios Internet voting system in [2, 15, 20, 31]. Some reported high rates of verification effectiveness [4, 8], others reported several issues [2, 3, 6, 20] including verification misconceptions, which resulted in participants being unable to verify their votes successfully. Note that some of these schemes were developed using a human-centered security approach. Therefore the high effectiveness rate is not too surprising. It shows once more, how important it is to take this approach when developing complex systems such as verifiable electronic voting systems. Several of such works focused on the effectiveness of code-based verification, for Estonian voting system in [12], for the Swiss voting system in [17] and for a mock system with code-based verification in [23]. In particular, the results in [12, 17] show that voters have difficulties with detecting manipulations if the adversary manages to tamper with the flow of the verification process, e.g. by removing the verification code and all mentions of it from the user interface of the voting webpage. Marky et al. [22] furthermore propose an improvement towards the

interface of the Swiss voting system that uses code-based verification, showing high verification effectiveness as the result of the improvement.

Most relevant for our current work is the work by Kulyk et al. [17], which proposed and evaluated modifications towards the Swiss voting system [1], showing that while a high percentage of participants were not able to detect vote manipulations introduced in the study using the original system, manipulation detection was improved in the modified version.

Furthermore, there is work on the usability of *code voting*: Marky et al. [21] investigated the usability of code voting, comparing different code modalities. While the authors in [21] only used QR-Codes to include 8-digit voting codes, they showed that using QR-Codes to enter the voting code is perceived as more usable than manually entering the 8-digit voting code. Thus, their results serve as basis for our research. The usability and acceptance of code voting as well as code-based verification including voting codes was also evaluated in [16]. This study reported that participants were more willing to use a system with the highest security assurance in a real-world election, even if it was less usable, which is likely given the complexity of entering and comparing different codes.

3 Voting Ceremony and Voters' Voting Material

The goal of this section is to introduce and explain our design process as well as our design decisions. Note that for this paper, our focus is on typical Swiss voting events, e.g., popular initiatives, referenda, i.e. one question with four options: yes/no/invalid/abstain. We discuss other election types in the discussion section.

3.1 Combining Code Voting with Verifiable Voting using Return/Confirmation/Finalisation Codes

For this paper, we consider a combination of code voting with individual verifiability proposals using return codes, confirmation codes and finalisation codes as used in Switzerland [1]. For such a protocol the voting ceremony is likely to be as follows: voters receive their individual voting material via postal service. Voters choose their desired option and their corresponding *voting code*. Voters enter this code in the voting equipment as depicted in fig. 1a. The device sends that code to the server side of the voting system. The server side responds to that voting code by sending back an according *return code*, depicted in fig. 1b. Voters can then verify by comparing the return code with the one that is provided for that specific option. In case the codes do not match, voters stop the voting process and report to the election management board (EMB). If, however, the return codes match, voters confirm the correct ballot casting by entering the *confirmation code* (see fig. 1c). The server side then acts again by returning the *finalization code* (see fig. 1d). If the code is wrong, voters report to the EMB. If it is correct, the vote casting process is finished and voters can be ensured that their vote has been cast-as-intended and recorded-as-cast.

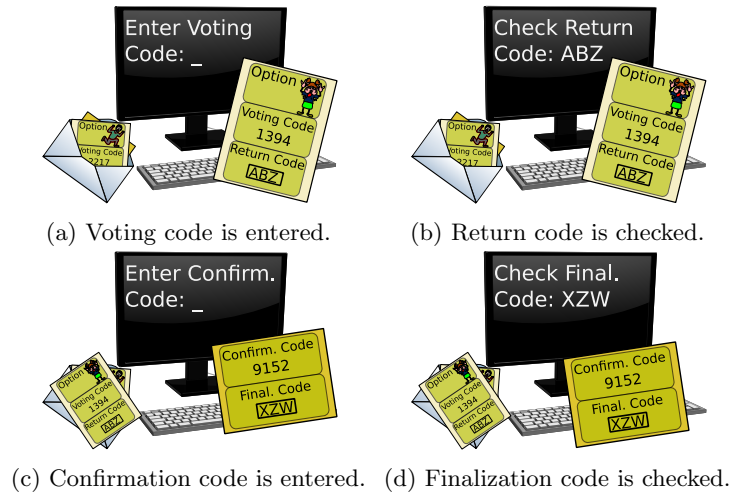


Fig. 1: Abstract Voting Ceremony

3.2 Design Decisions

Usage of QR-Codes and Smartphones. As described in the usability part of section 2.1 the length of the voting codes (or the need to have a voting card identifier plus voting codes) to enter directly depends on the size of the electorate times the number of voting options. To address these shortcomings and to enable more types of election settings for code voting, we propose that voters use their camera-equipped computer device, i.e. most likely smartphones, to cast a vote, as nowadays smartphones are capable to scan QR-Codes. Thus, both the voting codes and the confirmation code are provided as QR-Codes. Note that the initialisation code from the original Swiss approach as such is not needed anymore. The corresponding information is part of the voting code.

Election Webpage instead of App. In order to ensure cross-platform reach, we enable vote casting and verifying through a webpage which is accessible in common mobile browsers, instead of implementing it as an actual app.

Voting material from [17] and not [22]. As described in section 2.2, some research on the usability of voting materials using code-based verification exists already. In particular, the approaches from [17, 22] take the voting material used in Switzerland (for the return code approach) and improve its usability. The proposed improvements are furthermore evaluated with regard to usability as well as the ability of voters to detect election fraud. Both approaches to improve the Swiss voting material are very similar as they change the design to a more step-by-step instruction. Both studied the effectiveness wrt. voter's ability to detect various types of manipulations. Regarding simple manipulations, both improvements enabled all participants to detect them. A more advanced manipulation was only tested in [17]. While not all participants detected this advanced manipulation, the results were promising. Furthermore, the authors

conclude with proposals how to further improve the voting material to increase this detection rate. Therefore, we decided to base the voting material for our research on the proposal of [17] and try to address their proposals on how to further improve it. In particular, their results highlighted the importance of clearly defined voting steps; at the same time, it was shown that the voters primarily pay attention to the voting webpage rather than the voting sheet, which could be a problem if the voting client is compromised. For the latter reason we decided to design the webpage with only the minimal required information and controls; not distracting the voters from the voting sheet which is supposed to be their main source of instructions.

3.3 Security Considerations

In order to adopt the voting material from [17] for our purpose, i.e. for verifiable code voting, we first collected security requirements for the voting material. In particular, as the voting client is assumed to be malicious (see section 2.1), one would need to take into account the different possibilities in which a malicious smartphone could try to record everything with the camera, even without the voter granting permission. Based on that consideration, we came up with the following list of design requirements relevant to security:

- The smartphone, i.e. the camera of the smartphone, should never get knowledge about the voting options and the corresponding return codes [S0].
- The smartphone, i.e. the camera of the smartphone, should only get access to the voting code belonging to the option the voter wants to cast [S1].
- The smartphone should only have access to the confirmation code if the return code displayed on the screen of the smartphone is correct [S2].
- The smartphone should only have access to the finalisation code from the voting material after it is displayed on the screen of the smartphone [S3].

The aforementioned requirements, are addressed by the following proposals:

- There is one voting card per voting option. It shows on one side the option and the corresponding return code; and on the opposite side the corresponding voting code. Thus, in total there are four different voting cards for the election type we consider. This is required to address [S0, S1].
- The instructions in the voting material start with selecting the voting card and returning the voting cards not needed back into the envelope. Note that the election webpage has not yet been contacted nor were voters instructed to use their smartphone. This is required to address [S1].
- The instructions in the voting material afterwards state that voters should place the selected voting card on a specific area in the voting material while having the voting code visible but not the return code. Note that the voters were still *not* instructed to use their smartphone or even open the election webpage. This is required to address [S1].
- The voting material is delivered in form of a leaflet – meaning that in the inner part of the leaflet voters find the first instructions but only until the

- step in which they should check the return code. They are only asked to continue to the next and last page if the return code matches. This is required to address [S2].
- The voting material should hide the finalisation code under a scratch field. This is required to address [S3] as the finalisation code and the confirmation code are on the same side of the page.

3.4 Final Voting Material

The voting material and the election webpage have been developed and iteratively improved through feedback. The final version of the voting material is depicted in fig. 2 (voting cards). [17] also includes a proposal for how to improve the election webpage. However, as we wanted to display it only on a mobile device and most likely a smartphone, we decided to keep it as simple as possible and thus reducing any information on the election webpage to the bare minimum. Another reason for such a minimalistic approach is derived from the findings from previous research [17], namely, the need to ensure that the voters follow the instructions on their trustworthy voting materials and are not distracted by instructions on the untrustworthy voting webpage that might be manipulated by the attacker. The final version of the election webpage interfaces is depicted in fig. 4 and the voting card in fig. 3.

4 User Study

The purpose of the user study is to evaluate our proposed design for the voting materials. We want to answer the following research questions:

Effectiveness: Can voters successfully *cast* and *verify* their vote using the proposed system?

Satisfaction: What is the mean System Usability Scale (SUS) [5] score of the system?

In addition to these questions, we apply the modular User Experience Questionnaire [30] in order to measure the participants' impression of the proposed system wrt. *perceived efficiency*⁵, *perceived perspicuity*, *perceived dependability*, and *trust*. We furthermore aimed to collect qualitative user feedback, in order to identify problems and potential improvements.

4.1 Study procedure

Before the actual study, participants received the voting materials as well as supplementary materials either via postal service or if possible in person from one

⁵ Note that as opposed to efficiency as one of the three standard usability criteria, the UEQ measures the subjective impression of whether the system feels efficient to the user, as opposed to an objective measuring of e.g. time spent on using the system

Individual Voting Instructions

SUPPORT 0800 99 88 66

Before you start: This voting card allows you to participate in the referendum on the following topic:

Do you want to accept the initiative "For responsible businesses – protecting human rights and the environment" ?

Please note the attached informational documents.

To accept the popular initiative, vote **YES**, to reject it, vote **NO**. You are also able to **ABSTAIN** or **INVALIDATE** your vote.

For each of the four options you got a voting card with a voting code.

In the event of problems or irregularities, only call the telephone number provided at the top of the voting entitlement identification.

You are now able to start the voting procedure. To do so, open the inner side of this voting instructions and start with **Step 1. Selection**.

(a) front

5. Confirmation: Now, click "Scan confirmation code" on the election website. Scan the code below. By doing so, this vote is considered as cast.

CONFIRMATION CODE

6. Finalizing: The finalizing code is shown on the election website.

If this is not the case, contact the support immediately!

FINALIZING CODE

4946-0511

To reveal the finalizing code on your voting card, scratch it with a coin or your finger.

Check if the code matches the code on the election website.

If the code does not match, contact the support immediately!

If this is the case, vote cast is completed. You can now close the election website.

CHECK LIST

Vote cast is completed

(c) back

1. Selection: Decide on one of the voting options and place the **corresponding voting card** on the right side of this leaflet. Put the highlighted corner in the top right.

To avoid accidental scanning, place the remaining voting cards back into the envelope.

2. Election website: Open the election website on your smartphone:

<https://wahlwebsite.de>

3. Vote: On the election website, click "Scan voting code". To do so, **grant** the election website **camera access**. Scan the code from the on the right side at the same time as depicted.

4. Check code: The election website now shows a check code.

If no check code is shown, contact the support immediately!

No code

Please check if the check code on the election website matches the code on the backside of your voting card.

Wrong code

If this is not the case, contact the support immediately!

If it is the case, return the voting card in the envelope to the other cards in order to avoid accidental scanning. Afterwards, confirm the match on the election website.

Continue on the next page

(b) inner

SUPPORT 0800 99 88 66

PLACE VOTING CARD HERE

Fig. 2: Voting instructions

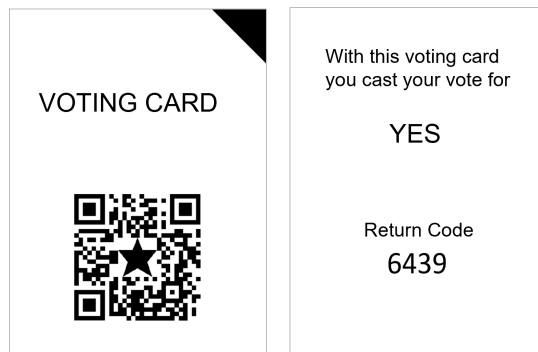


Fig. 3: Voting Card (front and back side)



Fig. 4: Voting webpage

of the authors. The supplementary materials include a description of the purpose of the study and the study procedure as well as a role card. In the role card, participants were told to imagine that they are voters living in Bern who want to cast their vote with regards to a mock popular initiative “For responsible business in protection of people and the environment”, with “yes/no/invalid/abstain” as available voting options. For the sake of the study, the participants were asked in the role card to cast their vote for a “yes” option. The participants were given a time frame of nine days during which they could cast a vote remotely and participate in the user study. For this study, the prototype webpage was developed and hosted on the “2021.wahlwebseite.de” domain and the support phone number on the voting material was the mobile phone number of one of the authors. Once participants cast their vote, the webpage provided a link to the SoSciSurvey survey platform to answer some questions. Note that no data was collected by the webpage. It was only counted whether the support hotline was called and whether it was possible to solve their issue.

On the survey page, participants were presented with a consent form for collecting their data within this survey. Once they agreed, they were asked questions on whether they were able to complete vote casting and whether they contacted the support before or during the process. Afterwards, the participants were presented with questions from the SUS and UEQ questionnaires, asked to elaborate on any problems they had experienced, what they liked and disliked about the voting materials and the election webpage, and whether they had any improvement suggestions. The survey ended with questions on demographic variables (such as age, gender, education), previous experience with Internet voting the participant might have had and information about the device (browser, OS) that the participant used for voting.

4.2 Recruitment and ethics

The participants for the study were recruited via the snowball principle starting with close friends and family. Participants were informed about the purpose of the study (i.e. to evaluate the usability of the voting material and if it enables voters to cast their vote), as well as told that they could withdraw from the study without providing any explanation. The study was designed to be anonymous, only collecting information that the participants chose to input in the survey platform. The participants were not offered any reimbursement.

4.3 Study results

Overall, 40 participants took part in the study (while 50 received the study and voting material), of them 22 women and 18 men. The participants were aged 19-81, with the median age being 44. Most of the participants had some higher education degree (25 out of 40). Only three out of 40 participants had previously cast their vote online. Three reported to have a background in security.

Effectiveness. Overall, 95% of the participants (38 out of 40) reported being able to complete the voting procedure. Two participants did not succeed in

completing the voting procedure even after having called the support. Their device could not read the QR-Code. Note that we send these two the link to the survey via email to collect their feedback. One of the 38 reported contacting the study examiner at some point during the study. The issue with the election webpage could be solved with the help of the support.

Satisfaction. The SUS scores given by the 38 participants ranged from 40 to 100, with a mean score of 84.14, corresponding to the grade “B” (good) according to [5] and a standard deviation of 13.1. The mean value is close to the one reported in [17], namely, 80.9.

UEQ. The participants rated the voting system highly across all the used scales, with mean score of 2.2 ($SD = 0.76$) for dependability, 2.14 ($SD = 0.9$) for efficiency, 2.07 ($SD = 1.18$) for perspicuity and 1.45 ($SD = 1.15$) for trust, all on a scale from -3 to 3.

Feedback. Two of the authors analysed the answer provided regarding aspects participants liked and disliked (including their proposals for improvements), as well as problems they had. While more positive than negative answers were provided, in this paragraph we will focus on the negative ones as they can help to further improve the usability.

Regarding the question on *encountered problems*, 23 of the 38 participants who could cast their vote reported that they had no problem at all. Seven reported that they first had an issue with opening the webpage (e.g. due to a typo in the webpage). Two reported that they had to switch browser / device as the first setting did not enable them to scan the QR-Code. Three participants first put the voting card the opposite way, i.e. they tried to scan the return code together with the QR-Code containing the triangle.

Regarding the *positive aspects* of both the voting material and the webpage, most people mentioned at least two of the following aspects: ‘easy’, ‘fast’, ‘well explained’, ‘clear’, ‘well structured’, ‘good usability’, and ‘easy language’. Often, participants were – in particular – referring in their answers to the step-by-step instruction.

