GESELLSCHAFT
FÜR INFORMATIK

Melanie Volkamer, David Duenas-Cid, Peter Rønne,
Jurlind Budurushi, Michelle Blom,
Adrià Rodriguez Pérez, Iuliia Spycher-Krivonosova,
Beata Martin-Rozumilowicz, Oliver Spycher
(eds.)

E-Vote-ID 2024:
Ninth International Joint Conference on Electronic Voting

October 2-4, 2024
Tarragona, Spain

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Proceedings
Series of the Gesellschaft für Informatik (GI)

Volume P-359

ISBN 978-3-88579-803-3
ISSN 1617-5468

Volume Editors

Melanie Volkamer, Karlsruhe Institute of Technology, Karlsruhe, Germany
David Duenas-Cid, Kozminski University, Warsaw, Poland
Peter B. Rønne, University of Luxembourg, Luxembourg
Jurlind Budurushi, Baden-Wuerttemberg Cooperative State University, Karlsruhe
Michelle Blom, University of Melbourne, Australia
Adrià Rodriguez Pérez, Pompeu Fabra University, Spain
Iuliia Spycher-Krivonosova, University of Bern, Bern, Switzerland
Beata Martin-Rozumilowicz, Independent Electoral Expert, United Kingdom
Oliver Spycher, Swiss Federal Chancellery, Switzerland

printed by Köllen Druck+Verlag GmbH, Bonn

Preface

The Ninth International Joint Conference on Electronic Voting, E-Vote-ID 2024, was held during October 2–4, 2024.

E-Vote-ID 2024 took place in Tarragona, in the south of Catalonia, and was hosted by Universitat Rovira i Virgili.

The E-Vote-ID Conference resulted from merging EVOTE and Vote-ID and counting up to 20 years since the first E-Vote conference, in Austria. Since the first conference in 2004, over 1800 experts have attended the venue, including scholars, practitioners, representatives of various 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; it has turned out to be an important global referent concerning this issue.

This year, as in previous editions, the conference consisted of:

- Security, Usability, and Technical Issues Track;
- Governance of E-Voting Track;
- Election and Practical Experiences Track;
- PhD Colloquium;
- Poster and Demo Session.

E-VOTE-ID 2024 received 50 submissions for consideration in the first three tracks (Technical, Governance and Practical Tracks). Each submission was reviewed by 3 to 5 program committee members using a double-blind review process. As a result, 15 papers were selected from the three tracks to be presented in this volume of Lecture Notes in Informatics (i.e. 30% of the submissions). The selected papers cover a wide range of topics connected with electronic voting, including experiences and revisions of the actual uses of E-voting systems and corresponding processes in elections.

We would like to thank the local Prof. Jordi Castellà, Prof. Jordi Barrat, and the Fundació Universitat Rovira i Virgili, for their excellent collaboration in preparing the conference. We would also like to extend our gratitude to the Port de Tarragona for their invaluable support and for allowing the use of their installations for the conference. The gratitude also goes to the KASTEL Security Research Labs for funding these proceedings. Last, but not least, we would like to thank and appreciate the international program members for their hard work in reviewing, discussing, and shepherding papers. They ensured, once again, the excellence of this proceedings with their knowledge and experience.

December 2024

Melanie Volkamer, David Duenas-Cid, Peter Rønne, Jurlind Budurushi, Michelle Blom, Adrià Rodriguez Pérez, Iuliia Spycher-Krivonosova, Beata Martin-Rozumilowicz, Oliver Spycher

Committees

General Chairs:	Melanie Volkamer, Karlsruhe Institute of Technology David Duenas-Cid, Kozminski University Peter B. Rønne, University of Luxembourg
Track Chairs:	Security, Usability, and Technical Issues Jurlind Budurushi, Baden-Wuerttemberg Cooperative State University, Karlsruhe Michelle Blom, University of Melbourne
	Governance Issues: Iuliia Spycher-Krivonosova, University of Bern Adrià Rodriguez Pérez, Universitat Pompeu Fabra
	Election and Practical Experiences Beata Martin-Rozumilowicz, Independent Expert Oliver Spycher, Swiss Federal Chancellery
Poster and Demo Session:	Michael Kirsten, Karlsruhe Institute of Technology
PhD Colloquium:	Alexandre Debant, INRIA, Nancy Cecilia Passanti, Université Paris Cité

Programme Committee

Security, Usability, and Technical Issues

Araujo, Roberto	Universidade Federal do Pará, Brazil
Beckert, Bernhard	Karlsruhe Institute of Technology, Germany
Benaloh, Josh	Microsoft, USA
Bernhard, Matthew	Enhanced Voting, USA
Clark, Jeremy	Concordia Institute for Information Systems Engineering, USA
Collazos, César	Universidad del Cauca, Colombia
Cortier, Veronique	Centre National de la Recherche Scientifique, France
Dragan, Catalin	University of Surrey, England
Essex, Aleksander	University of Western Ontario, Canada
Gaudry, Pierrick	Centre National de la Recherche Scientifique, France
Gibson, J Paul	Mines Telecom, France
Giustolisi, Rosario	IT University of Copenhagen, Denmark
Gjøsteen, Kristian	Norwegian University of Science and Technology, Norway
Gore, Rajeev	The Australian National University, Australia

Grimm, Ruediger	University of Koblenz, Germany
Haenni, Rolf	Bern University of Applied Sciences, Switzerland
Haines, Thomas	The Australian National University, Australia
Jacobs, Bart	Radboud University, The Netherlands
Jamroga, Wojciech	Polish Academy of Sciences, Poland
Kirsten, Michael	Karlsruhe Institute of Technology, Germany
Koenig, Reto	Bern University of Applied Sciences, Switzerland
Kulyk, Oksana	IT University of Copenhagen, Denmark
Küsters, Ralf	University of Stuttgart, Germany
Mayer, Andreas	Hochschule Heilbronn, Germany
Mueller, Johannes	University of Luxembourg, Luxembourg
Neumann, Stephan	Landesbank Saar, Germany
Pereira, Olivier	Université catholique de Louvain, Belgium
Reisert, Pascal	University of Stuttgart, Germany
Renaud, Karen	University of Strathclyde, Scotland
Roseman, Stefan	Federal Office for Information Security, Germany
Ryan, Mark	University of Birmingham, England
Ryan, Peter Y. A.	University of Luxembourg, Luxembourg
Schneider, Steve	University of Surrey, England
Schuermann, Carsten	IT University of Copenhagen, Denmark
Stark, Philip	University of California at Berkeley, USA
Syta, Ewa	Yale University, USA
Teague, Vanessa	Thinking Cybersecurity, Australia
Truderung, Tomasz	Polyas GmbH, Germany
Vukcevic, Damjan	Monash University, Australia
Wen, Roland	The University of New South Wales, Australia
Willemson, Jan	Cybernetica, Estonia
Zagorski, Filip	University of Wroclaw, Poland

Governance Issues

Aranyossy, Marta	Corvinus University of Budapest, Hungary
Barrat i Esteve, Jordi	Election Observation and Democracy Support, Belgium
Darnolf, Staffan	International Foundation for Electoral Systems, USA
Eenmaa, Helen	University of Tartu, Estonia
Fernández Riveira, Rosa Mª	Universidad Complutense de Madrid, Spain
Germann, Micha	University of Bath, England
Goodman, Nicole	Brock University, Canada
Kersting, Norbert	University of Muenster, Germany
Loeber, Leontine	University of East Anglia, England
Montathar, Faraon	Kristianstad University, Sweden
Musial-Karg, Magdalena	Adam Mickiewicz University, Poland
Nemeslaki, Andras	Budapest University of Technology and Economics, Hungary
Nurmi, Hannu	University of Turku, Finland