Regarding the *negative aspects of the voting material*, 13 of the 38 mentioned that there is nothing they did not like. At least three times the following types of input were provided: eleven either asked specifically for further information (five particularly asked for security related information) or – from their answers – we deduced that they had misconceptions (e.g. one participant proposed to put the actual option e.g. yes on the same page of the voting card as the QR-Code is) which could be addressed by providing more information. Eight made a concrete proposal how to rephrase or extend sentences. Furthermore, three thought that the scheme is not very environmentally friendly and three, again, mentioned that typing in the URL is error prone. Regarding the *negative aspects of the webpage*, 23 of the 38 participants did not mention anything, three only mentioned again the issues with entering the URL, and two added a remark that the webpage would look more official if it would be an actual election (e.g. it would have an imprint). There was only one aspect which was mentioned at least three times: the request for more information either through a video and/or by providing

more information regarding the status in the process on the webpage. The request for more information about the security and the reason for each of the steps was also mentioned by four of the seven participants who provided input in the final-remark-question.

4.4 Result discussion and limitations

Our user study shows that it is possible to design verifiable code voting systems that achieve a high level of usability – with our proposals scoring high both in terms of effectiveness (with almost all of our study participants being able to successfully complete the voting procedure) as well as in terms of satisfaction and user experience. It is worth mentioning, that the problems with the smartphone and the camera would not have happened in a typical lab user study (as conducted in [17,22]) in which participants would have used lab equipment. Due to the remote character of our user study, we did not know which smartphone participants used and were limited wrt. testing various combinations of smartphones-OS versions - webbrowsers. Before using such an approach in the field, the webpage would need to be tested with more potential combinations, to further reduce such issues.

Beside the positive usability and user experience, our results show potential for further improvements and directions for future research: e.g. the URL for the election webpage should also be provided as QR-Code to avoid errors when typing the URL on the smartphone and it should be better explained in which way the voting card should be placed before scanning it. As it is critical that the camera of the voting device used to scan the QR codes does not capture any information that is supposed to be hidden (most notably, the correspondences between the voting cards and the voting options on them), additional studies need to be done to make sure that the voters are aware of this aspect, and that their interactions with the system do not lead to errors that might violate their vote secrecy.

We also received comments unrelated to the actual usability but which are worth to be considered as future work: Our participants wished for more transparency regarding the scheme, including information on the security of the system and explanations on why the individual steps are needed. While providing such information was out of scope for our study, developing ways to communicate it would be an important research direction for real-world elections. Such information could also explain why it is needed to have paper-based materials sent to voters and why the actual option cannot be printed on top of the QR-Code. Furthermore, as mentioned by participants, our proposal relies on availability of smartphones with cameras capable of scanning QR-Codes; while such assumption might be trivially fulfilled in Switzerland (with recent surveys showing more than 97.2% of the population in possession of a smartphone)⁶, consideration of other alternatives might be useful for the applicability of code voting in other settings. A related issue is the accessibility of the system; in particular, both the original

⁶ <https://de.statista.com/statistik/daten/studie/537944/umfrage/besitz-von-smartphone-bzw-tablet-in-der-schweiz/>

Swiss system as well as our improvement could require additional assistance to be used by voters with vision impairments. Alternatives to the use of QR codes, including more accessible ways to represent the codes (such as codes that are designed to incorporate error correction [7]), need to be further investigated.

Limitations. The focus of our user study was on usability and user experience aspects. We did so by sending participants their voting material home. By doing so, we increased external validity compared to previous research being conducted in the lab. However, casting the vote from home is a less controlled environment. Thus, we cannot know whether participants asked others in the household for help. We also don't know how carefully they read the materials before and while casting their vote. Different to [17, 22], we leave an evaluation of the effectiveness of the cast-as-intended verification for future work. As a consequence, we could show good usability wrt. vote casting but we do not know whether voters would be able to detect manipulations.

5 Conclusion

We proposed voting material and election webpage to enable verifiable secrecy-preserving e-voting schemes based on machine readable data aka. QR-Codes. This idea allowed us to remove the step in which the initialization code is entered by voters. We evaluated our proposal in a user study. The results of the user study show that users are able to actually use code voting in combination with the return code approach from [1]. Thus, the introduction of QR-Codes on the code sheet sent via postal service is a game changer: instead of hushing away from the voters completely non-trustworthy camera-equipped, connected computing device, we now can endorse or even mandate its usage in order to handle big data chunks correctly, without lowering usability. From a cryptographical perspective, the ability to confidentially handle big chunks of data at the voter's side has an immediate effect on the protocol design. This change enables the system to provide the voter with cryptographically sound data such as digitally signed encrypted data or zero knowledge proofs. This in turn allows to offload some of the strong trust-assumptions at the server side and paves the way for much more robust verifiable e-voting schemes using code voting. Now, it is up to the community to propose corresponding schemes knowing that it is possible to design usable voting materials and election webpages.

References

1. Verordnung der Bundeskanzlei über die elektronische Stimmabgabe (VEleS) (July 1st 2018). Die Schweizerische Bundeskanzlei (2018)
2. Acemyan, C.Z., Kortum, P., Byrne, M.D., Wallach, D.S.: Usability of voter verifiable, end-to-end voting systems: Baseline data for Helios, Prêt à Voter, and Scantegrity II. *The USENIX Journal of Election Technology and Systems* **2**(3), 26–56 (2014)
3. Acemyan, C.Z., Kortum, P., Byrne, M.D., Wallach, D.S.: From error to error: Why voters could not cast a ballot and verify their vote with Helios, Prêt à Voter, and

- Scantegrity II. *USENIX Journal of Election Technology and Systems* **3**(2), 1–19 (2015)
4. Acemyan, C.Z., Kortum, P., Byrne, M.D., Wallach, D.S.: Summative Usability Assessments of STAR-Vote: A Cryptographically Secure e2e Voting System That Has Been Empirically Proven to Be Easy to Use. *Human Factors* pp. 1–24 (2018)
 5. Bangor, A., Kortum, P., Miller, J.: Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of Usability Studies* **4**(3), 114–123 (2009)
 6. Bär, M., Henrich, C., Müller-Quade, J., Röhrich, S., Stüber, C.: Real world experiences with bingo voting and a comparison of usability. In: *IAVoSS Workshop On Trustworthy Elections (WOTE)* (2008)
 7. Blanchard, N.K., Gabasova, L., Selker, T.: Consonant-vowel-consonants for error-free code entry. In: *International Conference on Human-Computer Interaction*. pp. 19–37. Springer (2019)
 8. Budurushi, J., Renaud, K., Volkamer, M., Woide, M.: An investigation into the usability of electronic voting systems for complex elections. *Annals of Telecommunications* **71**(7-8), 309–322 (2016)
 9. Chaum, D.: Surevote: technical overview. In: *Proceedings of the workshop on trustworthy elections (WOTE'01)* (2001)
 10. Distler, V., Zollinger, M.L., Lallemand, C., Roenne, P., Ryan, P., Koenig, V.: Security–visible, yet unseen? how displaying security mechanisms impacts user experience and perceived security. In: *ACM CHI*. pp. 605:1–605:13 (2019)
 11. Fuglerud, K.S., Røssvoll, T.H.: An evaluation of web-based voting usability and accessibility. *Universal Access in the Information Society* **11**(4), 359–373 (2012)
 12. Gjøsteen, K., Lund, A.S.: An experiment on the security of the norwegian electronic voting protocol. *Annals of Telecommunications* **71**(7-8), 299–307 (2016)
 13. Helbach, J., Schwenk, J.: Secure internet voting with code sheets. In: *E-Voting and Identity*. pp. 166–177. Springer (2007)
 14. Joaquim, R., Ribeiro, C., Ferreira, P.: Veryvote: A voter verifiable code voting system. In: *E-Voting and Identity*. pp. 106–121. Springer (2009)
 15. Karayumak, F., Olembo, M.M., Kauer, M., Volkamer, M.: Usability Analysis of Helios-An Open Source Verifiable Remote Electronic Voting System. In: *Proceedings of the 2011 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections. EVT/WOTE'11, USENIX Association* (2011)
 16. Kulyk, O., Neumann, S., Budurushi, J., Volkamer, M.: Nothing comes for free: How much usability can you sacrifice for security? *IEEE Security & Privacy* **15**(3), 24–29 (2017)
 17. Kulyk, O., Volkamer, M., Müller, M., Renaud, K.: Towards improving the efficacy of code-based verification in internet voting. In: *FC: VOTING Workshop*. pp. 291–309. Springer (2020)
 18. MacNamara, D., Gibson, P., Oakley, K.: A preliminary study on a DualVote and Prêt à Voter hybrid system. In: *CeDEM*. p. 77 (2012)
 19. MacNamara, D., Scully, T., Gibson, P.: Dualvote addressing usability and verifiability issues in electronic voting systems (2011)
 20. Marky, K., Kulyk, O., Renaud, K., Volkamer, M.: What Did I Really Vote For? In: *ACM CHI*. p. 176 (2018)
 21. Marky, K., Schmitz, M., Lange, F., Mühlhäuser, M.: Usability of Code Voting Modalities. In: *ACM CHI* (2019)
 22. Marky, K., Zimmermann, V., Funk, M., Daubert, J., Bleck, K., Mühlhäuser, M.: Improving the Usability and UX of the Swiss Internet Voting Interface. In: *ACM CHI* (2020)

23. MARKY, K., ZOLLINGER, M.L., ROENNE, P., RYAN, P.Y., GRUBE, T., KUNZE, K.: Investigating usability and user experience of individually verifiable internet voting schemes. *ACM Trans. Comput.-Hum. Interact* **28**(5) (2021)
24. Neumann, S., Feier, C., Sahin, P., Fach, S.: Pretty understandable democracy 2.0 (2014), <https://eprint.iacr.org/2014/625.pdf>, [Online, May 14th 2021]
25. Olembo, M.M., Renaud, K., Bartsch, S., Volkamer, M.: Voter, what message will motivate you to verify your vote. In: USEC. Internet Society (2014)
26. Olembo, M.M., Bartsch, S., Volkamer, M.: Mental Models of Verifiability in Voting. In: *E-Voting and Identity*. pp. 142–155. Springer (2013)
27. Oostveen, A.M., Van den Besselaar, P.: Users' experiences with e-voting: A comparative case study. *Journal of Electronic Governance* **2**(4) (2009)
28. Oppliger, R.: How to address the secure platform problem for remote internet voting (2002), http://pubs.esecurity.ch/sis_2002.pdf, [Online, May 14th 2021]
29. Schneider, S., Llewellyn, M., Culnane, C., Heather, J., Srinivasan, S., Xia, Z.: Focus group views on prêt à voter 1.0. In: REVOTE, pp. 56–65. IEEE (2011)
30. Schrepp, M., Thomaschewski, J.: Design and validation of a framework for the creation of user experience questionnaires. *International Journal of Interactive Multimedia & Artificial Intelligence* **5**(7) (2019)
31. Weber, J., Hengartner, U.: Usability study of the open audit voting system Helios (2009)
32. Wikipedia: Electronic voting in estonia, https://en.wikipedia.org/w/?title=Electronic_voting_in_Estonia&oldid=1012067015, [Online, May 14th 2021]
33. Winckler, M., Bernhaupt, R., Palanque, P., Lundin, D., Leach, K., Ryan, P., Alberdi, E., Strigini, L.: Assessing the Usability of Open Verifiable E-Voting Systems: a Trial with the System Prêt à Voter. In: ICE-GOV. pp. 281–296 (2009)

“Just for the sake of transparency”: Exploring Voter Mental Models of Verifiability

Marie-Laure Zollinger¹, Ehsan Estaji¹, Peter Y.A. Ryan¹, and Karola Marky²

¹ University of Luxembourg, Esch-sur-Alzette, Luxembourg
 {marie-laure.zollinger,ehsan.estaji,peter.ryan}@uni.lu

² University of Glasgow, Glasgow, UK
 karola.marky@glasgow.ac.uk

Abstract. Verifiable voting schemes allow voters to verify their individual votes and the election outcome. The voting protocol Selene offers verification of plaintext votes while preserving privacy. Misconceptions of verification mechanisms might result in voters mistrust of the system or abstaining from using it. In this paper, we interviewed 24 participants and invited them to illustrate their mental models of Selene. The drawings demonstrated different levels of sophistication and four mental models: 1) technology understanding, 2) meaning of the verification phase, 3) security concerns, and 4) unnecessary steps. We highlight the misconceptions expressed regarding Internet voting technologies and the system design. Based on our findings, we conclude with recommendations for future implementations of Selene as well as for the design of Internet voting systems in general.

1 Introduction

Elections are the foundations of democracy. To improve access to elections, several countries introduced ways to conduct elections over the Internet (e.g., Estonia [12], or Switzerland [30]). To uphold democratic principles, voting researchers have proposed secure and robust systems ensuring the integrity of Internet elections. The goal is to satisfy two main security features among others: privacy and verifiability. Privacy, in particular vote-secrecy, is well-known as it is also mandated by the law in many countries. Verifiability comprises *individual verification* meaning that each voter can check that their vote has been correctly recorded, and *universal verification* meaning that the outcome of the election can be confirmed by any observer [4]. Verification mechanisms seek to provide assurance of the correct execution of an election and hence in the outcome.

Verifiability must provide convincing proof to any voter that their votes are correctly cast-as-intended, recorded-as-cast, and counted-as-recorded [4]. To achieve this, Internet voting schemes rely on cryptography, often at the expense of usability (cf. [2, 19, 20]). Research on voting has shown that voters are concerned by risks related to security [31, 33] affecting their trust, especially as voters can consider verifiability mechanisms as privacy breaches [24, 28], or question their necessity [2, 19]. This might be due to the novelty of verification, which has been

used in only a few real elections with high stakes, e.g., [12, 30]. It might also be due to the complexity of the verification, requiring the voters to perform extra steps, understand complex mechanisms, or compare cryptographic data [5, 7, 8].

To counter this, the e-voting scheme Selene has been developed to minimize the voters' interaction with cryptography while providing individual and universal verifiability [26]. Selene's usability has already been demonstrated in studies with voters [11]. However, usability studies of Internet voting protocols have shown that mere usability is not sufficient in convincing voters about the correct processing of votes [2, 13, 19]. This might be because the voters' mental models do not align with the verification procedure.

Mental models are the internal representations that humans derive from interacting with a technology [25]. Mental models using the Selene protocol have been evaluated in a previous study [37]. In this paper, we investigate an improved implementation of the Selene protocol that builds on previous results. We evaluate voters' perceptions of the Selene e-voting protocol with 24 participants. To achieve that, after letting them interact with the app, we asked the participants to draw their understanding of voting and verifying using Selene.

Our contributions. We explore the voters' understanding of the verification mechanism in the Selene Internet voting protocol. For that, we performed an analysis of the drawings and the answers and extracted four categories of mental models: 1) technology understanding, 2) meaning of the verification phase, 3) security concerns, and 4) unnecessary steps. We also classified the understanding of participants into levels of sophistication of their mental models. Finally, we discuss our findings and propose a list of recommendations applicable to Selene and to other Internet voting systems, focused on 1) education of voters on risks, 2) need for correctness and transparency, 3) integration of simple interactions with security features, and 4) design of several levels of verification.

Related work. *Mental models* are internal representations that humans derive from the real world to interact with technology [14, 25]. The level of sophistication of a mental model can differ amongst humans [9, 14, 15] and the mental models must be sound enough that users can effectively interact with a technology [16]. Generally, two types of mental models can be observed: functional and structural models [25]. Functional models mean that users know how to use a technology, but they do not how it works in detail. Structural models offer a more detailed understanding of how technology works. Once a mental model has been established, it is difficult to shift [32]. There are different ways to capture mental models, such as interviews (cf. [34]), sketching, or think-aloud techniques [15]. Related work in the domain of privacy has demonstrated that the combination of sketching and think-aloud is effective to capture the mental models [35].

Mental models have been investigated within the scope of security and privacy [1, 3, 15, 36] indicating that misconceptions within mental models can lead users to engage in insecure behaviours, or in behaviours that do not match their intentions. Mental models within the scope of verifiable voting have also

been investigated. Acemyan et al. [3] let voters draw their mental models after interacting with the three electronic voting schemes Helios, Prêt à Voter, and Scantegrity II. This study reveals that mental models are almost exclusively based on the voting process from their perspectives in all three protocols. Thus, voters expressed rather functional mental models that did not describe how the voting schemes worked. 75% of participants expressed to have recognized that their votes have been encrypted when using the Helios protocol. The usability of Helios [5] has been studied in many papers, such as [2,3,19]. Later investigations of Helios confirmed that the probabilistic nature does not align with voters’ mental models, and because of that, voters considered verification to be unnecessary [19]. In a previous study of the Selene protocol, Zollinger *et al.* investigated mental models of the participants regarding technical properties that are required for security [37]. Their results show that voters are aware of potential security issues in Internet voting, but the presented verification mechanism did not convince them to mitigate these security issues. Our study takes into account this previous result.

Another line of research investigated perceptions of vote verification. As part of the trials to deploy online voting in Norway, participants failed to determine whether their votes had been submitted, although the scheme offers verification [13]. Using Helios, between 10 and 43% of participants were able to verify successfully [2,19]. Information provided to voters is crucial for the acceptance of verification [20].

In summary, research has shown that mental models have to be sound enough such that users can effectively interact with a technology. If voters have misconceptions, they might be unsuccessful in verifying their votes, consider it redundant, or question the security of the voting scheme. Adding to this body of research, we report a detailed investigation of mental models regarding the Selene Internet voting protocol.

2 The Selene Internet Voting Protocol

The app used for the user study is an implementation of the Internet voting protocol Selene [26]. Selene allows voters to identify their plaintext vote in the tally using a tracking number (or tracker) which is revealed to the voters *after* the election’s outcome has been published. This is to provide coercion mitigation: letting voters identify another tracker to show to a coercer. However, this feature is not in the scope of the paper. Showing the voter the plaintext vote in the final tally should be more understandable than more conventional verifiable schemes that require the voter to check an encryption of the vote in the input to the tally.

2.1 Voter Experience and Protocol Setup

In this section, we summarize Selene’s cryptographic setup³.

³ A full cryptographic description of the protocol can be found in [26].

Preliminaries: Each voter has a public/private key pair for use in the verification phase. The keys are generated and handled by the app; the voters do not have to interact with them. An election public key is generated with a corresponding private key. The public key is included in the app to avoid direct interaction with the voters.

Setup (Authorities): First, the election authorities generate a list of unique trackers. These trackers are encrypted with the public election key, secretly shuffled, and each of them is associated with a voter. A commitment to each tracker is created, sealing the relation between a tracker and a voter without revealing it. Each commitment can be opened only by its associated voter, using the voter's private key and a secret term delivered after the tally has been published.

Voting (Voters): To cast their votes voters log in to the voting app with credentials that they received before the election. After a welcome page, they select a candidate. Then, the app computes an encryption of their vote under the election public key and sends it to the election authority. The latter stores the encrypted votes next to the encrypted tracking number.

Tally (Authorities): When voting is over, the authorities extract the pairs of encrypted trackers and votes, which they shuffle and decrypt to obtain the pairs of plaintext trackers and votes which are then posted to the bulletin board.

Verifying (Voters): After the election, the secret value associated with the commitment is sent to the respective voter. The app combines the secret and the commitment and uses the voter's private key to reveal the tracker, without revealing the value to anyone else. We also highlighted one positive aspect regarding verification: the correctness of the records can be verified by anyone.

2.2 App Design

To increase security, the interfaces are split into two apps: one for voting and one for verifying. In case the voting app is compromised, it should not impact verifiability. This should also indicate to voters that their vote is not recorded by the voting app: when they check their vote, they retrieve a tracker and verify the associated vote.

Within the interfaces, we do not communicate all the information regarding the protocol. Instead, we stick to the interactions the voters perform: voting and verifying. In particular, the setup phase was not communicated in advance, and the tally is computed between the voting and the verification phases. Also, most security interactions that voters have with the protocol are related to encryption/decryption. In our app, the voter must explicitly push a button with the label "Encrypt", while the trackers are automatically decrypted. The information is shown to the user through a loading screen.

Finally, the possibility to choose another tracker in case of coercion is not provided in this version of the app since coercion mitigation was out of this investigation's scope.

3 Method

To evaluate the users’ perceptions and understanding of the Selene Internet voting protocol, we conducted a user study with 24 participants.

Selene has been partially implemented as a demonstrator in the UK with a commercial partner [27]. For our study, we developed an interface where the voters can perform the required tasks: voting and verifying their vote. The interface design was informed by guidelines for Internet voting interface from the literature [20]. We also implemented a backend server where the authorities can set up elections, store votes, and compute the tally pairs (tracker, vote). The apps simulated an election for a past parliament for Germany to give a realistic scenario as recommended in [21, 29]. Therefore, we used the ballots and results from the last election in the constituency where the study took place. The election had two contests, the first one had six candidates and the second one 20.

3.1 Procedure

Before interviewing the participants, they interacted with the Internet voting scheme. With this, we wanted to know whether the participants were able to verify their votes successfully. To capture this, we randomly manipulated one of the two contests for all participants. This means that the voting option next to the tracker did not correspond to the voter’s choice. The procedure of our study was as follows.

We welcomed the participants by explaining that we are investigating an online voting protocol and that they are going to vote in an Internet (dummy) election followed by an interview. Then, we let them read the consent form and the study’s data protection policy. Each participant provided demographics consisting of age, gender, education, and occupation. We also asked about previous voting experiences. The participants were introduced to the voting materials and devices consisting of a letter with sealed voting credentials (voter ID and password) as in a real election. Each participant received randomly chosen voting instructions, i.e. voting option for each ballot. This was to preserve the participants’ vote privacy since we took screen-recordings [21]. Note that we explored additional user experience and usability aspects related to tracker-based protocols, that we elaborate in [22].

Each participant cast two votes since we wanted them to experience the voting scheme with and without a manipulated vote. In each round, the participants were asked to cast a vote matching the instructions. In the second round, we randomly manipulated one vote of a contest. When the participants reported completion, the examiner gave the following scenario: two weeks have passed⁴ since the voting phase and the election results are now available. The participants were asked to use the verification app⁵.

⁴ In Germany, where the study was conducted, this is the standard time frame between the end of the voting phase and the announcement of the outcome.

⁵ The emphasis was placed on the individual check of the tracking number. We did not explicitly ask the participants to recount the votes for universal verifiability.

After the interaction with Selene was completed, we proceeded with the interview part. We explained that we would like them to draw their understanding of the following questions and that there are no wrong answers. The drawing area was recorded with a camera, and the participant's comments were audio-taped. We told the participants when we started the recording and proceeded with the semi-structured interview which was guided by the following main questions:

- Could you sketch how vote casting works according to your understanding?
- What to your understanding is the purpose of the tracking number?
- Why to your understanding is it necessary to see the list of all votes and not only your own one or is it not necessary at all?
- How to your understanding does the vote verification work?
- Why do you think voters are asked to verify?
- Consider an election, would you want information on how the vote verification works? Where or when would you like to receive this information?

In each question, the participant could integrate cards with pictures that we provided into their drawings. The cards had pictures of the following components: an icon representing the voter, a ballot, a ballot box, a smartphone, the icon of the app, an icon representing the Internet, an icon for encryption and a server. We provided the items to facilitate the drawing for the participants.

Participant were invited to ask questions, or to provide further feedback. We did not compensate individual participants but they could participate in a raffle for a voucher for online shopping in the value of about 100 Dollars.

3.2 Participants and Ethical Considerations

We recruited the 24 participants by mailing lists, social networks, and poster advertisements that did not mention verifiability. Fifteen participants identified as male, eight as female and one as other. The average age was 24.8 years (Min=19, Max=40, SD=5.37). All participants reported daily Internet usage. The study followed the guidelines provided by the ethics commissions at the authors' institution and conforms to strict national law. In particular, our studies must limit the collection of personal data to preserve the privacy of participants. To anonymise the data, each participant received a randomly assigned identifier. Before the study, each participant signed a consent form that was recorded separately such that data cannot be linked to participants' identity. The following information were provided to the participants: goal and procedure of the study, risks associated with the participation, and how data storage and analysis is handled. Finally, it has a paragraph regarding data protection policy. The study was conducted before COVID-19.

3.3 Data Analysis

We transcribed the interviews and used a deductive coding methodology to categorize the data. The categories were discussed before starting the coding and

emerged from the questions given to participants and the existing literature on the analysis of voters’ perception. Then, two researchers coded the interviews independently. The agreement is given by Cohen’s Kappa was calculated at 0.822, referring to an almost perfect agreement [10]. Then, the coders compared their findings and resolved disagreements. The drawings were categorized by two researchers, ordering them according accuracy and then relating them to the participants feedback about their experience.

3.4 Limitations

Although we took precautions and recruited participants beyond the university campus, some of our participants had background in computer science or were students. Consequently, our sample might not be representative. Our aim was to provide a explorative stepping stone for further investigations.

Furthermore, if technology-savvy voters already demonstrate understandability issues, those are likely to be exacerbated in a more general sample.

Another limitation is that the study was run in a lab hence a controlled environment [17, 18], potentially leading to biased answers from the participants. However, for the voting area, it is hard to conduct experiments over real elections while preserving voters’ privacy [21].

One feature of Selene, the coercion mitigation mechanism, was not in the scope of this study. Hence, we cannot draw conclusions about the voters’ mental models regarding this feature. Finally, the results and conclusions are applicable to countries with similar cultures (Germany).

4 Results

In this section, we present the results of our study. Previous studies that investigated Selene demonstrated a good user experience but some misconceptions remain [11, 37]. In this paper, we want to go further by asking the participants explicitly how they understand Selene and represent it in drawings to reveal their understanding of the verification mechanisms and their beliefs regarding Internet voting technologies. In the remainder of this section, we first present *levels of sophistication* before detailing the observed *mental models explicitly*.

4.1 Levels of Sophistication

Many participants had a good overview and provided a good explanation of how the system works according to their understanding. We classified the drawings in two types of mental models as described in [25]: 1) functional models and 2) structural models. The functional model describes the drawings in which the participants used the provided components without linking those components together. The structural model describes the drawings depending on the use of components and their relations to each other.

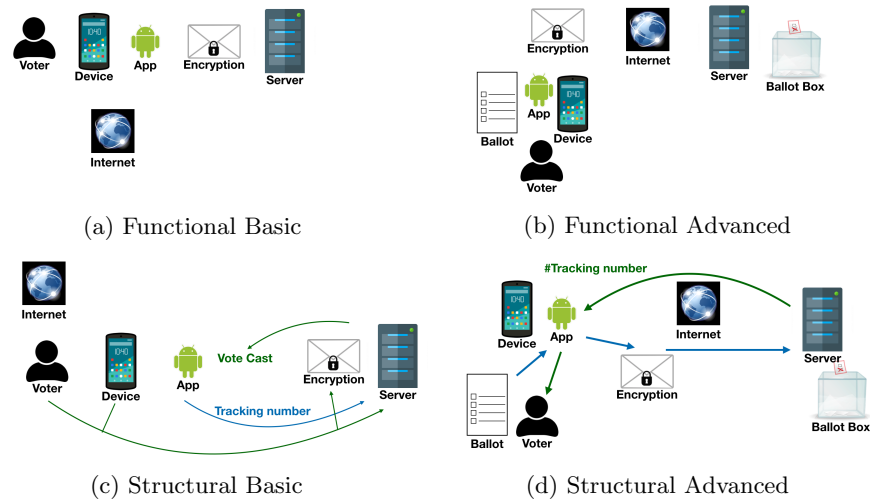


Fig. 1: Four levels of understanding.

We can deduce two levels of understanding inside those two main categories: basic and advanced. Figure 1 shows an example of drawing (reproduced) for every category mentioned below. We describe the four levels as follows:

1. **Functional basic** (one participant): some components are used but are not detailing the entire procedure. In Figure 1a, P22 used some components, but the ballot and the ballot box are missing.
2. **Functional advanced** (seven participants): the components are used in a specific order to express the functional tasks that were performed. In Figure 1b, P21 used all components and grouped them while explaining his experience.
3. **Structural basic** (nine participants): some components are used and related to each other. In Figure 1c, P14 used some components and tried to relate them but misplaced or did not use all of them.
4. **Structural advanced** (seven participants): the components and their relations were correctly set. In Figure 1d, P01 used all components and explained the correct structure and relations between all of them.

Regarding the vote manipulation, we counted 20 participants who clearly reported that they have seen a problem to the examiner.

Besides the general level of sophistication, the comments and drawings from the participants can be grouped into: 1) *technology understanding*, 2) *meaning of the verification phase*, 3) *security concerns* and 4) *unnecessary steps* which we detail in the following sections.

4.2 Technology Understanding

All participants gave their vision of how the voting system is designed and how they understood the technology behind it. As described in Section 3, we provided several components to participants. Some of them were related to a standard paper-based system (ballot, ballot box), and others were related to online technologies (Internet, app, device, encryption, server).

Overall Voting Technology. From the 24 participants, seven did not use the paper-based components in their drawings. All other components were used but sometimes misplaced. Six participants thought that encryption occurs on the server side, although the app mentioned that it is done locally. Three participants placed the ballot box in the smartphone instead of the server. For instance, participant P16 said: *“The smartphone is the ballot box and the app is a tool.”*

Nineteen participants provided a good description of their experience and the technology in use. For example: *“The smartphone uses the app to apparently retrieve the data from the server of the electoral authorities and show the voter which vote he has cast based on this tracking number.”* (P06); or *“The encrypted vote is then forwarded via the Internet and placed in the digital ballot box and added. And this ballot box is stored on a server where all election results are then uploaded. And where they can then be retrieved again by the voter after the election, for example by the verification app.”* (P12).

Verification Phase. The procedure itself for individual verification was understood, but the overall process remained unclear. As described in Section 2, the app contains information about the tracking number in the verification phase, but does not provide details on how the connection between trackers and votes are made. Nevertheless, seven participants described how their vote was linked to their tracking number. For example, participant P19 said: *“It’s probably generated from some data from my smartphone because it has to store it somehow because I didn’t have to enter it anywhere”.*

Only one participant (P17) misunderstood the content of the bulletin board and thought that it shows links to voters: *“We receive the list of, as far as I understood, all voters.”*

4.3 Meaning of the Verification Phase

Besides technical details of verification, we asked the participants to explain the purpose of the tracking number and the bulletin board. Finally, we asked them why verification is required. We describe their answers based on comments about 1) *individual verifiability*, 2) *universal verifiability*, and 3) *general purpose*.

Individual Verifiability. 23 of 24 participants explained individual verification with the tracking numbers. Fifteen of them explicitly mentioned the *correctness* of recorded votes, others explained a comparison between the recorded vote and their vote intention. For example: *“Here is a list of numbers and votes and these*

will be sent to my smartphone and I can compare them with my tracking number to see if what I voted for finally reached the server.” (P04); or *“I as a voter I can check if I have voted correctly.”* (P12).

Universal Verifiability. Participants expressed difficulties in explaining why they can see all votes instead of only their own. Eleven participants talked about recounting even if the app did not offer an intuitive way to do it, for example, P02 said: *“It wasn’t possible with the app but theoretically I could use all the votes to check if everything is correct and of course I need all the votes for that.”* Two participants also mentioned that the bulletin board was necessary to find their own vote as the tracking number was stored locally on the phone. For instance, P09: *“I need them all at some point because I have to find my own.”*

General Purpose. Three participants compared Selene’s features to actual voting systems that do not allow them to verify. For instance, P01 said: *“It offers a way to see if the vote is present at all because in old systems that’s not there at all”*. Four participants mentioned transparency as a goal, e.g.: *“[...] that we can offer the citizens a certain transparency.”* (P16); or *“I do think it is necessary just for the sake of transparency.”* (P23). Five participants also mentioned it as a confidence or trust feature, like P11 mentioned that it is *“to give a little more confidence.”*

4.4 Security Concerns

All participants mentioned different security concerns and considerations during the interviews. With respect to the previous section about the meaning of the verification phase, the correctness of the result and the integrity of the elections were mentioned by 15 participants as a security concern, e.g., P07 said: *“There is a bit of certainty that it was done correctly”*.

All participants noticed the encryption of the votes. Three participants questioned the encryption of other parts, for instance, the encryption of the channel between the app and the server and encryption of the data on the server itself, e.g., P05: *“I didn’t pay attention to it but I hope there was an encrypted connection to the infrastructure of the election office, via the Internet”*.

Sixteen participants mentioned that they wanted to have information regarding the verification phase in advance during the registration process for three different reasons: First, four said that it would help them decide whether they choose this app to cast their vote. Second, eleven mentioned that they would evaluate the reliability of the app, and third six said that it would provide more time to voters to understand how it works.