Pammett, Jon	Carleton University, Canada
Peña-López, Ismael	Universitat Oberta de Catalunya, Catalonia
Plescia, Carolina	University of Vienna, Austria
Sandri, Giulia	European School of Political and Social Sciences, France
Sasvari, Peter	National University of Public Service, Hungary
Serdült, Uwe	Ritsumeikan University, Japan
Smith, Rodney	The University of Sydney, Australia
Solvak, Mikhel	University of Tartu, Estonia
Trumm, Siim	University of Nottingham, England
Vinkel, Priit	E-governance Academy, Estonia
Von Nostitz, Felix-Christopher	Université Catholique de Lille, France

Election and Practical Experiences

Bailey, Matt	International Foundation for Electoral Systems, USA
Bismark, David	Votato, Sweden
Caarls, Susanne	Election Consultant, The Netherlands
Chanussot, Thomas	International Foundation for Electoral Systems, USA
Chaudhary ,Tarun	International Foundation for Electoral Systems, USA
Chelleri, Riccardo	European Commission, Belgium
Driza Maurer, Ardita	Zentrum für Demokratie Aarau/Zurich University, Switzerland
Erni, Barbara	State Chancellery of Thurgau, Switzerland
Giroud, Kayle	Global Cyber Alliance, Belgium
Hofer, Thomas	Objectif Securité, Switzerland
Leclère, Olivier	State of Geneva, Switzerland
Levine, David	German Marshall Fund, USA
Loeber, Leontine	University of East Anglia, England
Macias, Ryan	RSM Election Solutions, USA
McDermott, Ronan	Mcdis, Switzerland
Misev, Vladimir	OSCE/ODIHR, Poland
Past, Liisa	Ministry of Economic Affairs and Communications for Estonia, Estonia
Petrov, Goran	OSCE/ODIHR, Poland
Plante, Stéphanie	University of Ottawa, Canada
Thornton, Laura	German Marshall Fund, USA
Van Kerckhoven, David	Federal Public Service Home Affairs, Belgium
Vollan, Kåre	Quality AS, Norway
Wenda, Gregor	BMI, Austria
Wolf, Peter	IDEA, Sweden
Yard, Michael	International Foundation for Electoral Systems, USA

Inhaltsverzeichnis

Workshops

Track 1: Security, Usability and Technical Issues

Track 2: Governance Issues

Track 3: Election and Practical Experiences

Autorenverzeichnis

Workshops

Track 1: Security, Usability and Technical Issues

Verifying ElectionGuard: a theoretical and empirical analysis

Markus V. G. Jensen[1], Hans-Christian Kjeldsen[1], Andreas S. Nielsen[1], Niklas B. Olesen[1], and Diego F. Aranha[1]

Abstract: End-to-end-verifiable voting systems can only meet their goals if independent auditors are capable of verifying election outcomes as easily and efficiently as possible. ElectionGuard is a prominent effort in this direction, in which simplifying the verification process guides several design decisions, from the choice of cryptographic group to the building blocks for encryption and zero-knowledge proofs. In this paper, we present the development of optimized ElectionGuard verifiers in the Go programming language, targeting versions 1.1 and 2.0 of the specification, focused on analyzability and efficiency. Our verifiers are built on an architecture emphasizing efficiency that exploits parallelism to achieve a running time up to 10 times faster than related work. We also show that version 2.0 of the specification introduces several changes that improve the verification performance by an overall factor of 2, with the decryption process being around 24 times faster. We expect that our design can be used as a reference for future ElectionGuard verifiers.

1 Introduction

Elections form a crucial component of any democratic society, enabling citizens to actively engage in the process of electing their leaders and representatives. Electronic voting comes with the potential to improve the accuracy of elections and increase trust in the process due to the possibility of independent verification. Nevertheless, the widespread adoption of electronic voting has been impeded by concerns with security, transparency, and privacy.

In response to these challenges, Microsoft has designed ElectionGuard [Re24] to provide end-to-end verifiability for electronic elections. An end-to-end-verifiable voting system gives assurance that votes are *cast as intended*, *recorded as cast* and *tallied as recorded*. In particular, ElectionGuard allows voters to confirm that their votes have been accurately recorded by the system and properly included in the final tally along with correct aggregation of ballots. By incorporating asymmetric cryptography and zero-knowledge proofs, ElectionGuard establishes a secure and transparent mechanism for recording and tallying votes, all while safeguarding voter privacy and preserving trust in the electoral system.

Microsoft has recently released ElectionGuard 2.0 [BN18], which replaces the previous 1.1 specification [BN23]. Notable improvements include a significant reduction in proof sizes and the replacement of placeholder selections with range proofs allowing for diverse voting systems like cumulative voting, range voting, STAR-voting, Borda count, among others. The update also features pre-encrypted ballots, instant verification, and refined

[1] Aarhus University, Department of Computer Science, Åbogade 34, 8200 Aarhus, Denmark, 202008836@post.au.dk; 202004086@post.au.dk; 202005664@post.au.dk; 202006523@post.au.dk; dfaranha@cs.au.dk, https://orcid.org/0000-0002-2457-0783

doi:10.18420/e-vote-id2024_01

cryptographic parameters, enhancing efficiency in the construction, storage, and verification of proofs. Lastly, the update also eliminates the need to verify individual guardian data making verification simpler and faster.

Since a full production-ready implementation of ElectionGuard 2.0 is not available yet, the goal of our research is to analyze the potential security and performance improvements of specification 2.0. Therefore our contributions in this paper are the following:

- A brief analysis of the key differences between versions 1.1 and 2.0 of ElectionGuard, with an eye towards verification of results.
- Verifiers in the Go programming language for ElectionGuard 1.1 and 1.91.18; for which the latter is a hybrid specification covering many of the verification steps for ElectionGuard 2.0, but notably without range proofs.
- A sandbox implementation of range proofs to generate mock election records for ElectionGuard 2.0, which has been benchmarked against an equivalent ElectionGuard 1.1 election.
- Theoretical and empirical analysis of the verification time for the versions 1.1 and 2.0, and a performance comparison with related work.

We designed a verifier architecture for efficiency that produces high-speed ElectionGuard verifiers. We observed that our verifiers are up to 10 times faster than related work, and that the election record produced by ElectionGuard 2.0 is around 5 times smaller and 2 times faster to verify than version 1.1.

The remaining sections of our paper are organized as follows. Section 2 introduces ElectionGuard with its cryptographic building blocks, and summarizes the differences between versions 1.0 and 2.0. Section 3 covers the theoretical comparison between ElectionGuard 1.1 and 2.0, focusing on the expected verification time and size of election record. Section 4 covers the implementation of our verifier, including the creation of a Go sandbox environment that enables us to benchmark an ElectionGuard 2.0 verifier. Section 5 covers the empirical comparison between ElectionGuard versions 1.1 and the sandboxed 2.0, when benchmarked with the most recent election conducted with ElectionGuard. Lastly, Section 6 concludes the paper and suggests relevant future work.

2 ElectionGuard

In this section, we give an overview about how ElectionGuard works in its various flavors. We start by summarizing version 1.1 as a foundation to understand the improvements that ElectionGuard 2.0 brings, and additionally discuss some of the key differences in verification in both versions of the specification.

2.1 Overview

There are three main phases of the ElectionGuard process: the pre-election key generation, the ballot encryption, and the post-election decryption of the tally. Ballot encryption and post-election decryption also need to take into account the possibility of ballots being spoiled by the voters to check the correctness of the encryption process. In terms of cryptographic parameters, ElectionGuard is instantiated with a Schnorr group [Sc91] at the 128-bit security level, chosen as a multiplicative subgroup of $\mathbb{Z}_p^*$ and denoted as $\mathbb{Z}_p^r$. Concretely, p is a fixed 4096-bit prime, such that $p - 1 = qr$ is divisible by a 256-bit prime q. A generator g of order q is also fixed in the specification.