Nine participants questioned the implementation and said it had a direct impact on their trust. For example, some participants questioned the origin of the tracking number and whether it indeed shows their cast vote. For instance, P15 said: *“It was cryptic in the sense that I just received it from the server, I couldn’t understand if this is really the vote I cast.”* Furthermore, three participants

questioned the system by describing it in a skeptical way, e.g. P05: *“Hopefully the votes cast are stored there in encrypted form.”*

Attacks or bugs in the system were mentioned by nine participants, such as ballot rigging or possible manipulations: *“I would know theoretically whether they were manipulated or not”* (P20); *“The votes that were cast could also, as it was the case with me once, simply have been wrong somehow”* (P10).

Four participants mentioned anonymity as one of their concerns linked to the tracking number, e.g. P01: *“They are anonymous because nobody has any idea which tracking number the other person has.”*

Concerns about dispute resolution were also mentioned by three participants, two of them noticed that they cannot prove how they voted afterwards so questioned how to prove a mistake, e.g. P19: *“Somehow nobody can prove that you have actually chosen something else.”*

Only two participants mentioned the decryption of the tracking number in the app, and one of them questioned the origin of the keys in use in the app. Finally, two participants believed that from the bulletin board, one might figure out whom a specific voter voted for, hence breaking privacy.

4.5 Unnecessary Steps

Thirteen participants perceived some verification steps as unnecessary. In particular, the bulletin board was considered as useless. For example, P09 mentioned: *“If I already know what my tracker is, I honestly don’t see the point of seeing all of them.”* Even if participants mentioned recounting of votes as an option, they were not interested in doing it. For example, P06 said: *“You could say that it’s about comparing this list of votes with the overall election results, of course. But then again, I do not see how the normal voter should actually do that with several million eligible voters or several million votes cast”*.

Even if most of the participants understood the purpose of individual verification, two were not convinced by the provided information and questioned the need of making the system verifiable. For instance, P03 mentioned that *“It also doesn’t help me to check if this is really part of the final result or not.”*

5 Discussion

5.1 Lessons Learned

In this section, we discuss observations from our interview study and its results.

Impact of Vote Manipulation. In our study, the participants executed the protocol twice. In the second round, the cast vote was modified, since vote manipulations are recommended to measure the execution of a given mental task [21, 29]. In our case, this manipulation showed a possible source of errors in an Internet election to the participants. Twenty participants clearly reported it to the examiner, we cannot tell if the four remaining participants did not

understand or lacked confidence to highlight the issue, as mentioned in [23]. In previous studies on the Selene protocol [11,37], such a threat was not shown to the participants and the participants had more difficulties explaining why verification is useful. Combined to the explanations provided in the app experiencing an incorrect vote had a positive impact on the understanding of the participants. Indeed, almost all of them had a good idea of the meaning of the verification phase, in particular, showing the correctness of the result. Of course, we cannot trigger such an attack and make voters experience errors in a real election, but it shows that being aware of the risks helps understanding the meaning to a given task.

Different Needs for the Users. Several participants expressed a need of learning more details about the setup and the origin of tracking numbers, or wanted to have additional proofs. The correct understanding of the available features was not enough to convince them. This had a negative impact on participants as it raised many questions and affected their trust in Selene. Several participants said that they would prefer to have information regarding the system before the elections to ensure their correct understanding and the reliability of the system. Selene can provide additional mathematical proofs, and it is specified in the original protocol that more verifiable data is available to the public.

More than half of the participants did not consider it necessary to access the complete list of votes, even if some of them explained the possibility of recounting the votes and the transparency that it provides. As mentioned in Section 2, one important security feature in the protocol design, not tested here, is the accessibility of the bulletin board in order to let a possibly coerced voter choose another tracking number. It has been highlighted in a previous study that this missing feature might help the voters to understand better the opportunity of accessing the complete list of votes [37].

Bada *et al.* [6] acknowledged that risk awareness and understanding are prerequisites to change security behaviours. However, they also highlighted that additional factors must be taken into considerations, in particular the adaptability to the audience and to its needs is encouraged.

Impact of the Security Communication. In the apps, security-related information was communicated on several screens. First, several loading screens between the direct interactions with the users showed the following information: authentication, encryption of votes, and decryption of the tracking numbers. Furthermore, before the vote encryption, the users were explicitly pushing a button indicating “Encrypt” to encrypt their vote. Finally, the anonymity of the trackers was explained inside the app before the verification. In two prior studies of Selene [11,37], the authors highlighted that the security, even if visible, remained unseen by the participants of their study. A possible reason for that could have been the lack of interactivity with the security features. In our study, using an “encryption button”, we observed that all participants mentioned this feature. However, the drawings revealed that the location of the encryption com-

putation sometimes remained unclear. This might be due to a lack of knowledge in security properties and software design but it did not have a negative impact on the participants. On the contrary, interacting with encryption had a positive impact on the security concerns of the participants, as it made them aware that a security feature is implemented. Similarly to previous studies, participants did not notice the decryption of tracking number, since it was mentioned only in the loading screen.

5.2 Recommendations

Based on our observations and the lessons learned above, we distill four recommendations to inform the design of future verifiable voting schemes, applicable to Selene but also other verifiable schemes.

1. **Provide information to support transparency.** The security concern regarding correctness was often mentioned during interviews when explaining the meaning of the verification phase. As discussed above, one reason might be the impact of the vote manipulation but we can also mention the verification app that gives several insights on verifiability to voters, among which the correctness of records was cited. On the other side, few participants mentioned transparency, but this did not justify the display of all votes for them. For future implementations, our first recommendation concerns the clear designation of each entity that a user might deal with and their purpose to ensure a complete understanding of the expected tasks.
2. **Provide education materials about risks.** The vote manipulation made participants aware of possible risks related to online voting and let them better understand the meaning of the verification phase. We highlighted that risk communication, control over verifiability procedures and easy security interactions can lead to a better understanding of the tasks one must perform. To be accepted, an Internet voting system needs to convince enough voters to perform those additional individual checks. It is recommended to provide voters with materials to educate themselves on possible risks related to Internet voting, and how to counter them. The Swiss Post voting protocol, for instance, provides such access to voters [30]. In addition, informative materials, such as TV spots or websites, could use an incorrect vote and show voters how it can be detected with a verification mechanism.
3. **Provide simple interactions with a security emphasis.** The interaction with the encryption button has shown to raise the awareness of participants regarding the security implementation. Other screens in the app where security was shown without interactivity were mentioned by participants only twice. This confirms the previous studies with this voting protocol [11, 37] and related voting schemes [20]. Therefore, we recommend to communicate security through simple interactions whenever possible. Following the example of the encryption, naming the security tasks on a simple interaction like pushing a button is enough to raise the awareness of users.

4. **Provide different levels of verification.** Many participants understood the verification features but were not always convinced by them, while other participants considered certain information as unnecessary. Hence, we recommend organizing the verifiable data and information such that it is displayed only on demand. We can distinguish three levels of verifiable information: 1) a minimal display, showing the individual vote to be verified only, 2) a full display, showing the individual vote and the entire bulletin board, and 3) a full access for experts, containing detailed specification on how to perform additional checks. This last level will let any expert (eligible to vote or not) verify more steps of the protocol.

6 Conclusion and Future Work

We investigated mental models of 24 participants using the Selene Internet voting protocol. We let them draw their understanding of voting and verification using Selene and we interviewed them. The mental models demonstrated different levels of sophistication, security concerns, and understanding. We also highlighted that the tracker used to verify their individual votes was not enough for all users; in contrast the full bulletin board given for universal verifiability was stressed as unnecessary. Furthermore, we found that direct interaction with security features had a positive impact on the awareness of a secure implementation. These findings helped us to understand the users' expectations in Internet voting applications, and highlight their need of transparency and correctness for elections, as well as more interactions with security features and more control on the process. Some features were not explored yet in this study and as future work, we will test their impact on the voters' understanding and trust in the system. Also, the mental models of voters with paper voting systems might differ and a comparison of voters' models with paper and internet voting schemes will be explored. Finally, having a misconception of how the system works might not prevent a voter to use it correctly. This is also an interesting future direction for our research.

Acknowledgements This research was supported by the Luxembourg National Research Fund (FNR), under the joint INTER project SeVoTe (INTER/FNRS/15/11106658/SeVoTe) as well as the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 251805230/GRK 2050.

References

1. Abdi, N., Ramokapane, K.M., Such, J.M.: More than smart speakers: Security and privacy perceptions of smart home personal assistants. In: Proc. Symposium on Usable Privacy and Security. pp. 1–16. USENIX Association, Berkeley, CA, USA (2019)
2. Acemyan, C.Z., Kortum, P., Byrne, M.D., Wallach, D.S.: Usability of voter verifiable, end-to-end voting systems: Baseline data for Helios, Prêt à Voter, and Scantegrity II. The USENIX Journal of Election Technology and Systems **2**(3), 26–56 (2014)

3. Acemyan, C.Z., Kortum, P.T., Byrne, M.D., Wallach, D.S.: Users’ mental models for three end-to-end voting systems: Helios, prêt à voter, and scantegrity II. In: International Conference on Human Aspects of Information Security, Privacy, and Trust (2015)
4. Adida, B.: Advances in Cryptographic Voting Systems. Ph.D. thesis, Massachusetts Institute of Technology (2006)
5. Adida, B.: Helios: Web-based Open-Audit Voting. In: Proc. USENIX Security Symposium. pp. 335–348. USENIX Association, Berkeley, CA, USA (2008)
6. Bada, M., Sasse, A.M., Nurse, J.R.C.: Cyber security awareness campaigns: Why do they fail to change behaviour? (2019)
7. Ben-Nun, J., Fahri, N., Llewellyn, M., Riva, B., Rosen, A., Ta-Shma, A., Wikström, D.: A new implementation of a dual (paper and cryptographic) voting system. In: International Conference on Electronic Voting (2012)
8. Benaloh, J., Byrne, M., Kortum, P.T., McBurnett, N., Pereira, O., Stark, P.B., Wallach, D.S.: Star-vote: A secure, transparent, auditable, and reliable voting system. CoRR (2012)
9. Borgman, C.L.: The user’s mental model of an information retrieval system: An experiment on a prototype online catalog. *International Journal of man-machine studies* **24**(1), 47–64 (1986)
10. Cohen, J.: A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* **20**(1), 37–46 (1960)
11. Distler, V., Zollinger, M., Lallemand, C., Rønne, P.B., Ryan, P.Y.A., Koenig, V.: Security - visible, yet unseen? In: Proc. CHI Conference on Human Factors in Computing Systems (2019)
12. Estonian national electoral committee. <https://www.valimised.ee/en/archive/statistics-about-internet-voting-estonia> (2019)
13. Fuglerud, K.S., Røssvoll, T.H.: An evaluation of web-based voting usability and accessibility. *Universal Access in the Information Society* **11**(4), 359–373 (2012). <https://doi.org/10.1007/s10209-011-0253-9>
14. Johnson-Laird, P.N.: *Mental models: Towards a cognitive science of language, inference, and consciousness*. No. 6, Harvard University Press (1983)
15. Kang, R., Dabbish, L., Fruchter, N., Kiesler, S.: “my data just goes everywhere:” user mental models of the internet and implications for privacy and security. In: Proc. Symposium on Usable Privacy and Security. pp. 39–52. USENIX Association, Berkeley, CA, USA (2015)
16. Kulesza, T., Stumpf, S., Burnett, M., Yang, S., Kwan, I., Wong, W.: Too much, too little, or just right? ways explanations impact end users’ mental models. In: Proc. IEEE Symposium on Visual Languages and Human Centric Computing. pp. 3–10 (Sep 2013). <https://doi.org/10.1109/VLHCC.2013.6645235>
17. Lallemand, C., Koenig, V.: Lab testing beyond usability: Challenges and recommendations for assessing user experiences. *Journal of Usability Studies* (2017)
18. Levitt, S.D., List, J.A.: What do laboratory experiments tell us about the real world. *Journal of Economic Perspectives* (2007)
19. Marky, K., Kulyk, O., Renaud, K., Volkamer, M.: What did I really vote for? On the usability of verifiable e-voting schemes. In: Proc. CHI Conference on Human Factors in Computing Systems. pp. 176:1–176:13. ACM, New York, NY, USA (2018). <https://doi.org/10.1145/3173574.3173750>
20. Marky, K., Zimmermann, V., Funk, M., Daubert, J., Bleck, K., Mühlhäuser, M.: Improving the usability and ux of the swiss internet voting interface. In: Proc. CHI Conference on Human Factors in Computing Systems. pp. 640:1–640:13. ACM, New York, NY, USA (2020). <https://doi.org/10.1145/3313831.3376769>

21. Marky, K., Zollinger, M.L., Funk, M., Ryan, P.Y., Mühlhäuser, M.: How to assess the usability metrics of e-voting schemes. In: *Financial Cryptography and Data Security, Workshop on Advances in Secure Electronic Voting* (2019)
22. Marky, K., Zollinger, M.L., Roenne, P.B., Ryan, P.Y., Grube, T., Kunze, K.: Investigating usability and user experience of individually verifiable internet voting schemes. *ACM Transactions on Computer-Human Interaction* **28**(5) (2021), <https://kaikunze.de/papers/pdf/marky2021investigating.pdf>
23. Moher, E., Clark, J., Essex, A.: Diffusion of voter responsibility: Potential failings in e2e voter receipt checking. *USENIX Journal of Election Technology and Systems (JETTS)* **1**(3), 1–17 (Dec 2014), <https://www.usenix.org/jets/issues/0301/moher>
24. Nestas, L., Hole, K.: Building and maintaining trust in internet voting. *Computer* **45**(5), 74–80 (May 2012). <https://doi.org/10.1109/MC.2012.35>
25. Norman, D.A.: Some observations on mental models. In: *Mental models*, pp. 15–22. Psychology Press (2014)
26. Ryan, P.Y.A., Rønne, P.B., Iovino, V.: Selene: Voting with transparent verifiability and coercion-mitigation. In: *Proc. Financial Cryptography and Data Security* (2016)
27. Sallal, M., Schneider, S., Casey, M., Dragan, C., Dupressoir, F., Riley, L., Treharne, H., Wadsworth, J., Wright, P.: Vmv: Augmenting an internet voting system with selene verifiability (2019)
28. Schneider, S., Llewellyn, M., Culnane, C., Heather, J., Srinivasan, S., Xia, Z.: Focus group views on prêt à voter 1.0. In: *International Workshop on Requirements Engineering for Electronic Voting Systems* (2011)
29. Selker, T., Rosenzweig, E., Pandolfo, A.: A methodology for testing voting systems. *Journal of Usability Studies* **2**(1), 7–21 (Nov 2006), <http://dl.acm.org/citation.cfm?id=2835536.2835538>
30. Serdült, U., Germann, M., Mendez, F., Portenier, A., Wellig, C.: Fifteen years of internet voting in switzerland [history, governance and use]. In: *Proc. Second International Conference on eDemocracy eGovernment* (2015)
31. Serdült, U., Kryssanov, V.: Internet Voting User Rates and Trust in Switzerland. In: *Proc. International Joint Conference on Electronic Voting*. pp. 211–212 (2018), <https://doi.org/10.5167/uzh-156867>
32. Tullio, J., Dey, A.K., Chalecki, J., Fogarty, J.: How it works: A field study of non-technical users interacting with an intelligent system. In: *Proc. SIGCHI Conference on Human Factors in Computing Systems*. p. 31–40. Association for Computing Machinery, New York, NY, USA (2007). <https://doi.org/10.1145/1240624.1240630>
33. Warkentin, M., Sharma, S., Gefen, D., Rose, G.M., Pavlou, P.: Social identity and trust in internet-based voting adoption. *Government Information Quarterly* **35**(2), 195–209 (2018)
34. Wash, R.: Folk models of home computer security. In: *Proc. Symposium on Usable Privacy and Security*. Association for Computing Machinery, New York, NY, USA (2010). <https://doi.org/10.1145/1837110.1837125>
35. Zeng, E., Mare, S., Roesner, F.: End user security & privacy concerns with smart homes. In: *Proc. Symposium on Usable Privacy and Security*. pp. 65–80. USENIX Association, Berkeley, CA, USA (2017)
36. Zimmermann, V., Gerber, P., Marky, K., Böck, L., Kirchbuchner, F.: Assessing users' privacy and security concerns of smart home technologies. *i-com* **18**(3), 197–216 (2019)
37. Zollinger, M., Distler, V., Rønne, P.B., Ryan, P.Y., Lallemand, C., Koenig, V.: User experience design for e-voting: How mental models align with security mechanisms. In: *Proc. Joint International Conference on Electronic Voting* (2019)

PhD Colloquium

Evaluating Voter's Assessment of Verifiability Using Text Classification with Machine Learning

Ehsan ESTAJI

University of Luxembourg
ehsan.estaji@uni.lu

Abstract. We develop a pipeline which collects the results of an online survey about general understanding of people about verifiability. But in this case, this survey is shared on social media and we will collect data automatically by web crawler tools and using machine learning algorithm will train the algorithm to classify the received answers.