Pre-election key generation. The pre-election key generation consists of a public ceremony conducted by a set of n guardians to generate the election public key K. A *guardian* is a trusted entity responsible for generating its own individual key pair and for being available to later join their individual public keys K_i. For a guardian G_i, the key pair is given by $(s_i \in \mathbb{Z}_q, K_i = g^{s_i} \bmod p)$. The joint public key $K = \prod_{i=1}^{n} K_i \bmod p$ is used to encrypt individual ballots under exponential ElGamal [Ad08; El84], an additively homomorphic cryptosystem, such that ballots can be tallied while protecting the privacy of the voters. When generating the joint public key, all guardians also send verifiable Shamir secret sharings of their private keys s_i using polynomials P_i to all other guardians G_j with $j \neq i$, and non-interactive zero-knowledge (NIZK) proofs of knowledge of the corresponding private keys [Sc91]. From the key shares, k guardians can act on behalf of missing guardians, ensuring that the aggregated tallies can always be decrypted by any quorum of k guardians.

Ballot encryption. A ballot consists of at least one, but possibly many contests. A *contest* is a list of *options*, where the voter will give each option either zero or one. The encryption of each option in a ballot is accompanied by a disjunctive proof stating that each value is well-formed (0–1 NIZK proof) [CDS94]. Each contest is also given a selection limit L, which specifies how many options a voter is allowed to select. To prevent a voter from exceeding the limit, a NIZK proof is generated to prove that the number of selections does not exceed L. Voters are also allowed to submit undervotes, where the number of selections is less than L; and null votes, when none of the options are selected. When aggregating the options, placeholders are introduced to prevent differentiation between undervotes and regular votes. For a vote to be valid, the selected options, including the placeholders, must therefore always equal the selection limit.

Spoiled ballots. When a voter has submitted their encrypted ballot, they need to be confident that their selections have been correctly encrypted (*cast-as-intended* property). To accommodate this, ElectionGuard allows voters to either cast or spoil their submitted ballot. If the voter spoils their ballot, they are given the possibility to submit a new ballot, and the spoiled ballot does not influence the election outcome. In practice, if a voter discovers the decryption of their spoiled ballot to be different from their intended vote, some administrative measure needs to take place (e.g. replacing the specific voting machine accountable for the encryption). At the end of the election, spoiled ballots are verifiably decrypted.

Post-election decryption. Once the election is complete, the homomorphic property is used to aggregate all ballots cast, thus creating the encrypted tally. To decrypt a tally or a spoiled ballot, multiple secret keys from the guardians are required. Each guardian will therefore do a partial decryption, which are then used to form the full decryption. In addition to partial decryptions, each guardian also produces proofs that their decryptions are correct using a Chaum-Pedersen proof [CP92]. At the end of the election, each encrypted ballot and all proofs are published to allow voters and independent parties to verify the outcome, satisfying the *recorded as cast* and *tallied as recorded* requirements for verifiability.

Verification of election record. The published election record will be used by an independent verifier to prove the integrity of the election. Currently, the only verifier developed for version 1.91.18 is provided by MITRE, thus this has been chosen to serve as reference for benchmarking our verifier.

2.2 Comparison between versions

Table 1 below emphasizes the most important differences between ElectionGuard 1.1 and 2.0. There are a few other subtle differences, but this provides a general overview of the differences influencing the verifier and its running time. Some of the differences not mentioned in the table are changes to hashing, and pre-encrypted ballots which enable the voter to vote-by-mail or use pre-printed ballots. The latter are not relevant within the scope of this paper as they are not features in ElectionGuard 1.1 and are therefore omitted.

Functionality / Version	**1.1**	**2.0**
Ballot encryption	Exponential Elgamal	Variant for faster NIZK proofs
Ballot integrity	0–1 NIZK proof	Range proofs
Missing guardians	Individual CP proofs	Not necessary
Verifiable decryption	Individual CP proofs	Joint CP proof

Tab. 1: Comparison of ElectionGuard versions, showcasing the main differences in cryptographic building blocks. The abbreviation “CP proof” stands for a Chaum-Pedersen proof of discrete logarithm equality.

Ballot encryption. In ElectionGuard 2.0, the encryption of individual selections in a ballot is defined by

$$Enc_K(m, \epsilon) = (\alpha, \beta) = (g^{\epsilon} \bmod p, K^{\epsilon} K^{m} \bmod p).$$

This is a variant of the exponential form of the ElGamal cryptosystem, where $\epsilon \in \mathbb{Z}_q$ is chosen randomly. In ElectionGuard 1.1, the value β is defined as $\beta = K^{\epsilon} \cdot g^{m} \bmod p$. The new definition does not affect the security of the encryption scheme, but it reduces the cost of computing NIZK proofs.

Ballot integrity. The verifier must detect overvotes in the case that the selections in a contest exceed the selection limit and/or the vote limit. In ElectionGuard 1.1 this is handled by having placeholders and NIZK proofs, such that the aggregation of selections is always equal to the vote limit L. In ElectionGuard 2.0, the voter can place multiple votes on the same candidate. To allow for this, a range-proof is constructed that enables verification that the selection ℓ is within the range $[0, R]$ of the selection limit R. In addition, a voting limit of L must be obeyed for a ballot to be considered valid, which is also a range proof.

The following NIZK range proof is a generalization of the disjunctive 0–1 Chaum-Pedersen proof [BPW12]. Let j denote the values in the range $0 \leq j \leq R$. For each value of j, the prover will compute $R + 1$ commitments of the form $(a_j, b_j) = (g^{u_j} \bmod p, K^{t_j} \bmod p)$, and $t_j = (u_j + (\ell - j)c_j) \bmod q$, for c_j and u_j independently chosen random elements in $\mathbb{Z}_q$. Note that if $\ell = j$ then $t_j = u_j$, which means the prover will not pick c_ℓ, hence ElectionGuard ensures that there is a single commitment which proves knowledge of ϵ such that (α, β) is an encryption of ℓ. The remaining challenge $c_\ell = (c - \sum_{j \neq \ell} c_j) \bmod q$ is defined in terms of a challenge hash $c = H(K, \alpha, \beta, a_0, b_0, \ldots, a_R, b_R)$ along with all c_j for $j \neq \ell$. This results in the following values $v_j = (u_j - c_j \epsilon) \bmod q$ for $0 \leq j \leq R$, since we would like to prove knowledge of ϵ given the commitment nonce u_j and the challenge c_j. This means the verifier receives a list of (v_j, c_j) for all j along with (α, β) where it computes

$$a_j = g^{v_j} \cdot \alpha^{c_j} \bmod p, \qquad b_j = K^{v_j - j \cdot c_j} \cdot \beta^{c_j} \bmod p,$$

and then checks that c is the correct hash and $c = c_0 + c_1 + \ldots + c_R$. The benefit of the range proofs is a significant reduction in the size of the ballots, because the values in the proof are modulo q rather than modulo p.

Missing guardians. ElectionGuard 1.1 and 2.0 handle missing guardians by computing polynomials of degree $k - 1$ for a quorum k of n guardians. It then distributes the shares in encrypted form to the respective guardians which can verify that the share is correct using the commitments to the coefficients in the polynomial. The decryption process in ElectionGuard 2.0 does not consider the number of available guardians, as long as the quorum is meet. The idea is that having n polynomials where $P_i(0) = s_i$, then the sum of these will construct a polynomial P such that $P(0) = s$ where s is the joint secret key. The implication of this is that the 2.0 verifier is drastically simplified since it does not have to handle missing guardians as separate steps. In the proof, the commitments are combined into a single hash which results in a single common challenge value. Each of the individual guardian proofs are combined into a single proof for simpler verification. For ElectionGuard 1.1, each of the partial decryptions are associated with a proof. In case of missing guardians, each reconstruction-share of the partial decryption is also associated with an individual proof, which imposes a large overhead compared to ElectionGuard 2.0 that does not handle missing guardians as separate steps.