Keywords: E-voting scheme · Data Mining · Machine Learning · Online Survey · Text Classifier.

1 Introduction

Elections are indispensable chunks of democracy. To boost voter's attendance several countries (e.g. Australia, Estonia and Switzerland) employ various manners to use technology and particularly Internet. Since the integrity of an election is a crucial ingredient, secure and robust systems recommended to defend democratic doctrines. Verification procedure, principally, designate to assure the authentic result and accomplishment of an election. Generally, to bring about this Internet voting schemes sacrifice their usability by employing cryptographic protocols. Recent studies on voting affirmed that security is one the major worries of voters. Actually some of them notice the verification operation as confidentiality infringement [3,2], or their obligation was interrogated by the voters [1]. To retaliate this drawback it is vital to abate voter's communication with cryptography in the course of providing verifiability. Following this perspective, the Internet voting scheme Selene introduced. Nonetheless the Selene's usability is demonstrated in studies, but it is not acceptable in satisfying voters about the correct handling of votes. The voter's intellectual pattern might be a justification for not cooperating with the verification. There are qualitative studies concerning this problem to reveal more details about the mental model of voters.

2 Machine Learning Overview

We are currently living in a "data era," where a colossal load of data is collected and stockpiled regularly. On the basis of this burgeoning quantity of data, machine learning methods have become inevitable. Text classification is a machine learning method that automatically corresponds tags or categories to text. With the aid of natural language processing (NLP), text classifiers can inspect and sort text by sentiment, topic, and customer intent – faster and more accurately than purpose.

3 Main Idea

One of the biggest hardship conducting qualitative studies is that they usually evaluate a little number of participants and even that takes along time. This motivates us to overhaul the previous method and mix it with ML methods. In the first place, we need to design an online survey to ask our desired questions for estimating voter's mental model. Then share our online survey on several social media platforms and ask people to answer these surveys (Most likely we should have a nice incentive to convince them to engage on answering the survey). Our idea is to automate the whole procedure as much as possible. The web crawlers are our big tools to collect our required data automatically. Then we should polish the gathered data and prepare them for processing on deep learning model which is trained before. To be more precise we can specify the whole process on the following smaller steps:

- **Designing an online survey:** (it might contains more graphical features since in this case we loss the personal counselling of participants media platforms.
- **Ethical Considerations:** Accompanying the emerge of data mining from social media, its ethical guideline was announced. For example, even the information is public it must always consider with respect to individuals' privacy.
- **Limitations** Incorrect information can also be the main drawback of data mining systems. When a user interacts on the social platform, he doesn't assure us that he has been pristine in his thoughts.
- **Data Engineering:** Data preparation and cleansing tasks can take a substantial amount of time. Surveys of machine learning developers and data scientists show that the data collection and preparation steps can take up to 80 % of a machine learning project's time.
- **Choosing a text classifier:** Model selection is the process of choosing between different machine learning approaches. So in short, different models.
- **Initializing the model:** Now it's time to feed our model and train it.
- **Ready to classify:** In this stage, our model is trained and ready to predict(here we mean the classification . So it takes a text and add a tag to this text.

References

1. Claudia Z. Acemyan, Philip Kortum, Michael D. Byrne, and Dan S. Wallach. Usability of voter verifiable, end-to-end voting systems: Baseline data for Helios, Prêt à Voter, and Scantegrity II. *The USENIX Journal of Election Technology and Systems*, 2(3):26–56, 2014.
2. L. Nestas and K. Hole. Building and maintaining trust in internet voting. *Computer*, 45(5):74–80, May 2012.
3. Steve Schneider, Morgan Llewellyn, Chris Culnane, James Heather, Sriramkrishnan Srinivasan, and Zhe Xia. Focus group views on prêt à voter 1.0. In *2011 International Workshop on Requirements Engineering for Electronic Voting Systems, REVOTE 2011*, 2011.

Fault Tolerant Result Collation Scheme for Nigerian Electoral System

Olaoluwa Adeoye Olayinka and Ganiyu Adesina Rafiu

Ladoke Akintola University of Technology, Nigeria
yinkani@gmail.com

Abstract. We briefly describe a fault tolerant election result collation scheme for a hybrid election system which is conceptualized on homomorphic tally across multiple levels of collation hierarchy. This is to attain better user experience and inspire voter participation.

Keywords: Fault tolerance, result collation, homomorphic tally

1 Introduction

Prevalent voter abstention remains a challenge in most democracies with an average abstention rate of as much as 40–50 percent [1]. Democracy is fundamentally threatened by voter apathy occasioned by lack of confidence in the outcome of the electoral processes. Without trust or confidence that elections will produce fair outcomes, voters may choose to stay home, thereby compromising the legitimacy of the government [2]. This is evident in Nigerian case where voter turnout is abysmally low. For instance, during the 2019 national election, out of 84,004,084 registered voters, only 27,324,583 valid votes were recorded, while 56,679,501 voters were disenfranchised [3]. The challenges of delay in the final collation of results, integrity of result and security of collation officers (CF) from the risk of travelling in the dead of the night to get results submitted for final collation coupled with falsification of figures and recorded loss of life are unnerving [4]. Though Nigerian election management body is unrelenting about delivering a free and fair election, reports from both external and local observers still show that Nigerian elections were marred by poor organization, lack of essential transparency [5], widespread procedural irregularities, and significant evidence of fraud particularly during the result collation process and numerous incidence of violence. To change this trend, elections must be secured, the outcome must be convincing to all parties and safety of all stake holders must be ensured.

Majority of hybrid voting schemes in literature were designed to prevent electoral fraud during the ballot casting phase of elections in developed countries while result collation (RC) with fault tolerance and verifiability are not considered. Also most studies focused on the security requirements and technical specifications of its design without significant attention paid to social, political and environmental factors which has arguably determine voter confidence and participation. This approach considers the Nigerian context, where violence is a ready option by politicians, to present a fault

tolerant secured result collation scheme. This will be embedded with disaster recovery plan to ensure result integrity in the eventuality of an attack. It will be incorporated in the Nigerian voting system to reduce the use of violence by politicians as its usage will not affect the outcome of the elections.

The scheme

It is an enhanced result collation system using fault tolerance and homomorphic cryptography for security. Voters' intentions captured on paper ballots aggregated in ballot box as marked ballots are tallied publicly within the polling unit while stake holders observe as demanded by the electoral law. Modifications include; generation of electronic copy of the result sheet (ER-sheet) in electronic format by scanning a signed copy of the result sheet with provided equipment or filling an electronic form of the result obtained and sending it in an encrypted form differently to the different layers in the result collation hierarchy. ER-sheet contains the result of voting from the polling unit with signature of observing parties who observed the tallying process. The aggregation will be done with Paillier Cryptosystem, which allows constructing a publicly verifiable proof that all lower-level results were included in the final tally. To achieve verifiability of result collation process, a code will be generated from each polling unit and attached with each result displayed so all interested voters could check on the Bulletin Board. The plaintext paper copy is demanded by the law while encryption, homomorphic tallying and fault tolerance enhances integrity, verification, disaster recovery and voters trust.

Contribution

Our main contribution is to develop a fault tolerant hybrid (paper and electronic) tallying system that is easier to use, deploy, verify and audit for better voters confidence and participation. This work addresses problems associated with accessibility, verifiability and result collation, in a bid to improve trust and participation of voters. The method developed distributes the responsibility of delivering reliable result among various collation hierarchy and groups in a way that the final result gives higher confidence in the correctness of the overall outcome than can be obtained by a singular channel as it used to be. This scheme also introduces a disaster recovery plan to dissuade the use of violence during result collation (RC).

References

1. David M Farrell. Electoral systems: a comparative introduction. Macmillan International Higher Education, 2011.
2. Kervel, Yann. 2009. Election Management Bodies and Public Confidence in Elections: Lessons from Latin America. Washington, DC: IFES. 2009.
3. Sule Babayo. The 2019 Presidential Election in Nigeria: An analysis of the voting pattern, issues and impact Malaysian Journal of Society and Space 15 issue 2 (129-140) 2019.
4. Ayeni T.P and Esan A.O. The impact of ICT in the conduct of election Nigeria. American Journal of Computer Science and Information Technology, vol. 6, pp. 1 -6, 2018
5. European Union Election Observation Mission Nigeria. General elections 2019: first preliminary statement, 25 February 2019

A New Technique for Deniable Vote Updating

Najmeh Soroush

SnT, University of Luxembourg, najmeh.soroush@uni.lu

1 Introduction

In any e-voting protocols it is important to have a *coercion-resistance* property. On the other hand, a secure e-voting protocol needs to be secure, universally verifiable, practical and usable. Despite numerous attempts, designing a coercion-resistance e-voting which simultaneously provides all of the aforementioned basic properties is still an open problem and many voting schemes have been designed that aim to meet this property by proposing some counteracts at a technical level. Our idea is to design a new voting scheme that offers receipt-freeness, coercion-resistance and end-to-end verifiability properties.

In *coercion-resistance* systems, each coerced voter has the option to run some *counter-strategy* to achieve her own goal instead of obeying the coercer. At the same time, the coercer should not be able to distinguish whether the coerced voter followed his instructions or ran the counter-strategy. From a technical perspective, there exist three different approaches in the literature which implement this concept: *fake credentials* [2] *masking* [1] and *deniable vote updating* [3] which enables each voter to overwrite her previously submitted ballot, that she may have cast under coercion.

In both fake credential and masking approach, the counter-strategies appear to be hardly usable by human voters. It is therefore likely that these two concepts are ineffective in real practical elections and achieving coercion-resistance via deniable vote updating is more promising.

2 Main Idea

Our idea is to propose another coercion-resistance voting protocol is the so-called deniable vote updating. The idea is simple: while the voter might be coerced to cast a particular vote in presence of an adversary, she can cast another vote, overwriting her previous one, when the adversary is gone. The coercion resistance property, in particular, is achieved due to deniability of vote updating the adversary should be unable to tell whether the voter has cast another vote, even if the voter would try to prove that they did not do it.

The starting point is using a public-key encryption scheme that allows to re-encrypt a ciphertext without knowing the original message. This allows the voter to encrypt her vote and then some other parties to re-encrypt the voter's ballot. Then for sake of verifiability, we need to make both encryption and re-encryption steps, verifiable by using some zero-knowledge proof system and for sake of coercion-resistance these two step should be indistinguishable.

This is one of the main challenges we need to deal with. On one hand we need to assign the ballot to legitimate voter and this usually is done by voter signing the ballot which is required by voter knowing some secret key. On the other hand the party who re-encrypt the ballot needs to prove the relation between two ciphertexts by proof of knowledge of the randomness used in re-encryption algorithm. These two proofs should be indistinguishable while the later party has to do it without knowing the secret key of the voter. Our solution for this challenge is either using some zero-knowledge proof system or none-interactive witness-indistinguishable proof of knowledge, then define a system of equation that connects all part of the ballot to the public key of the voter, and finally generate a proof of knowledge of the solution for the system of equation.

Based on the above idea we propose a protocol that in its *registration phase* for each legitimate voter V_i a pair of key (pk_i, sk_i) is generated. Then a list $list_i = [pk_i, b_0 = Enc(0)]$ is published on the bulletin board.

In *voting phase* some valid components are added to the voter list, $list_i$. There are two ways to generate and add valid components. The first one, “fresh-vote” is generated by the honest voter, the owner of the secret key and the second one “reEncrypt-vote” is generated by the posting trustee. To generate a fresh-vote the voter V_i chooses her choice $vote_i$ and encrypts it then add the proof (proof of knowledge) of the and also a proof of knowledge of her secret key. Formally the voter run the $NIZK(x, w)$ algorithm to generate π_{ct} for relation $\mathbb{R}(x, w)$ which in this case $P_1(x, w)$ is true. The reEncrypt-vote is generated by the “Posting Trustee” in random times as follows: Choose some random number(s) and reEncrypt the last component of $list_i$ and run the $NIZK(x, w)$ algorithm to generate π_{ct} for relation $\mathbb{R}(x, w)$ which in this case $P_2(x, w)$ is true.

$$\begin{aligned} \mathbb{R}(x, w) = \text{TRUE} &\iff (P_1(x, w) = \text{TRUE}) \vee (P_2(x, w) = \text{TRUE}) \\ x = (list^{pk_i} = [pk_i, b_0, (b_1, \pi_1), \dots, (b_j, \pi_j)], b), w_1 = (vote, r^{ct}, sk_i), w_2 = (r^{re. enc}) \\ P_1(x, w_1) = \text{TRUE} &\iff (b = Enc_{PK}(vote; r^{ct})) \wedge (vote \in cList) \wedge PoK(sk_i) \\ P_2(x, w_2) = \text{TRUE} &\iff (b = Re.Enc_{PK}(b_j; r^{re. enc})) \end{aligned}$$

References

1. M. Backes, M. Gagné, and M. Skoruppa. Using mobile device communication to strengthen e-Voting protocols. In A. Sadeghi and S. Foresti, editors, *Proceedings of the 12th annual ACM Workshop on Privacy in the Electronic Society, WPES 2013, Berlin, Germany, November 4, 2013*.
2. E. Estaji, T. Haines, K. Gjøsteen, P. B. Rønne, P. Y. A. Ryan, and N. Soroush. Revisiting practical and usable coercion-resistant remote e-voting. In Krimmer, Volkamer, Beckert, R. Küsters, O. Kulyk, D. Duenas-Cid, and M. Solvak, editors, *Electronic Voting*, pages 50–66, Cham, 2020. Springer International Publishing.
3. O. Kulyk, Teague V, and M. Volkamer. Extending Helios Towards Private Eligibility Verifiability. In R. Haenni, R. E. Koenig, and D. Wikström, editors, *E-Voting and Identity - 5th International Conference, VoteID 2015, Switzerland, September 2015, Proceedings*, Lecture Notes in Computer Science.

Why does e-voting have to be perfect?

Tamara Finogina

UPC & Scytl Election Technologies, Spain
tamara.finogina@scytl.com

It is well known that elections matter and that people must be confident in the fairness of election outcomes for democracy to function. It is also known that computers are vulnerable and can be hacked. As a consequence, the combination of two, e-voting, is doomed to be mistrusted.

Many experts insist that computers should not be trusted for voting, especially if it is done remotely. The main fear is that an adversary can undetectably modify enough ballots to change the result of an election or influence the counting process. Additionally, there are concerns regarding privacy and coercion, which make many people see paper-based voting as a more trustworthy option.

It was not always like that, however. Ironically, the very first mechanization done in a polling place was for preventing fraud in paper-based voting. Early machines were counting the number of ballots to prevent ballot stuffing [4]. This was more efficient than a glass ballot box, which could prevent votes insertion, but did not guarantee that *all of them* would reach the tallying. Thus the process of mechanization started with an intent to shift trust from electoral officials to machines.

Nowadays, relying on computers is often treated as a source of problems rather than a solution, even though machines are still better suited for repetitive tasks and calculations. The study of the hand counting of votes finds error rates of up to 2% [3] for candidate counts. This raises an interesting question of how to audit elections when the margin of victory is less than a human error and recount might give a different result. In the NSW 2015 state election, it would have taken to miscount only 10,398 votes out of 4.56 million, which is about 0.002% of all ballots, to shift 8 seats [2]. Such accuracy is beyond human capabilities.

Technically, every single election already relies on computers for maintaining a list of eligible voters. Nevertheless, e-voting is still largely feared. One of the reasons is contradictory requirements that make any e-voting system not perfect. Each solution is a tradeoff between benefits and complexity with some trust assumptions and expectations regarding voters. Since there is no consensus on what tradeoffs are optimal, every system is guilty of prioritizing one property over another because there are always circumstances where such a choice is unwise. While some residual risk is unavoidable, perfection is expected.

Consider cast-as-intended (CAI) verification, which is typically a must requirement. If a system requires a second device for vote verification, it is assumed that it affects equal participation since not everyone has several devices. Similarly, even a relatively powerful voting device that can handle cryptography might be deemed a problem to equal suffrage. Alternative approaches based on pre-delivered paper cards are accused of requiring trusted delivery channels

and trusting printing authorities. Whenever CAI is not provided, a system is regarded as vulnerable to vote modification. When it is, the system is guilty of not protecting voters from coercion. If coercion is prevented, then CAI must have been forgotten, or the coercer is too limited.

Encrypting vote in a voting device means that the voting device will see the plaintext and breach voter privacy. Vote codes might increase the risk of vote-selling. A public bulletin board threatens long-term privacy, but the integrity of a private one requires trusting system parts or auditors. The homomorphic tally cannot handle all types of elections. The mixing process does not protect against an Italian attack[1] on privacy. Trusting electoral officials is deemed to be dangerous, but a distributed system is called inefficient and impractical. Sharing a secret key with no threshold implies the risk of denial of service, while with a threshold, an adversary needs to corrupt fewer players to get the key. Complex verification procedures (e.g. cast-or-challenge, proof simulation, credential faking, etc.) are typically misunderstood, so voters perform verification poorly. Simpler procedures require some additional private data and extra trust assumptions. The list goes on and on. It is always possible to find why a system is not perfect.

Additional challenge comes from ambiguous definitions. For example, it is still unclear what coercer can do or whether a forced abstain should be considered as coercion attack. Similarly, there is no standard definition for privacy or CAI; all existing ones are tailored to a particular system. Should the Italian attack be considered an attack on privacy? If so, what about systems that are based on shuffling? Things get even more complicated if we consider electoral laws. For example, since e-voting is rarely the only voting channel, it has to be possible to link a voter to their vote to prevent multiple voting, thus no vote casting with anonymous credentials, which is the only known solution to strong coercion.

The open question is how *any* e-voting scheme can satisfy all requirements. Should each scheme customize definitions, or should we have standard ones? Should we have a consensus on what properties are more important, or should it be case-based? Should e-voting even try to solve all problems?

References

1. Bernhard, D., Cortier, V., Pereira, O., Warinschi, B.: Measuring vote privacy, revisited. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security. p. 941–952. CCS '12, Association for Computing Machinery, New York, NY, USA (2012)
2. Blom, M., Stuckey, P.J., Teague, V.J.: Computing the margin of victory in preferential parliamentary elections. In: Krimmer, R., Volkamer, M., Cortier, V., Goré, R., Hapsara, M., Serdült, U., Duenas-Cid, D. (eds.) Electronic Voting. pp. 1–16. Springer International Publishing, Cham (2018)
3. Goggin, S., Byrne, M., Gilbert, J.: Post-election auditing: Effects of procedure and ballot type on manual counting accuracy, efficiency, and auditor satisfaction and confidence. *Election Law Journal: Rules, Politics, and Policy* **11** (05 2012)
4. Jones, D.W., Simons, B.: Broken Ballots Will Your Vote Count? Center for the Study of Language and Information, Stanford, California (2012)

Scrutiny of Big-Scale E-Voting in Russia in 2021

Bogdan Romanov¹

¹University of Tartu
Tartu, Estonia
bogdan.romanov@ut.ee

Abstract. September 2021 has provided academia with the chance to unpack not only the e-ballot-box but also look at the nature of the first large-scale Internet voting precedent in the autocratic context of Russia with its eagerness for legitimisation and regime maintenance. In general, this might be a precedent for the oxymoron of autocratic e-/i-voting technologies in the CIS area, which occurred with the case of ‘sovereign Internet’. Therefore, this newly implemented phenomenon should be studied in the initial stages; otherwise, the *terminus quo* will be lost.

Keywords: I-voting · E-voting · Autocracies · Hybrid regimes

1 Diffusion of Innovations?

The ability to cast votes remotely via smartphones and other digital devices was granted to Russia’s citizens only in September 2021 and only in the limited number of federal units. Before that, there were several local attempts with city-level portals [1] implemented sporadically in Moscow, for example. However, the coronavirus pandemic constituted a convenient ‘window of opportunity’ for the test-run of the Internet voting, justified by the intentions to ensure the physical distance among the voters during the voting days [2]. Overall, this sounds like an instance of the government taking care of its citizens in a complicated epidemiologic context.

Unfortunately, the whole scenario of innovation diffusion is gloomed by the preceding voting for the constitutional amendments, which were held offline from 25th of June and until 1st of July — right during the middle stage of the first coronavirus wave. Thus, we can assume that pandemic was not the initial rationale for implementing Internet voting.

Even if one falls for the ‘benevolent ruler’ bait, there are several questions that should be answered to have a comprehensive understanding of the case — why does Russia need Internet voting? These questions will be addressed in the following section and will compile a baseline narrative for the publication series, which are devoted to the question of Internet voting in a competitive autocracy / hybrid regime.

2 Why and what to study

As I mentioned, remote voting is not used in all over 85 units of Russia; for the September elections, only seven subjects were selected for the pioneer launch [3] (Fig. 1). This selection raises the first question — why precisely these regions were selected? Are they the most loyal to the ruling party? Or do they have the sufficient capacities to conduct a successful test? A positive answer to any questions would imply the blissful outcome for the ruling party due to the unlevel electoral field. Still, it might also expose the fear of it and the abusive nature of Internet voting.

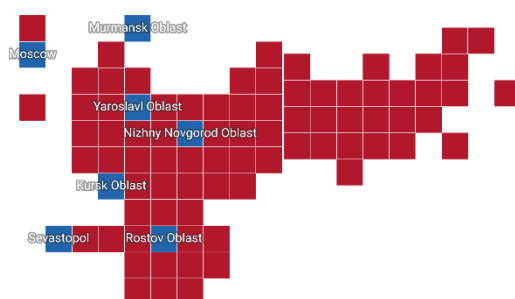


Fig. 1. A choropleth map with the indication of the regions, which introduced the Internet voting for the September 2021 elections

The second question primarily relates to the digital divide and the image of an average voter: who “attend” elections via smart devices, and how do they vote? This question might shed light on the institutional trust in the government and the presence or absence/suppression of i-voting’s democratisation effect.

The last question is about the rationality of introducing i-voting in an autocratic state with already twisted electoral mechanisms. The most common answer to the question ‘why do autocracies hold regular elections’ was ‘for the sake of legitimisation’, but why would anyone add another layer of it with the help of initially democratic innovation?

Undoubtedly, these three questions do not cover the whole phenomenon of autocratic e-/i-voting but at least initiate the discussion in this direction, since it seems that e-technologies obtained a more instrumental nature than ever and might be used by every regime.

References

1. Schlauffer, C.: Why do nondemocratic regimes promote “e-participation”? The case of Moscow’s active citizen online voting platform. *Governance*. (2020). <https://doi.org/10.1111/gove.12531>;
2. Krivonosova, I.: Electoral events in Russia during the COVID-19 pandemic: remote electronic voting, outdoor voting and other innovations. 19 (2020).

Regulating Internet voting by analogy: does it work?

Challenges and concerns for secret suffrage

Adrià Rodríguez-Pérez^{1, 2}[0000-0002-5581-1340]

¹ Scytl Election Technologies, S.L.U., 08021 Barcelona, Spain

² Universitat Rovira i Virgili, 43002 Tarragona, Spain
adria.rodriquez@scytl.com

Keywords: remote electronic voting, secret suffrage, analogy.

1 Introduction and *rationale*

The Italian government has recently adopted a decree for the gradual introduction of Internet voting (i-voting). One of the requirements in the decree is that “[t]he vote cast must not be attributable to the voter”¹. When the government of Catalonia submitted a draft i-voting law, the regulation enshrined a similar requirement². But governments are not alone, the first draft standards on i-voting for Canada’s municipal elections read “[t]he moment a vote is cast, the ballot must be severed from the voter” (8.1.1.1.)³.

In i-voting, the resort to analogies with paper-based elections is a common practice. It has been used by governments, courts, election observers, and academia. In Switzerland, the Federal Chancellery has acknowledged that requirements for secret suffrage in i-voting are “analogous” [sic] to those in postal voting⁴. In Estonia, analogies are usually drawn with postal voting’s double-envelop system and the country’s Supreme Court compared the possibility to re-vote with a “virtual voting booth”⁵.

Notwithstanding, remote electronic voting neither uses a double-envelop system neither the possibility to re-vote can be compared with a virtual voting booth. In our opinion, these analogies are not only insufficient, but they are counterproductive. For this reason, we suggest an alternative approach to assessing and regulating secret suffrage in i-voting. This approach could be extended to other electoral principles as well.

¹ In the original: “*Il voto espresso non deve essere riconducibile all’elettore*” (art. 4.4). Available at: <https://www.interno.gov.it/sites/default/files/2021-07/decreto_ministro_su_sperimentazione_voto_elettronico_9.7.2021.pdf>

² In the original: “[...] *a l’efecte que no es puguin relacionar els vots emesos amb els votants*” (art. 5.1). Available at: <<https://www.parlament.cat/document/bopc/178938.pdf#page=58>>

³ Available at: <<http://ciostrategycouncil.com/standards/technical-committees/technical-committee-review-2/can-ciosc-111-x/>>

⁴ In the original, “*Les exigences en matière de secret du vote posées aux trois systèmes de vote par Internet sont analogues à celles posées au vote par correspondance*”. Available at: <https://www.bk.admin.ch/dam/bk-intra/fr/dokumente/pore/bericht_des_bundesrateszuvoteelectronique-auswertungdereinfuehru.pdf.download.pdf/rapport_du_conseilfederalsurlevoteelectronique-evaluationdelamis.pdf>

⁵ See, for example: <<https://digikogu.taltech.ee/en/item/08baa269-0baa-40b7-9b6a-f3635d349bfc>>

2 Secret suffrage: a new framework for analysis?

To assess and regulate secret suffrage in i-voting, we want to understand the legal assets that this principle is aimed at protecting. By understanding the goals behind the institutionalisation of secret suffrage, we expect that clearer requirements could be elicited for the fulfilment of this principle in remote electronic voting.

At the international level, there is an extensive assessment as well as academic literature on the principle of secret suffrage (see for example [1]). In general, it “is [considered] an aspect of free suffrage, which aims to shield voters from any pressure that might result from the knowledge of his [sic] choice by third parties” [2]. The Parliamentary Assembly of the Council of Europe [3] has taken a further step by breaking down secret suffrage in three minimum standards:

- Individuality: each voter makes an individual choice.
- Confidentiality: only the voter should know how they have voted, and that the voter should be able to make their choice in private.
- Anonymity: there must be no link between the vote cast and the voter's identity.

3 Findings and open questions

Assessing secret suffrage against these standards is better suited to regulate i-voting. First, their focus is on the legal assets that need protection (e.g., the importance of anonymity is recognised, yet it is not prescribed when the link between the vote cast and the vote is “severed”). Second, this approach better bridges legal standards and technical requirements (e.g., confidentiality is usually achieved by means of encryption, but which cryptographic schemes can fulfil it? Which implementations are secure enough?)

Notwithstanding, some of the challenges in i-voting are not necessarily well captured by these three standards. In this sense, some questions remain open:

- How can the standard of individuality be understood in remote (electronic) voting?
- Is there no room for analogy at all? Could we balance contending principles based on existing assessments for remote paper-based voting?
- How to balance the need for accuracy in remote electronic voting regulations with the need to convey how i-voting works to non-expert publics?

References

1. Bertrand, Romain; Briquet, Jean-Louis ; Pels, Peter (eds.) (2006) *The Hidden History of the Secret Ballot*. Bloomington, Indiana: Indiana University Press.
2. Lécuyer, Yannick (2014) *Le droit à des élections libres*. Strasbourg: Council of Europe.
3. Parliamentary Assembly (2007) *Secret ballot – European code of conduct on secret balloting, including guidelines for politicians, observers and voters. Report*. Strasbourg: Council of Europe. Available at: <<https://assembly.coe.int/nw/xml/XRef/X2H-Xref-ViewHTML.asp?FileID=11738&lang=EN>> [accessed: 31/07/2021]

To i-vote or not to i-vote: Drivers and Barriers to the Implementation of Internet Voting

Nathan Licht^[1(0000-0002-6699-9879)]

Abstract. This paper investigates the drivers and barriers of internet voting and the implications of a global pandemic for the development of the respective technology. In contrast to the expected uptake in the early 2000s of internet voting, the technology is still rather seldomly used in election systems around the world. The paper at hand explores the different forces that drive or impede internet voting adoption from a political, social, legal, organizational, contextual, economic and technological perspective. In an exploratory approach, 18 expert interviews and extensive complementary desk research were conducted.

The findings identified 15 general drivers and 15 general barriers for the process of internet voting adoption. The evidence suggests that for a large part, the political features, trust and perception are the most pivotal factors to internet voting development.

Keywords: Internet Voting, Drivers and Barriers, Framework of Internet Voting, Technology adoption, e-Democracy

Previous research¹ either looked at part drivers and barriers or facilitating conditions in specific contexts. However, no comprehensive study has been conducted so far that investigates general drivers and barriers that are observable along the various adoption and trialled contexts. In line with that identified research gap, this paper poses the following research question: *What is driving internet voting and what barriers exist to further adoption?* To answer this question, the work at hand conducted in an exploratory way some 18 expert interviews and extensive complementary desk research. The applied methodology used for this paper, is explained subsequently.

To study what hinders or benefits the implementation of internet voting, I want to identify its drivers and barriers. To do so, I conducted a qualitative empirical study with a nonexperimental design including expert interviews, as promoted by Brown & Hale (Brown & Hale, 2014). This research was conducted using an inductive epistemological approach to acquire knowledge. The inductive process, as opposed to the deductive method, is a “bottom-up [technique in which] evidence is collected first, [from the observation of the world] and knowledge and theories built from this” (Ormston et al., 2014, p. 6). In order to guide the data analysis, a conceptual model was created *ad hoc*², integrating propositions included in five innovation diffusion theories. This model (see Figure 1) explains how different dimensions are embedded into one context that shapes the process of diffusion of internet voting, in an evolutionary process that is impacted by perceptions, adopter categories and discourses. Furthermore, it establishes the differentiation of internet voting adoption on two levels: political and individual. The model presents five dimensions, various stakeholders and factors that impact the technology acceptance process within societies. The following empirical research will explore the drivers and barriers and their allocation on the respective level of adoption

The data collection of this research was conducted via semi-structured expert interviews and complemented by desk research, allowing cross checking experts opinions with other sources. The study followed the framework provided by Krimmer’s mirabilis (Krimmer, 2012) that aids to identify the respective stakeholders involved in the implementation process of e-voting technology. In the context of this research, it was limited to three stakeholders: i) Media/observer, ii) election management and iii) inventors or vendors of voting technology. More precisely, it was focused on practitioners/EMBs/policymakers, scholars and election observers, as well as vendors or inventors of i-voting technology. A total of 18 interviews were conducted, transcribed, confirmed and analyzed in NVivo, via a deductive codification approach proposed by Mayring (Mayring, 2014). Data triangulation is granted through confirming cross-checking answers against either statement of other interviewees or findings from the literature (Flick, 2008)³.

¹ See Licht (2021b) for further information.

² For a better understanding, see: (Licht, 2021a)

³ The empirical findings will be cited as in-text citations with the interview number in brackets, in the following format: e.g., single citation (1), multiple citations (1;2; 3...).

From the empirical findings, I identified the drivers for the political decision level, to be universal access and accessibility for disabled voters, the pursuit of a contactless democracy, the aspiration to appear modern, the vendor's push, the process improvements, the perception of technology to be a neutral third party, the perception of increased administrative integrity, cost reductions, strong lobby groups, expected increase in voter turnouts and the presence of high socioeconomic power and well-established technical infrastructure. On the individual adoption level, I presented evidence that drivers exist such as convenience voting, spill-over effects within a digital society and the socioeconomic status of voters. Following barriers were identified for the political level adoption process: the middleman paradox, political crisis, change of government, security concerns, theoretical technical vulnerabilities, strong opposition from CSOs and academia, lack of a framework, lack of technological infrastructure, lack of verifiability, procedural barriers and the change of legal requirements. Barriers to adoption on the individual level have been identified as path dependency, cultural traditions, mistrust in technology and mistrust in EMBs and governments.