Verifiable decryption. Let the aggregated tally in a contest be represented as the products $(A, B) = (\prod_i \alpha_i \bmod p, \prod_i \beta_i \bmod p)$ of all the encryptions for all selections. To prove decryption is computed correctly, the available guardians will produce a Chaum-Pedersen proof for the secret key s such that $M = A^s \bmod p$ and $K = g^s \bmod p$. Each participating guardian G_i, of which there are at least k, starts by committing to the following pair using a random value $u_i \in \mathbb{Z}_q$:

$$(a_i, b_i) = (g^{u_i} \bmod p, A^{u_i} \bmod p).$$

ElectionGuard combines the commitments into a single aggregated commitment $(a, b) = (\prod_{i \in U} g^{u_i} \bmod p, \prod_{i \in U} A^{u_i} \bmod p) = (g^u \bmod p, A^u \bmod p)$. In other words, the joint random value is defined as $u = \sum_{i \in U} u_i \bmod q$, but is computed implicitly in the exponent, and thus never revealed. The guardians construct a NIZK proof with the challenge hash $c = H(K, A, B, a, b, M)$ including a and b, meaning all guardians will have to commit to (a_i, b_i) before the challenge can be computed. This makes perfect sense since it is a joint-proof, and no guardian should be able to proceed before the joint challenge has been computed.

As a result, each guardian G_i will compute a challenge $c_i = (c \cdot w_i) \bmod q$ for their proof, where w_i is the value associated with Shamir's secret sharing, and the value $v_i = (u_i - c_i P(i)) \bmod q$. Each guardian will respond non-interactively to the challenge and verify the responses. If each v_i verifies, then they can be combined into a single response that concludes the shared knowledge of s such that $M = A^s \bmod p$ and $K = g^s \bmod p$. While the guardians check individual proofs, the verifier only checks that the aggregated proof holds using $v = \sum_{i \in U} v_i$.

3 Theoretical comparison

Now that the differences between ElectionGuard 1.1 and 2.0 are established, we can proceed with a comparison of the theoretical verification time and election record size between the two versions. This serves as a theoretical explanation of the empirical results that can be seen later in Section 5.

3.1 Verification time

To theoretically compare the verification time of ElectionGuard 1.1 and 2.0, we analyze the specific steps that have changed the most from 1.1 to 2.0 in terms of *ballot correctness* and *decryption*. Each verification phase, along with the associated individual steps, can be found in Table 2.

In the following, only the steps that dominate the verification time are included; these are exponentiation and multiplication modulo p. This means that the sub-verification steps that include hashing, arithmetic modulo q, etc. have all been removed from the analysis.

Phase / Version	1.1	2.0
Parameter/key validation	1. Parameter validation 2. Guardian public-key validation 3. Election public-key validation	1. Parameter validation 2. Guardian public-key validation 3. Election public-key validation 4. Extended base hash validation
Ballot correctness	4. Correctness of selection encryptions 5. Adherence to vote limits 6. Validation of confirmation codes 7. Correctness of ballot aggregation 16. Correctness of spoiled ballots	5. Well-formedness of selection encryptions 6. Adherence to vote limits 7. Validation of confirmation codes 8. Correctness of ballot aggregation 15–18. Validation of pre-encrypted ballots
Decryption	8. Correctness of partial decryptions 9. Correctness of substitute data for missing guardians 10. Correctness of replacement partial decryptions 11. Validation of decryption of tallies 12–18. Correctness of decryptions for spoiled ballots	9. Correctness of tally decryptions 10. Validation of decryption of tallies 11. Correctness of decryptions of contest data 12–14. Correctness of decryptions for challenged ballots

Tab. 2: Aggregation of verification steps in phases for specifications 1.1 and 2.0.

We will let $c \in C$ denote a particular contest within the set of all contests C where $|C| = 1$ for simplicity. Additionally, s_c denotes the number of selections, R_c is the selection limit, and L_c is the voting limit for a particular contest. Also, n denotes the total number of guardians, k is the quorum needed to decrypt, and mg is the number of missing guardians. Lastly, b denotes the amount of cast ballots, and b_s is the number of spoiled ballots. Note that all the modular exponentiations are of an element in $\mathbb{Z}_p^r$ to an integer exponent in $\mathbb{Z}_q$. Lastly, we will use the fact that exponentiation modulo p is approximately as expensive as $\log q$ multiplications modulo p, since the exponent is of size $\log q$.

Ballot correctness. To argue for the running time of the verification steps associated with ballot correctness, we define a function $g_x(s_c, R_c, L_c)$ that expresses the number of multiplications modulo p to verify a single ballot with one contest where x denotes the ElectionGuard version. Note that multiple steps have been aggregated as argued in Table 2:

$$g_{1.1}(\ldots) = b \cdot (9 \cdot \log q \cdot s_c \cdot R_c + 9 \cdot \log q \cdot L_c + 5 \cdot \log q)$$
$$g_{2.0}(\ldots) = b \cdot (4 \cdot \log q \cdot s_c \cdot R_c + 4 \cdot \log q \cdot L_c + 4 \cdot \log q + 4 \cdot \log q \cdot s_c)$$

When looking at the asymptotic running times of these functions we see that they are equal, namely $O(b \cdot s_c \cdot R_c + b \cdot L_c)$. However, the coefficients for ElectionGuard 2.0 are around two times smaller compared to ElectionGuard 1.1 indicating that verification of *ballot correctness* for ElectionGuard 2.0 should be approximately twice as fast as s_c, R_c, and L_c increases.

Verifiable decryption. The verification steps that concern decryption for ElectionGuard 1.1 and 2.0 can be seen in Table 2. Again we define a function $f_x(s_c, R_c, L_c, n, k, mg, b_s)$ that expresses the number of multiplications modulo p to theoretically compare the running time of verifiable decryption. Note that to verify a spoiled ballot with the same contest one would need just as many multiplications modulo p along with a small constant to also verify the placeholder selections.

$$\begin{aligned} f_{1.1}(\dots) = {} & \log q \cdot s_c \cdot R_c \cdot (mg \cdot k^2 + b_s + 1) + 4 \cdot \log q \cdot s_c \cdot R_c \cdot n \cdot (b_s + 1) + \\ & 5 \cdot \log q \cdot s_c \cdot R_c \cdot mg \cdot k + 5 \cdot \log q \cdot mg \cdot b_s \cdot k \cdot (s_c \cdot R_c + 1) + \\ & \log q \cdot mg \cdot b_s \cdot k^2 \cdot (s_c \cdot R_c + 1) + 4 \cdot \log q \cdot b_s + L_c \cdot b_s \cdot (\log q + n) \\ f_{2.0}(\dots) = {} & b_s \cdot (5 \cdot \log q \cdot s_c + 4 \cdot \log q) + 5 \cdot \log q \cdot s_c + 4 \cdot \log q \end{aligned}$$

When looking at the asymptotic running times of these functions we see that ElectionGuard 1.1 has a running time of $O(s_c \cdot R_c \cdot mg \cdot b_s \cdot k^2)$ and ElectionGuard 2.0 has a running time of $O(s_c \cdot b_s)$. This shows that asymptotically ElectionGuard 2.0 scales better, since the complexity depends on fewer terms.

In summary, the difference in the amount of time it takes to verify ballot correctness and decryption is significant. Concretely, the selection limit drastically increases the verification time for 1.1 compared to 2.0, which is evident from the partial derivatives of the functions with respect to R_c.

3.2 Size of election record

To reason about the size of an election record, we have p and q being the two primes used. Here p is the large prime with a size of 4096 bits, q is the small prime (subgroup order) with a size of 256 bits, and let $|p|$ and $|q|$ denote their sizes. For the sake of simplicity, we assume that we have no missing guardians, which means that we have n guardians present during the decryption ceremony.

Step / Version	1.1	2.0
Ballot selections	$b \cdot 2 \cdot \lvert p \rvert \cdot \sum_{c \in C} (s_c \cdot R_c + L_c)$	$b \cdot 2 \cdot \lvert p \rvert \cdot \sum_{c \in C} s_c$
Selection limit	$b \cdot (4 \cdot \lvert p \rvert + 4 \cdot \lvert q \rvert) \cdot \sum_{c \in C} (s_c \cdot R_c + L_c)$	$b \cdot 2 \cdot \lvert q \rvert \cdot \sum_{c \in C} s_c \cdot R_c$
Vote limit	$b \cdot \lvert C \rvert \cdot (2 \cdot \lvert p \rvert + 2 \cdot \lvert q \rvert)$	$b \cdot 2 \cdot \lvert q \rvert \cdot \sum_{c \in C} L_c$
Tally	$2 \cdot \lvert p \rvert \cdot \sum_{c \in C} s_c \cdot R_c$	$2 \cdot \lvert p \rvert \cdot \sum_{c \in C} s_c$
Decryption proof	$n \cdot (2 \cdot \lvert p \rvert + 2 \cdot \lvert q \rvert) \cdot \sum_{c \in C} s_c \cdot R_c$	$2 \cdot \lvert q \rvert \sum_{c \in C} s_c$

Tab. 3: Theoretical comparison of election record size.

Each entry in the table represents the number of bits stored in the election record concerning a particular verification step. It is important to note that the information stated in Table 3 does not cover the full election record, however, the table emphasizes some of the relevant differences between 1.1 and 2.0.

After comparing the size of each step, it is easy to argue for most steps that 2.0 is expected to have a smaller election record, even if we favor ElectionGuard 1.1 with the best possible (but meaningful) parameters, such as setting $R_c = L_c = 1$. Additionally, we know that $p > q$. The analysis for the vote limit data is slightly more involved and we have to consider the two expressions inside the sum:

$$b \cdot |C| \cdot (2 \cdot |p| + 2 \cdot |q|) \geq b \cdot 2 \cdot |q| \cdot \sum_{c \in C} L_c \iff |C| \cdot (|p| + |q|) \geq |q| \cdot \sum_{c \in C} L_c$$

$$\iff |C| \cdot \left(\frac{|p|}{|q|} + 1 \right) \geq \sum_{c \in C} L_c \iff \sum_{c \in C} \left(\frac{|p|}{|q|} + 1 \right) \geq \sum_{c \in C} L_c.$$

In the 2.0 specification, we have that $\frac{|p|}{|q|} + 1 = 17$, which means that for $L_c \leq 17$, vote limit data in ElectionGuard 2.0 will be smaller compared to 1.1.

3.3 Discussion

It is clear that steps related to ballot correctness and verifiable decryption are faster as a function of the selection limit R_c and the voting limit L_c. Therefore we expect the verifier to be faster since the function expresses the amount of multiplications performed modulo p. More concretely we expect the verification of ballot correctness to be approximately two times faster in ElectionGuard 2.0 compared to 1.1 when we disregard non-dominant operations such as arithmetic modulo q and hash functions. In terms of verifiable decryption, it is also evident from $f_{1.1}(s_c, R_c, L_c, n, k, mg, b_s)$ and $f_{2.0}(s_c, R_c, L_c, n, k, mg, b_s)$ that 2.0 requires significantly fewer multiplications compared to 1.1 even in the case of $R_c = 1$.

This aligns with the empirical data, which suggests ballot correctness in version 2.0 to be 1.5 times faster and verifiable decryption to be 24 times faster, which is explained in detail in Section 5.2. Additionally, the results show that the number of missing guardians plays a major role for verifiable decryption in ElectionGuard 1.1 which is due to the proofs associated with the replacement of missing partial decryptions. These proofs are not present in ElectionGuard 2.0 since the reconstruction of missing partial decryptions is handled implicitly when decrypting. This shows how a change in the specification can lead to an improvement in verification time without compromising security. The theoretical analysis of the size of the election record showed that the election record is expected to be smaller in ElectionGuard 2.0 compared to ElectionGuard 1.1, which means that deserialization is expected to be faster. Empirically, 2.0 is shown to be a factor of 4.3 faster and is covered in detail in Section 5.2.

4 Implementation details

To make a fair comparison of the two verifiers for version 1.1 to 2.0, it is important to look at their structure. Here both our verifiers follow the same structure; the election record is deserialized into a Go struct, verified, and an accurate output is produced. The verification is divided into steps according to the specification which makes it easier for the reader to follow and confirm. The modular arithmetic is implemented in the same way across the two versions and the hashing convention is implemented according to the specification. The two projects both have a single-threaded and multi-threaded strategy such that the running time can be reduced by utilizing multiple cores in the CPU. This includes running the steps in parallel, but also dividing the computationally-heavy steps into multiple threads for better CPU utilization. Note that this strategy is the same across the two verifiers, thus the comparison is fair.

4.1 Dealing with incomplete data

To benchmark an ElectionGuard 1.1 verifier against a 2.0 verifier, it is important to have an election record with the same number of ballots, spoiled ballots, contests, and a corresponding number of selections. However, a complete election record for ElectionGuard version 2.0 does not yet exist as ElectionGuard 2.0 is still under development. Hence, we will describe how we have generated an election record for the 2.0 specification before the release. For this purpose, we have used ElectionGuard 1.91.18 used in the College Park Maryland election. This version is a hybrid implementation between ElectionGuard versions 1.1, 1.53, and 2.0, hence the hybrid election record can serve as a good basis to build an election record for ElectionGuard 2.0. However, some data needs to be constructed to achieve a fair comparison between versions 1.1 and 2.0. Note that neither of the individual specifications can be used to verify the College Park Election, however, MITRE, along with Microsoft researchers/engineers, have created a specification[2] for the hybrid implementation. The MITRE specification serves as the foundation of understanding the key differences between 1.91.18 and 2.0.

ElectionGuard 1.91.18 does not yet implement the full ElectionGuard 2.0 specification. The main gap impacting verification performance are the range proofs for ballot integrity and adherence to vote limits, which should produce a significant decrease in the running time of ElectionGuard 2.0 compared to ElectionGuard 1.1. To deal with the missing range proofs, we have implemented a sandbox environment[3] which can construct artificial ballots, with range proofs as specified in ElectionGuard version 2.0. The constructed range proofs will for obvious reasons not verify in the context of the College Park Election. Still, it provides a fair estimation of the running time of the verifier.

[2] https://www.electionguard.vote/elections/College_Park_Maryland_2023/

[3] https://github.com/AU-HC/electionguard-sandbox-go

Therefore, using this sandbox environment, it is possible to augment a 1.91.18 election record to fit the 2.0 specification which has the same selection limit, voting limit, and the same amount of cast ballots. This ensures that the artificial ballots can be used in verification steps 5 and 6 from the 2.0 specification [BN23, p. 69-70]. The benchmarking results and arguments for why the comparison is fair can be found in Section 5.1.

4.2 Sandbox environment

This sandbox environment was also written in Go, and includes an implementation of the exponential ElGamal as explained in Section 2.2, along with the hash function for the ElectionGuard 2.0 specification. The sandbox for generating artificial ballots relies on a set of input parameters to tailor the generation process. The two key input parameters are:

1. **Number of ballots:** Integer value representing the desired amount of artificial ballots to be generated.
2. **Manifest:** Path to a `json` file. This file contains a representation of the contests in the election along with the voting limit, selection limit, and selections for each of the particular contests.

Once the `manifest.json` file has been loaded into Go, it iterates over each one of the contests. For each of these contests, it will then pick a random vote for all selections and make sure that they adhere to the selection limit and voting limit. At this point, the nonces will be randomly generated and the selections will be encrypted under that nonce. Using the ElectionGuard 2.0 specification, it then generates range proofs for both the selection limits and voting limit.