In conclusion, the research question can be answered through the depicted evidence showing that in total, 15 drivers, 12 on the political and three on the individual level and 15 barriers, with 11 on the political and four on the individual level, have been identified. Strong driving and impeding forces alike were found on the political level to be the absence or presence of political will, necessity, and the so-called middleman paradox. Even if the list of drivers and barriers is balanced, the reality shows that the implication of them is not following the same pattern, since the reduced number of adopters of i-voting brings to the conclusion that barriers play a more important role in the process of adoption than drivers. Further detailed case studies in selected countries could shed new light on how these drivers and barriers interact in particular administrative and political contexts and bring to the final decision of implementing or not i-voting. Additional research would be necessary in the field of trust in elections and specifically in election technology as well as the respective roles attributed to building or harming trust through the two discourse drivers that are academia or CSOs and on the other side the media. From the interviews it became apparent that these groups another study is merited but in which their roles especially in the individual diffusion process is further investigated. Possible questions to consider could be how can trust be measured and how can trust-building of new voting technologies be formed and what roles do media and academia play in that process? Last, to understand how various contexts, deal with electoral crises and why certain regions stopped their internet voting, while others remain to deploy IVS in their elections, a follow-up study on Estonia's foreign cyber interference, France's discontinuation in 2017 and Norway's case of their technical vulnerabilities may be appropriate. In this proposed study, it would be sensible to look at the positioning of academia and CSOs and the reasons why that may be the case and under what circumstances that might change and impact the adoption and diffusion of internet voting. In summary, internet voting has been around for more than two decades and identified to be a logical tool for democracy and yet lacks large-scale adoption. In this paper I analysed and presented general drivers and barriers that impact the adoption and diffusion process and illustrated further research areas that merit further investigation. Internet voting, being a process in a political process is also highly impacted by political factors itself and therefore significant qualitative differences between the respective drivers and barriers for the respective contexts might exist.

References

- Brown, M., & Hale, K. (2014). *Applied research methods in public and nonprofit organizations*: John Wiley & Sons.
- Flick, U. (2008). *Triangulation - Eine Einführung* (2nd ed.): VS Verlag. .
- Krimmer, R. (2012). The evolution of e-voting: why voting technology is used and how it affects democracy. *Tallinn University of Technology Doctoral Theses Series I: Social Sciences*, 19.
- Licht, N. (2021a). *Insights into Internet Voting: Adoption Stages, Drivers & Barriers, and the Possible Impact of COVID-19*. Tallinn University of Technology, Retrieved from <https://digikogu.taltech.ee/en/Item/bd12d1d6-a1e6-405c-b855-1a7ce4c8a80a>
- Licht, N. (2021b). *Insights into Internet Voting: Adoption Stages, Drivers & Barriers, and the Possible Impact of COVID-19*. Tallinn University of Technology
- Mayring, P. (2014). Qualitative content analysis: theoretical foundation, basic procedures and software solution.
- Ormston, R., Spencer, L., Barnard, M., & Snape, D. (2014). The foundations of qualitative research. *Qualitative research practice: A guide for social science students and researchers*, 2, 52-55.

Technological Change of ONPE, an Electoral Management Body in Peru – Voters and civil servants’ perceptions

Pablo Hartill¹

¹ KU Leuven, Leuven 3001, Belgium

pabloandres.hartillmontalvo@student.kuleuven.be

Abstract. This research focuses on how the administrative capacities of one of the Peruvian electoral bodies are affected by the implementation of technology. In this particular case, on the implementation of the Automated Scrutiny System in the 2021 General Elections.

It explores whether the expectations and/or perceptions of the citizens and especially of the users of this technology-citizen polling station members and technical coordinators hired by ONPE-collide or harmonize with those of the electoral management body.

Finally, based on these expectations and perceptions, it is shown how the capacities of the street-level bureaucrats are also affected by the use of this technology and how the actions of these stakeholders affect the implementation of technology.

Keywords: Administrative capacity, Technological capacity, Street-level bureaucrats.

1 Research design

1.1 Theoretical Framework

In this research, the process of implementation of the Automated Scrutiny System will be analyzed using the framework of Lember, Kattel & Tonurist [1]; this framework evaluates the impact of digital capabilities in the administration of public entities through the analysis of how “routines” -internal and external- and “selection mechanisms”-citizens/users, market-type behavior, networks, hierarchical behavior- affects and are affected by each other.

1.2 Data Collection

As part of the research, citizens who served as polling station members were surveyed and other stakeholders involved in the technology implementation process were interviewed, such as the senior management of the electoral management body, the ICT

area, representatives of other Peruvian electoral bodies, specialists in technology law and officials of the executive branch related to technology and digitization.

The information collected was compared with the data obtained as part of the electoral process in its first and second elections. In this sense, what was obtained was how the perceptions about the implementation of the Automated Scrutiny system corresponded with the expected results in terms of reducing the number of observed tally sheets, easing the workload of polling station members, and reducing the time required to digitize and compute the results of the polling stations

2 Discussion and Results

As an initial conclusion, the perception of the majority of stakeholders towards the use of the Automated Scrutiny System is positive. Even, under certain criteria, it is thought of as a solution that should be replicated and used at the national level. However, the results show that there are a series of external limitations that do not allow this tool to be fully exploited.

Although there are tangible benefits from the use of the automated scrutiny system in certain areas of the country, it falls short of what is expected. Among the limitations found are the poor training -both virtual and on-site- of users, the current electoral regulations governing data computing processes, an extremely manual culture of the electoral process that political organizations understand, the low digital infrastructure of the country, among others.

Furthermore, the implementation of technology in an electoral process must be confronted with the concept of electoral integrity. Transparency, honesty and security of results are the greatest assets of any electoral management body. For ONPE the implementation of the automated scrutiny system mustn't lead to a detriment of its capabilities. This is why the organization must be in constant struggle with the powers of other actors and find a point where it can demonstrate that technology serves the process and does not harm it.

For ONPE, to massify the use of this technology, there is a need of a great process of sensitization of society and political organizations, working with the legislative power and proposing changes that are adapted to the reality of the current electoral processes, seeking to eliminate the black boxes that technology represents and allowing citizen and academic audits of this type of systems. It is with this that ONPE will improve its administrative capacities, standardizing the capacities of those officials who provide the service in an exogenous manner due to their capacities and autonomy in decision making. By making the right changes, it could achieve the objective of providing an effective and efficient service, which improves its image and could be replicated.

References

1. Lember, V., Kattel, R., & Tõnurist, P.: Technological capacity in the public sector: the case of Estonia. *International Review of Administrative Sciences*, 84(2), 214–230. (2018).

Multi-dimensionality of trust towards e-voting

Peeter Leets, University of Tartu
peeter.leets@ut.ee

1 Research Idea

Intention to vote electronically is believed to be dependent upon one's level of trust towards 1) the (state) institutions responsible for organizing the elections; 2) the actual technology used to implement the e-voting system [1,3], both components being about equally important. This distinction has been acknowledged before in e-government trust research but has not found empirical verification. The research idea I am going to propose here was inspired by Carter & Belenger (2005) and Schaupp & Carter (2005) papers that studied which factors influence e-government adoption and one's intention to use e-voting, respectively. Both relied on factor analysis to verify underlying latent structures in their data, as did they include two types of trust in their models — trust of the internet versus trust of state government [1,3]. However, their models also included a number of system-specific factors such as perceived ease of use and system compatibility. This prevented further latent dimensionality among trust variables from emerging because the effect of trust was largely overshadowed by usability-related factors [1,3]. 15 years later, with some vastly enriched data about e-voting at our disposal, it might be fitting to dig deeper and perform a similar analysis on trust-based components only. It has been established that the Estonian e-voting system is already highly compatible and very easy to use, therefore these aspects should not influence one's likelihood to e-vote as much.

2 Research Design

The central framework of analysis is going to be Structural Equation Modeling (SEM) and **Confirmatory Factor Analysis**, a statistical approach that allows hypothesis-testing between observed and latent variables [2]. Factor scores computed from the two latent factors will be set as response variables in two separate OLS models with various socio-demographic factors plus e-voting (dichotomous) as predictors.

Sample data for preliminary analysis was obtained from six cross-sectional surveys about the Estonian e-voting system (2013-2019, spanning 6 elections — 2 local, 2 national, 2 EP). Both voters and non-voters were included in the model, the total number of observations being 3595. 7 trust-related variables were recorded in the survey in 0-10 point Likert scale (treated as interval). Two of them, trust towards 1) internet transactions in general; and 2) electronic voting systems, were set to load on a latent factor that should ideally be interpreted as "trust towards technology". Four others, trust towards the 1) government,

2) parliament, 3) politicians, and 4) parties, were set to load on a factor what may be labelled as "institutional trust". The remaining variable, trust towards the president, was dropped from the analysis as it did not load clearly on either factor and simply hampered factoring reliability.

3 Preliminary Results

The resulting latent factors appeared strongly correlated with each other (cor. coef. 0.53). Generally, positive correlation makes sense — people who trust various state institutions would naturally also trust other things such as e-voting systems. Such a strong correlation, however, indicates that respondents may have subconsciously linked electronic voting with Estonian state institutions when answering that question, despite its corresponding survey question targeting trust towards electronic voting systems in general. It is only natural to make this connection since for the vast majority of respondents, Estonian state-organized elections are probably the only experience they have ever had with e-voting systems. In addition, the technology trust factor appears to be highly dependent upon a single observed variable - trust towards e-voting systems (characterized by low loading and high residual values for the other variable that was set to load on that factor — trust in internet transactions). This is a strong indication that in this model, the so called "technology trust" factor describes trust towards the Estonian e-voting system in particular and may not represent trust in e-voting technology in general. In order to alleviate this issue, one or more technology and/or internet related trust variables (e.g. trust in social media, trust in new technological solutions in general) should load on this latent factor.

Keeping in mind that the aforementioned caveat might have led to some misleading model coefficients, we may still carefully ponder over the most important observed relationships in our models. Unsurprisingly, the most important predictor in determining one's system trust factor score was whether or not they voted electronically. Similarly, being more computer literate, better educated, and being a native Estonian speaker increases one's score on that scale. Interestingly, party of choice (whom voted for) also seems to play a fairly significant role here — those who voted for Centre Party or Conservative People's Party scored notably lower than others, even lower than non-voters. This might suggest that these are the odd ones who do not take issue with responsible state institutions, but are sceptical about the actual e-voting technology itself.

References

1. Carter, L. & F. Belanger. 2005. "The utilization of e-government services: citizen trust, innovation and acceptance factors", *Info Systems J* (15), 5-25.
2. Hoyle, Rick H. 1995. *Structural Equation Modeling. Concepts, Issues, and Applications*. Thousand Oaks, London, New Delhi: SAGE Publications.
3. Schaupp, L. C. & L. Carter. 2005. "E-voting: from apathy to adoption", *Journal of Enterprise Information Management* 18(5), 586-601.

Demo and Poster Session

Towards a quantum resistant i-voting system ^{*}

Jordi Cucurull, Tamara Finogina, Aleix Amill, Noemí Folch, and Nuria Costa

Scytl Election Technologies S.L.,
Travessera de Gràcia 17-21, 7th floor,
08021 Barcelona - Spain,
{name.surname}@scyt1.com

1 Introduction

In recent years, several countries started introducing Internet voting (i-voting) systems for improving their democratic processes. I-voting offers more accurate and fast vote counts, reduces the logistic cost of an election organization, and also allows voters with disabilities to cast their votes independently.

One of the main requirements for i-voting systems is privacy, which states that voters are allowed to cast their vote in conditions of confidentiality (coercion-resistance) and guarantees anonymity of their choices: namely, that it is not possible to link the content of a vote to the identity of the voter. Most of the current i-voting systems ensure privacy by encrypting voters' choices and anonymizing collected ballots via a mixing process that breaks the link between the voter's identity and the cast ballot by applying a random permutation and a re-encryption. However, these algorithms are based on computational problems like factorization and discrete logarithm, which will be easily solved by quantum computers. As a consequence, the current state-of-the-art e-voting systems do not guarantee long-term privacy.

In the EU H2020 Prometheus project we design and implement a proof-of-concept (PoC) of a quantum-resistant i-voting system based on the state-of-the-art post-quantum cryptographic protocols. Our solution focuses on:

- Functional design of the i-voting system. It defines APIs and the detailed implementation notes of the solution using an interface description language.
- Implementation of a cryptographic protocol for proving correct shuffling in zero-knowledge. The mix-net was recently published in [2].
- Vanilla implementation of LPR encryption scheme [5].
- Implementation of the underlying building blocks required for the proof of a shuffle [1][3][4].

2 Implementation

The PoC implements three libraries that are integrated into an i-voting system with backend and frontend implemented in Java and Typescript respectively.

^{*} This work has received funding in the context of the EU H2020 project Prometheus under grant agreement no. 780701. <https://www.h2020prometheus.eu/>

- **Math layer:** implements the basic arithmetic operations over the polynomial ring $\mathbb{Z}_q[X]$, as well as encoding methods.
- **Cryptographic layer:** implements the following post-quantum cryptographic methods required for the vote encryption and mixing at the voting layer:
 - **LPR encryption:** An implementation of the lattice-based encryption scheme [5]. Provides keypairs generation, scheme (encryption and decryption) and encodings from/to byte arrays.
 - **CCM correct Shuffle:** An implementation of the verifiable re-encryption shuffle [2] over LPR ciphertexts.
 - **BKLP commitments:** An implementation of the commitment scheme [1]. Provides key generation, scheme (commit and open), as well as zero-knowledge provers and verifiers for arithmetic circuits.
 - **Preimages knowledge:** Proves/verifies knowledge of preimages under ivOWFs. Includes a dPL module [4], which gives incomplete ZK-proofs, and a CDXY module [3], which is a generic compiler to transform incomplete ZK-proofs into complete preimage knowledge proofs.
- **Voting layer:** implements the following methods, which internally utilize cryptographic layer:
 - **Setup:** Generates parameters and the election encryption keys.
 - **Vote:** Encrypts votes, generates voting receipts and validates the votes.
 - **Mixnet:** Verifiably shuffles and re-encrypts vote and generate zero knowledge proof of mixing correctness.
 - **Count:** Decrypts mixed votes.

Once fully implemented, this will demonstrate the feasibility of i-voting systems with long-term privacy, currently only possible at a theoretical level.

References

1. Fabrice Benhamouda, Stephan Krenn, Vadim Lyubashevsky, and Krzysztof Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In Günther Pernul, Peter Y A Ryan, and Edgar Weippl, editors, *Computer Security – ESORICS 2015*, pages 305–325, Cham, 2015. Springer International Publishing.
2. Nuria Costa, Ramiro Martínez, Paz Morillo, Andrea Bracciali, Jeremy Clark, Federico Pintore, Peter Ronne, and Massimiliano Sala. Lattice-based proof of a shuffle. In *Financial Cryptography and Data Security*, pages 330–346. Springer International Publishing, 2020.
3. Ronald Cramer, Ivan Damgård, Chaoping Xing, and Chen Yuan. Amortized complexity of zero-knowledge proofs revisited: Achieving linear soundness slack. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 479–500, Cham, 2017. Springer International Publishing.
4. Rafael del Pino and Vadim Lyubashevsky. Amortization with fewer equations for proving knowledge of small secrets. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 365–394, Cham, 2017. Springer International Publishing.
5. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Advances in Cryptology – EUROCRYPT 2010*, pages 1–23, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

Microcontroller-Based Voting Client for the Estonian Internet Voting System

Valeh Farzaliyev, Kristjan Krips, Jan Willemsen

September 10, 2021

In this demo, we will be presenting a dedicated voting client for the Estonian Internet voting system [1]. The client uses ESP32 IoT microcontroller platform as its basis, and runs nothing but firmware and a voting application.

The solution also supports the individual verification protocol of the Estonian i-voting system. As the screen of ESP32 is too small to show the QR-code, we have implemented two versions of the solution.

In the basic setting, data needed for verification is transferred from the voting device to the verification device via Bluetooth connection. This setup also requires a modified verification app.

In the extended setup, the client has an external LCD capable of showing the full QR code, so the verification app can be used in the out-of-the-box configuration.

The main rationale behind developing a dedicated hardware platform for voting client is the potential vulnerability of voter's PC environment. Vote manipulation can be detected with a verification app, but it is very hard for the voter to make sure that there is for example no malware on her PC trying to breach vote secrecy.