The output of our sandboxing generates a list of ballots that steps 5 and 6 in ElectionGuard 2.0 can use to validate the well-formedness of selection encryptions and the adherence to vote limits respectively [BN23, p. 32, 34].

5 Empirical comparison

This section will shortly introduce our ElectionGuard verifiers, and cover our methodology to benchmark ElectionGuard 1.1 against ElectionGuard 2.0. Lastly, we will present and analyze our benchmarking results.

We discuss three verifiers: the first targeting version 1.1[4], another targeting version 1.91.18[5] for the College Park Election; and a last one targeting 2.0[6],which uses the mock range proofs. The verifiers have been implemented according to the verification steps in the

[4] `https://github.com/AU-HC/electionguard-verifier-go/tree/main/version/1.1`

[5] `https://github.com/AU-HC/electionguard-verifier-go`

[6] `https://github.com/AU-HC/electionguard-verifier-go/tree/feature/sandbox`

MITRE Requirements document [LW23]. We prioritize *analyzability* [Ch20, p. 49-50] in our implementations. Therefore in our code bases, the files follow a structure such that it is easy to relate code to verification steps improving the ease for a third party to ensure that the verifier behaves correctly.

5.1 Methodology

The benchmarking will be executed over the election record of the College Park Election as a skeleton for an election record for specification 2.0. The College Park Election record has 1341 ballots of which 5 are spoiled, using a quorum of $k = 3$ out of $n = 5$ guardians. Using our sandbox environment, we generated 1341 ballots with range proofs as close to the College Park Election as possible. Therefore, we have the same number of ballots, spoiled ballots, with the same number of contests, and selections for each contest. In the College Park Election, there were six contests and each contest had a selection limit of one. These artificial ballots are used to replace the data used in verification steps 5 and 6, hence providing a fair estimation of the running time of an ElectionGuard 2.0 verifier. Because of the added ballots from our sandboxing, we have not incorporated deserialization of the election record in our benchmarking results of the verifiers. Thus this will be benchmarked separately in Section 5.2.

To ensure that we have an election record of a comparable size for ElectionGuard 1.1, we used the sample generator Python script[7] developed by Microsoft. Hence, we have created a corresponding election record with 1341 ballots and five spoiled ballots. It is important to note that the 1.1 specification uses placeholder selections for each contest to ensure that the vote limit is always met. To generate a 1.1 equivalent election record for the College Park Election, there have to be L placeholder selections for a vote limit of L in each contest.

Furthermore, we will also be verifying subsets of the total 1341 ballots to see how the verification time scales as a function of the number of ballots. This gives us a better understanding of how the verification time scales for the two specifications. This was done by simply removing ballots from the election record, which means that some checks in regards to aggregated tallies will fail to verify. Note that this does not compromise fairness, since it does not influence the verification time.

The verifiers were executed on two machines running Go version 1.19. The first machine is an Intel Core i9 10850k with 32 GB DDR4 and capable of reading 7,450 MB/s from disk. The second machine is an ARM M1 Air with 8GB of LPDDR4x RAM and capable of reading 2,830 MB/s from disk.

[7] https://github.com/microsoft/electionguard-python/blob/main/src/electionguard_tools/scripts/sample_generator.py

5.2 Running time comparison

This section compares the verification times of ElectionGuard versions 1.1 and 2.0 using the two verifiers across different numbers of ballots. This includes a comparison of the individual verification steps' running time, an aggregated comparison, a deserialization comparison, and a comparison of our verifiers with the MITRE verifiers.

Comparison of verification time for 1.1 **and** 2.0**.** To evaluate the performance of specification 2.0 against specification 1.1, we ran the two verifiers on election records containing the following number of ballots: 25, 50, 100, 250, 500, 750, 1000, and 1341 to give us a better understanding of how the number of ballots affects the verification time. Also, each election record was verified ten times to adjust for variance. As seen in Figure 1, the verification time of specification 2.0 is much faster, generally by a factor of two. However, for small elections with 25 ballots, this speedup is around a factor of five, and for 50 ballots, the speedup is about a factor of three. In order to explore and better understand why this speedup is observed, we have done an experiment benchmarking the running time of the steps in an aggregated format.

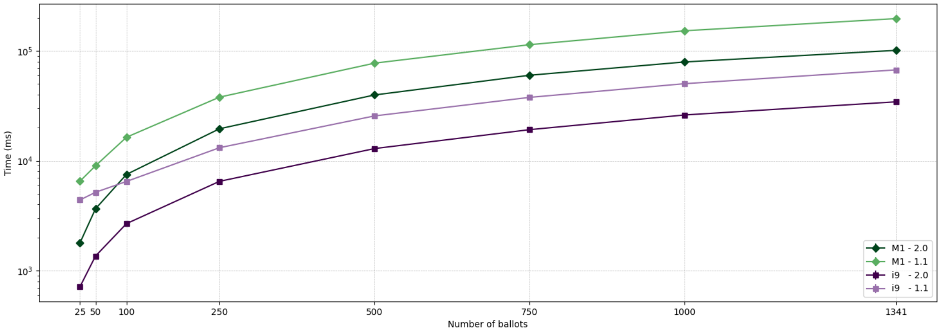


Fig. 1: Comparison of verification for 1.1 and 2.0 with i9-10850k and M1.

Aggregated steps comparison for 1.1 **and** 2.0**.** We verified 1.1 and 2.0 ten times with the i9 10850k processor and calculated the mean verification time for the total 1341 ballots. As seen in Figure 2, the steps regarding decryption are about a factor of 24 times faster in 2.0 compared to 1.1. Ballot correctness verification is 1.5 times faster in ElectionGuard 2.0 compared to 1.1. Both speedups match the theoretical comparison from Section 3.

Deserialization comparison for 1.1 **and** 2.0**.** The previous benchmarks did not take into account the deserialization of the election record, therefore we have also chosen to benchmark the time it takes to deserialize a 1.1 and 2.0 election record. As seen in Figure 3, deserialization time is a factor of 4.3 faster for 2.0 compared to 1.1. This speedup mainly stems from the fact that the ballot sizes are much smaller, due to the range proofs and the changes to verifiable decryption.

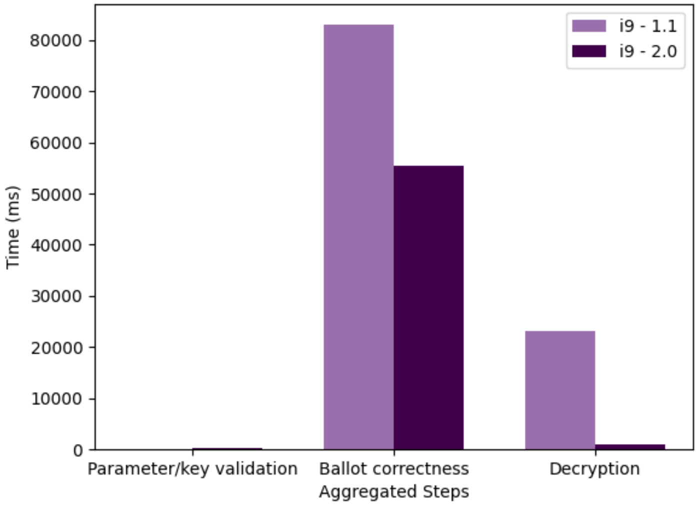


Fig. 2: Comparison of aggregated steps for 1.1 and 2.0

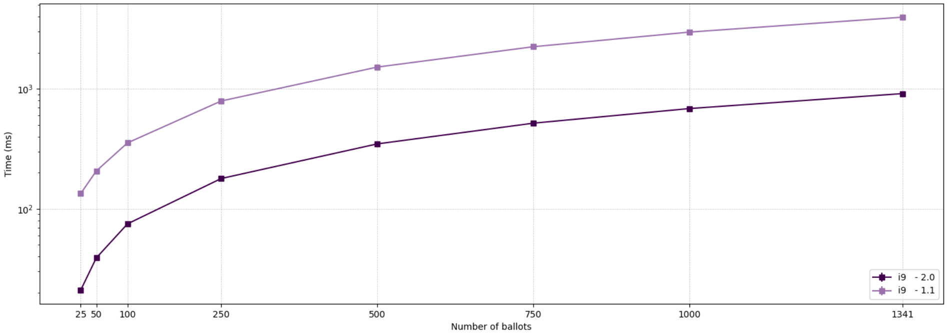


Fig. 3: Comparison for deserialization for 1.1 and 2.0 with i9-10850k.

Comparison with MITRE verifier for 1.1 **and** 1.91.18**.** We have also verified our Go verifier for 1.1 and 1.91.18 against MITRE's implementations in Julia[8] where both implementations are multithreaded. We did this for ten iterations and by only verifying the complete election record for the College Park Election containing 1341 ballots. As seen in Table 4, the i9 processor is notably faster by a factor of 6.7 in mean running time when running Go compared to Julia for version 1.91.18. Similarly, when considering the M1 processor, a significant contrast is observed between the mean running times of 30.18 and 142.70 seconds of the Go and Julia-based verifiers, respectively. This trend can also be observed for version 1.1, where the Go verifier is consistently faster for both processors.

[8] 1.1: `https://github.com/mitre/ElectionGuardVerifier.jl`
1.91.18: `https://github.com/mitre/ElectionGuardVerifier1X.jl`

	i9-Ours	i9-MITRE	M1-Ours	M1-MITRE
Average time for 1.1 (s)	67.21	682.43	196.97	617.40
Average time for 1.91.18 (s)	11.13	74.05	30.18	142.70

Tab. 4: Verification times for the 1.1 and 1.91.18 MITRE- and our Go-verifier using the full College Park election with 1341 ballots.

In summary, the results show that the i9 processor benefits from having 20 cores compared to the 8 cores on the M1. This difference is amplified when running the Go verifier, which seems to utilize the available cores to their full extent. Both the i9 and M1 processors demonstrate considerably better performance.

5.3 Election record size comparison

To fairly compare the election record size in ElectionGuard 1.1 and 2.0, we use the equations in Section 3.2 expressing the size of the election record as a function of b, $|p|$, $|q|$, s_c, R_c, and L_c. The comparison uses the election parameters from the College Park Election as explained in Section 5.1.

	ElectionGuard 1.1	ElectionGuard 2.0
Size of all proofs (bytes)	116, 131, 008	22, 759, 680

Tab. 5: Size of proofs for ElectionGuard 1.1 and 2.0 for the College Park Election.

As seen in Table 5, the size for all proofs of the election record in ElectionGuard 2.0 is a factor of 5.1 smaller compared to ElectionGuard 1.1. This corresponds to an 80% decrease which conforms well with the statement from Microsoft, which states that the size of the proofs in the election record has been reduced by more than 90% [Re24]. It is important to note that the 80% calculated was for the best possible case in ElectionGuard 1.1.

5.4 Discussion

The benchmarking results, conducted on two computers running Go, revealed substantial efficiency gains in the sandboxed ElectionGuard 2.0 compared to 1.1. In smaller elections, we saw a decrease in the running time by a factor of five and three for 25 and 50 ballots, respectively. In general, we saw at least a decrease by a factor of two. We chose to aggregate related steps, which highlighted an efficiency improvement in decryption verification by a factor of 24 when comparing ElectionGuard 2.0 to 1.1. The steps related to deserialization showed an efficiency improvement by a factor of 4.3 from 1.1 to the sandboxed 2.0, which is compatible with the estimated decrease of the election record by 5.1 times.

Lastly, the comparison between MITRE's Julia-based verifier and our Go verifier, on both i9 and M1 processors, emphasized the difference in verification time concerning both the choice of verifier and processor. This showed that our Go-based verifier was faster compared to the Julia verifier by a factor of 6.7 and 4.7 for the i9 10850k and Macbook Air M1, respectively. In summary, the updated verifier, along with empirical benchmarks, showcases the efficiency gains, we can expect from ElectionGuard 2.0 compared to 1.1, particularly concerning steps related to decryption and deserialization.

6 Conclusion

In this paper, we have presented the design and implementation of ElectionGuard verifiers targeting various versions of the specification. We have developed an independent verifier for ElectionGuard 1.1 that is up to 10 times faster than the MITRE verifier. This served as basis for implementing a verifier for ElectionGuard 1.91.18 that was able to verify the College Park Election from November 2023 around seven times faster than MITRE's.

Our verifier was then modified to support ElectionGuard 2.0 elections such that we could use artificial ballots and estimate how the running times between ElectionGuard 1.1 and 2.0 compare. These benchmarks show approximately the same results as expected by the theoretical analysis of the ballots' size and an estimate of the amount of multiplications modulo p. In the empirical results regarding ballot correctness, it is evident that ElectionGuard 2.0 is about 1.5 times faster compared to ElectionGuard 1.1, which approximates the 2-factor estimate from the theoretical analysis. The difference between theoretical and empirical results might be explained by hashing and arithmetic modulo q. Regarding verifiable decryption, it is clear that ElectionGuard 2.0 performs better than 1.1 in terms of verification time, with the asymptotic complexity from the theoretical analysis reducing from $O(s_c \cdot R_c \cdot mg \cdot sb \cdot k^2)$ to just $O(s_c \cdot sb)$. The empirical benchmarks also support that decryption is about 24 times faster for 2.0 to 1.1.

Moreover, we benchmarked the size of the election record both theoretically and empirically. The theoretical results show that the election record is smaller for ElectionGuard 2.0 compared to 1.1 when considering the size of the proofs for ballot selections, selection limit, vote limit, tally, and verifiable decryption. Empirically, the results show that deserialization is about 4.3 times faster comparing ElectionGuard 2.0 to ElectionGuard 1.1.

Future work. Future work could include comparing our sandboxed ElectionGuard 2.0 implementation with the real 2.0 implementation to determine the accuracy of the benchmarking we have carried out. In this report, we have used the College Park Election, which has a selection limit of $R = 1$. Hence, it would be interesting to see how varying the selection limit affects the running time compared to a corresponding ElectionGuard 1.1 election record.

ElectionGuard 2.0 introduces pre-encrypted ballots as a different way of voting. This allows for example voting methods like vote-by-mail which ease availability. In this report, we have omitted the analysis of pre-encrypted ballots as it is a 2.0 feature. Therefore, it would be interesting to see how pre-encrypted ballots work in practice and look at the cryptographic differences between pre-encrypted ballots and actual ballots for ElectionGuard 2.0.

References

[Ad08] Adida, B.: Helios: Web-based Open-Audit Voting. In: USENIX Security Symposium. USENIX Association, pp. 335–348, 2008.

[BN18] Benaloh, J.; Naehrig, M.: ElectionGuard Specification v1.1, 2018, URL: http://www.electionguard.vote/spec/, visited on: 02/01/2023.

[BN23] Benaloh, J.; Naehrig, M.: ElectionGuard Specification v2, 2023, URL: http://www.electionguard.vote/spec/, visited on: 01/30/2024.

[BPW12] Bernhard, D.; Pereira, O.; Warinschi, B.: How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios. In: ASIACRYPT. Vol. 7658. Lecture Notes in Computer Science, Springer, pp. 626–643, 2012.

[CDS94] Cramer, R.; Damgård, I.; Schoenmakers, B.: Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In: CRYPTO. Vol. 839. Lecture Notes in Computer Science, Springer, pp. 174–187, 1994.

[Ch20] Christensen, H.: Flexible, Reliable, Distributed Software 2nd Edition: Still using Patterns and Agile Development. LeanPub.com, 2020.