Of course, building a dedicated client for a special hardware is not a solution for an average voter. However, it provides an extra option for tech-savvy people interested in controlling every aspect of vote casting and willing to take extra actions for such an option.

As a by-product, we have also produced the first completely open source voting client for Estonian Internet voting. We intend to use it ourselves during the upcoming local elections in October 2021.

References

- [1] Valeh Farzaliyev, Kristjan Krips, and Jan Willemsen. Developing a Personal Voting Machine for the Estonian Internet Voting System. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing, SAC '21*, page 1607–1616, New York, NY, USA, 2021. Association for Computing Machinery.

Polys Online Voting System: Lessons Learned from Utilizing Blockchain Technology

Aleksandr Korunov¹, Aleksandr Sazonov¹ and Petr Murzin¹

¹ Kaspersky, Moscow, Russia
{firstname.lastname}@kaspersky.com

Abstract. This short paper provides a brief overview of Polys – a blockchain-based online voting system. We concisely concern blockchain utilization, vote privacy and verifiability, as well as real use cases.

Keywords: E-voting, Blockchain, Polys, Verifiability.

1 Polys overview

Polys is a blockchain-based online voting system. System development started in 2017 and since then we have made a substantial architectural overhaul, namely, migrated from Ethereum to Exonum blockchain framework and upgraded encryption and anonymization schemes.

Although blockchain has recently engendered a fair amount of criticism [1, 2], online voting systems can still benefit from applying blockchain technologies for two main reasons: immutability of data and verification of smart contract execution. Furthermore, blockchain technology itself has never been considered as a bulletproof that would solve all voting-related problems. In fact, it did not even aim to do so, e.g. vote privacy and voter eligibility are usually guaranteed by several additional techniques. We utilize blind signature scheme in order to provide voters' anonymity. Voters' Authentication is possible via predefined codes, SMS, e-mail or OAuth 2.0. In addition, there is a set of services that is called service layer. It is responsible for user authentication, blind message signing and several infrastructural tasks. Interested readers are referred to [3] for detailed description of our platform.

Polys is mainly intended for use in an unsupervised environment meaning that voters vote remotely via the Internet using their devices. Despite the fact that we could probably face an untrusted terminal problem and enhanced threats to coercion which are common not only in Internet voting but in other voting scenarios, e.g. postal voting, we suppose that this voting method is still viable in several cases provided that the security guarantees are implemented thoroughly.

If remote voting is an inappropriate option or higher guarantees for coercion-resistance needed, we provide voting machines for precinct-based elections. These machines work in the same blockchain ecosystem as the internet voting, so these two environments could be used simultaneously or separately.

Polys provides several ballot types: single-choice, multiple-choice, cumulative (with distribution of scores) and Yes No Abstain ballots for different voting scenarios.

Concerning vote privacy, which comprises ballot secrecy, receipt-freeness and coercion-resistance according to recent research, Polys guarantees ballot secrecy, i.e. voters obtain blind signature from ballot issuance service, encrypt the choice and send it directly to the blockchain via anonymous channel. Receipt-freeness and coercion-resistance are much stronger notions of vote privacy and are in hard tension with vote verifiability that we are currently most focused on.

Polys provides E2E verifiability meaning that it could be verified that the vote is cast-as-intended, recorded-as-cast and tallied-as-recorded. After casting a vote, the voter can find his/her transaction in the blockchain and verify how it is tallied after the voting ends and results are decrypted and published. Furthermore, we provide some tooling for vote observation – a piece of software that could access the blockchain auditor node in order to verify transaction signatures and zero-knowledge proofs.

2 Polys use cases

As of February 2021, Polys is extensively used by educational organizations – 62% of all votes conducted on our platform. We also have several cases for nonprofits, political parties, private companies and state authorities. The latter used our platform for participatory budgeting procedures in the Volgograd and Nizhny Novgorod regions of Russia and more than 340 thousands people in total were involved. For further details, please refer to [4].

References

1. Sunoo Park, Michael Specter, Neha Narula, Ronald L Rivest, MIT, Going from bad to worse: from Internet voting to blockchain voting, *Journal of Cybersecurity*, Volume 7, Issue 1, 2021, tyaa025, <https://doi.org/10.1093/cybsec/tyaa025>
2. Bernhard M. et al. (2017) Public Evidence from Secret Ballots. In: Krimmer R., Volkamer M., Braun Binder N., Kersting N., Pereira O., Schürmann C. (eds) *Electronic Voting. E-Vote-ID 2017. Lecture Notes in Computer Science*, vol 10615. Springer, Cham. https://doi.org/10.1007/978-3-319-68687-5_6
3. Polys — Online voting system, Whitepaper Version 2.0. [online] Available: <https://box.kaspersky.com/f/e68a161d8e7241909ea3/>
4. Polys Success Stories, <https://polys.me/success-stories>, last accessed 2021/09/13.

Internet Voting as Part of Mobile Cross-Border Government Services for Europe (mGov4EU)

Robert Krimmer¹ [0000-0002-0873-539X], Lisa Burgstaller², Tina Hühnlein³, Thomas J. Lampoltshammer⁴ [0000-0002-1122-6908], Noemí Folch⁵, Herbert Leitold⁶, Arne Tauber⁷, Detlef Hühnlein⁸ and Carsten Schmidt¹ [0000-0001-8435-4313]

¹ University of Tartu, Johan Skytte Institute for Political Studies, Center for IT Impact Studies, Lossi 36, 51003 Tartu, Estonia
 {firstname.lastname}@ut.ee,

² Technikon Forschungs- und Planungsgesellschaft mbH, Burgplatz 3a
 9500 Villach, Austria, burgstaller@technikon.com

³ go.eIDAS E.V., go.eIDAS e.V., Sudetenstr. 16, 96247 Michelau, Germany,
 tina.huehnlein@ecsec.de

⁴ Danube University Krems, Department for E-Governance and Administration, Dr.-Karl-Dorrek-Straße 30, 3500 Krems, Österreich, thomas.lampoltshammer@donauuni.ac.at,

⁵ Scytl Election Technologies S.L.U., A/ Travessera de Gràcia 17-21,
 08021 Barcelona, Spain, noemi.folch@scytl.com

⁶ Zentrum für Sichere Informationstechnologie – Austria, Seidlgasse 22, 1030 Wien, Austria, herbert.leitold@a-sit.at,

⁷ Technische Universität Graz, Rechbauerstraße 12, 8010 Graz, Austria, arne.tauber@egiz.gv.at

⁸ ECSEC GmbH, Sudetenstrasse 16, 96247 Michelau, Germany, detlef.huehnlein@ecsec.de

Abstract:

With the practical implementation of the “eIDAS regulation”, which has been fully applicable since July 2016, the European Union has made great strides in recent years in successfully simplifying the cross-border online identification process for citizens. In addition, the “Single Digital Gateway (SDG)” regulation for the establishment of a uniform digital access gate for administration in the EU entered into force in December and the transition phase will end in December 2023. A first part of became reality at the end of 2020. After all, there is an unbroken trend towards the mobile and self-determined use of administrative services: Today, citizens expect eGovernment services to always be conveniently usable via smartphone. Against this background, the mGov4EU (“Mobile Cross-Border Government Services for Europe”, <https://mGov4.EU>) project, funded by the European Union as part of the Horizon 2020 research and innovation program, has started to enable mobile cross-border administrative services in Europe. These services will be demonstrated via three different pilots. Internet voting will play an important role, as one of the three pilot use cases.

Keywords: Once-Only Principle, Single Digital Gateway, SDGR, DSM, mGov4EU, eID, eIDAS, mobile, eGovernment, Internet Voting;

1 mGov4EU in a nutshell

The mGov4EU project assembles leading European experts from government, business, and science in Austria, Belgium, Estonia, Germany, and Spain, to enable secure and privacy-friendly mobile government services across Europe. mGov4EU will put the citizen at the center of the considerations and offer them new, secure, fast, and privacy-protecting options for managing their identity and personal data – regardless of whether they or the eGovernment service are located in the home country or another Member State.

The regulatory framework for this project is provided by Regulation (EU) 2018/1724 on the establishment of a Single Digital Gateway (SDGR) and for the cross-border provision of services together with the eIDAS Regulation (EU) 910/2014 related to cross-border electronic identification and trust services for electronic transactions in the internal market. The mGov4EU project puts the requirements of self-sovereign and mobile citizens at the center of the considerations. It interconnects the existing eIDAS ecosystem with the new Single Digital Gateway to create a user-friendly overall system.

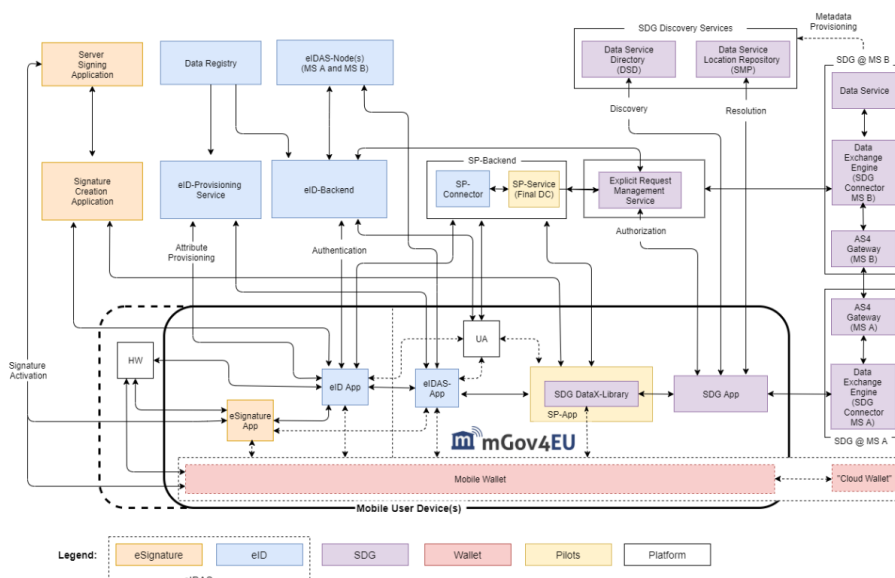


Fig. 1. mGov4EU –Reference Architecture at a Glance

2 Internet Voting as Part of the mGov4EU Piloting Efforts

The project builds upon the existing eIDAS-Layer and combines it with user-centric mobile-based authentication, including a Single Sign-On (SSO) approach. At the same time, a privacy-preserving identity and consent management is established to provide cross-border application scenarios concerning E-Government processes and services. In addition, mGov4EU embraces the SDG-Layer - based on the blueprint developed by the TOOP project (Krimmer et al., 2021) -, striving for a collaborative engagement with provisioning platforms concerning the delivery and re-use of digital services

throughout Europe while holding up the key elements of trust and accessibility.

During the three-year project period, several mGov4EU pilot applications will be designed and implemented to validate the solution, modules and infrastructure services. The pilot applications include internet voting, smart mobility based on subsidized taxi rides, and mobile signature. The aims of the three pilots are demonstrating cross-border mobility, cross-border collaboration, providing additional cross-border information, such as foreign residence and for remote electronic voting settings. The pilots are planned to demonstrate their feasibility under real life conditions and in real life environments like internet voting for the student's council at the University of Tartu.

3 Acknowledgement

This work received funding in the context of the EU H2020 project mGov4EU under grant agreement no. 959072.

References

1. mGov4EU Homepage, <https://www.mgov4.eu>, last accessed 2021/09/08.
2. Krimmer, R., Prentza, A., Mamrot, S., Schmidt, C. & Cepilovs, A. (2021). The Future of the Once-Only Principle in Europe. In R. P. Krimmer, Andriana; Mamrot, Szymon; (Ed.), *The Once- Only Principle* (Vol. LNCS 12621). Springer.

An Overview of the Voatz Election Platform

Nimit Sawhney¹, Simer Sawhney¹, Eric Landquist^{1,2}, and Philip Andrae¹

¹ Voatz, Inc., 50 Milk St. 16th Floor, Boston MA 02109, USA
 {ns, ss, el, pa}@voatz.com

² Department of Mathematics, Kutztown University, Kutztown PA 19530, USA
 elandqui@kutztown.edu

Keywords: Internet Voting · Mobile Voting · End-to-End Verifiable · Accessibility · Absentee Voting · Malware Protection · DDoS Mitigation · Blockchain

1 Introduction

Voatz, Inc. is an internet voting and election management company based in Boston, MA, USA that has been involved in over 80 public and private elections to date. Public elections have included elections in South America, Asia, and in five states in the United States. Private elections have included political conventions, universities, and non-profits. The overall platform is designed as a secure platform to increase access to elections and increase election integrity.

An election that uses the Voatz platform can have up to three means for a voter to access the election. Election access can be either in-person or remote.

- Voatz Mobile App (VMA), available for Android and iOS devices.
- Voatz Event Manager (VEM), available for iOS tablets.
- Voatz Web App (VWA), available for voting via modern internet browsers, only for users who cannot use VMA.

Security features in VMA restrict voting to one person per device. By contrast, VEM and VWA allow multiple voters per device and could be used for controlled environments in which kiosks are allowed to connect to the internet.

One of Voatz' major points of pride is our products' accessibility to voters with various disabilities, including voters with visual impairments, dexterity disabilities, cognitive disabilities, and a lack of mobility. In partnership with the National Center for Accessible Media, Voatz incorporates accessibility and usability design throughout its development process. Accessibility capabilities include VoiceOver and TalkBack screen readers, predictable layout and navigation, configurable font size, voice control (on iOS), speech-to-text (for write-ins), and flexible session timeout limitations. Voatz adheres to the following guidelines.

- The U.S. Elections Assistance Commission (EAC) Voluntary Voting System Guidelines (VVSG v1.1) for usability and accessibility
- Worldwide Web Consortium's Web Content Accessibility Guidelines 2.1 level AA (WCAG 2.1/AA)
- Apple iOS and Android best practices for accessibility

The two most significant concerns of mobile and internet voting are risks due to malware and distributed denial-of-service (DDoS) attacks. Voatz has applied multiple layers of security into its products to defend against these and other attacks on the platform and voters' devices. VMA and VEM check for insecure physical and wireless connections that could compromise device security. Anti-malware software in the app leverages a machine learning algorithm to scan for data anomalies and known malware; such heuristic methods are the only known way to detect previously unknown malware. Voatz utilizes redundancy and physically distributed servers to prevent single-point-of-failure attacks on hardware. Voatz servers, databases, and blockchain nodes are protected by leading network security technology that has the capacity to mitigate DDoS attacks that are 30 times larger than the largest known DDoS attack. This security enables voter privacy and vote verifiability on the platform. To date, there have been no successful attacks on the platform during a live election.

Voter privacy is also ensured by anonymizing ballots. When a voter submits a ballot, a Voatz server generates a unique, random ballot ID, called an *anonID*. The ballot is encrypted and digitally signed by the voter. This authenticates the ballot as legitimate and functions as a digital equivalent of a double envelope for typical remote paper ballots. The digital signature is stripped from the ballot data in a mixnet to complete the anonymization process. When a ballot is recorded on the blockchain and arrives to the election officials, there is no data linking the user to the ballot. The possible exception is when ballot approval is conditional upon election officials accepting an accompanying affidavit.

Finally, the Voatz platform enables end-to-end verification. An election system is end-to-end verifiable (E2E-V) if voters can verify that their votes were (1) cast as intended, (2) recorded as cast, and (3) tallied as recorded. Benaloh describes that one means to establish an E2E-V system is by using "bulletin boards which can be thought of as restricted shared memories" in which "each bulletin board can be read by every process, but it can only be written by its owner, and then only by appending new messages, not by altering old ones." Benaloh noted that at the time, implementing such a bulletin board was a difficult problem. Today, however, we recognize that a system based on distributed ledger technology (e.g., a blockchain) has the precise properties of such a bulletin board. A mere database allows data to be erased. Therefore, a key component of the Voatz platform is in fact a permissioned blockchain in which each cast ballot is immutably recorded on a distributed digital ledger. Any auditor, official, or voter who is given access to an audit system that reads the blockchain will be able to compare each ballot receipt and printable ballot with the blockchain record. In particular, this allows voters to check if their ballots were cast as intended (via the anonID on a ballot receipt), recorded as cast (on the election official's printable ballot records), tallied as recorded (if every ballot is recorded on the blockchain), and reported as tallied, thus establishing the Voatz platform as an E2E-V system.

For more information about Voatz, please visit <https://voatz.com>.