[CP92] Chaum, D.; Pedersen, T. P.: Wallet Databases with Observers. In: CRYPTO. Vol. 740. Lecture Notes in Computer Science, Springer, pp. 89–105, 1992.

[El84] ElGamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In: CRYPTO. Vol. 196. Lecture Notes in Computer Science, Springer, pp. 10–18, 1984.

[LW23] Liskov, M.; Wilhelm, M.: Requirements for ElectionGuard version 1.91 Verifiers, 2023, URL: https://www.electionguard.vote/images/MITRE-EG-CP-requirements.pdf, visited on: 01/30/2024.

[Re24] Research, M.: ElectionGuard, 2024, URL: https://www.electionguard.vote, visited on: 02/05/2024.

[Sc91] Schnorr, C.: Efficient Signature Generation by Smart Cards. J. Cryptol. 4 (3), pp. 161–174, 1991.

Recommendations for Implementing Independent Individual Verifiability in Internet Voting

Florian Moser[1], Rüdiger Grimm[2], Tobias Hilt[3], Michael Kirsten[3],
Christoph Niederbudde[3], and Melanie Volkamer[3]

Abstract: End-to-end verifiable systems are employed to safeguard the integrity of Internet voting. Voter-initiated verification for *individual verifiability* require that the ballot formed on the voter's device is audited on a second device, which is independent of a potentially manipulated voter's device. Further trust is gained by executing the verification procedure on a second device with independent implementations, in order to defend against a dishonest primary system operator. This paper formulates recommendations to implement such independent individual verifiability tools. Our recommendations are based on the experiences made in the GI elections 2023 where such independent tools were made available to the voters – to our knowledge the first project of its kind.

Keywords: Internet Voting, Individual Verifiability, Second Device

1 Introduction

Internet voting promises to improve access to elections, while reducing administrative effort, but also poses new challenges regarding security and trustworthiness of elections. To safeguard election integrity, end-to-end verifiable voting systems provide artifacts to independent parties to verify that the election proceeded correctly under given trust assumptions. End-to-end verifiability can be classified into *universal verifiability*, where anyone may verify, e. g., whether the votes are tallied as recorded, and *individual verifiability*, where voters verify individually, e. g., whether their vote is cast as intended. These checks must preserve the secrecy of the election and be performed under realistic trust assumptions.

Many approaches for individual verifiability perform an audit over the formed ballot. These approaches can be classified into *audit-or-cast* (also called the Benaloh challenge [Be87]), where the voter chooses to either audit or cast their ballot, and *cast-and-audit*, where the voter audits the actually cast ballot. It is crucial that the audit respects the given trust assumptions. To avoid full trust in the device which forms the ballot, the audit needs to be executed on a second, separate device. Further, to avoid full trust in the main system provider, the code running on the second device needs to be written independently and provided to the second device untampered (e. g., on servers operated by independent parties).

[1] INRIA Nancy, France, florian.moser@inria.fr, https://orcid.org/0000-0003-2268-2367
[2] Fraunhofer SIT, Darmstadt, Germany, grimm@uni-koblenz.de, https://orcid.org/0009-0005-5481-8419
[3] KASTEL Security Research Labs, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany,
tobias.hilt@kit.edu, https://orcid.org/0000-0001-9267-5109;
kirsten@kit.edu, https://orcid.org/0000-0001-9816-1504; udqps@student.kit.edu;
melanie.volkamer@kit.edu, https://orcid.org/0000-0003-2674-4043

doi:10.18420/e-vote-id2024_02

In this work, we formulate recommendations to implement such independent individual verifiability tools. These recommendations allow building upon the experiences we made, to continue what worked well, and at the same time to avoid potential pitfalls.

We base our recommendations on the experiences made developing and providing such tools for the German Informatics Society (GI) elections of 2023 using the Polyas system. In this election, three independently developed and hosted tools[4] allowed voters to perform individual verifiability checks on their second device. To the best of our knowledge, this is the first project of this kind, namely that the election organizer provided independently implemented individual verifiability tools to their voters as part of end-to-end verifiable Internet voting.

Related Work. Cortier et al. [Co23] report on a similar verifiability project for voting from abroad in the 2022 legislative elections in France, yet with a much more limited notion of verifiability and overall no end-to-end verifiability. Another related endeavor is Microsoft's open community project ElectionGuard [Be24], which – while mainly targeting in-place elections – also provides end-to-end verifiable voting and accompanied multiple pilot elections, e. g., the College Park 2023 General Election[5] with independent individual verifiability tools. Further note that, while not specifically reporting on similar projects, the Swiss Post System [Sw24], CHVote [Ha23] and Belenios [CGG19] are also extensively publicly documented end-to-end-verifiable voting systems, which allow implementing independent verifiability tools.

Moreover, we are aware of two other works that formulate recommendations within the development of Internet voting applications, which however have different foci. Hänni et al. [Ha20] give recommendations from the point of view of system developers, while Haines and Rønne [HR21] take the perspective of source code examiners. While these perspectives are quite different from ours (we neither design nor review the main system), some of our observations overlap with their reports, especially concerning the specification. As Hänni et al. describe election verifiers as "the ultimate way of challenging the protocol run", our hands-on experiences strengthen their findings. The recommendations by Haines and Rønne, however, might be possible next steps within the development process, if we were to have access to the source code of the voting system.

2 Context

We provide context to the independent individual verifiability tool implementations, which helps to understand the challenges faced by such a project.

[4] See `https://github.com/kastel-security/polyas-core3-second-device-verification`, `https://github.com/famoser/polyas-verification`, and Koppenhöfer [Ko23] for the individual tools.

[5] `https://www.electionguard.vote/elections/College_Park_Maryland_2023/`

2.1 Polyas Individual Verifiability Mechanism

The following gives a simplified overview of the individual verifiability mechanism used for the 2023 GI elections. We forgo a security analysis here and instead refer to Müller et al. [MT23] and the official specification [Tr24] for details.

After casting their ballot, the voter is presented with the available individual verifiability tools. For each tool, a QR code and a PIN (which changes every 30 seconds) is displayed. The QR code consists of a URL to the verifiability tool, which includes the voter ID and some randomness r. The voter then reads out the QR code on their second device (e. g., a smartphone), which loads the verifiability tool (e. g., by opening a webpage in the browser). The voter then enters the currently displayed PIN, and the tool downloads the voter's ballot and a re-encryption of that ballot from the server. Then, the tool performs an interactive zero-knowledge proof of correct re-encryption with the server over the two ballots (to assert that the ballot and its re-encryption encrypt the same plaintext). If successful, the tool decrypts the re-encrypted ballot using the randomness r from the QR code. The voter is shown this plain vote and the internal voter ID of the ballot, which the voter checks for correctness, or complains if incorrect. Further, the voter may download a receipt of the ballot signed by the server, which they can send to the election organizer for universal verifiability (to assert that all ballots are in the tally and no invalid ones were added). The verification window is limited to 30 minutes, after which the server refuses to participate. Further, the individual verifiability tool allows the voter to modify the displayed plain vote (without impacting the to-be-tallied ballot), rendering screenshots of this view unconvincing.

2.2 GI Experience with Verifiable Elections

Since 2004, the annual presidential and board elections of the "GI – Gesellschaft für Informatik" (German Informatics Society) are performed in a hybrid form, online with Polyas and by postal voting. In 2008, the GI developed a protection profile for a basic set of security requirements for online voting products (CC-PP-0037-2008, which is today outdated[6]), and since 2010 approached independent verifiability [OSV11]. However, such certifications are limited [BNV14], as well as the particular approaches taken for independent verifiability [Ol12; OSV11].

In 2019, the GI switched to a new Polyas system, which provided cryptographic artifacts for universal verifiability, and could be extended towards full end-to-end verifiability. GI researchers of the Karlsruhe Institute of Technology (KIT) and University of Stuttgart developed universal verifiability tools, which were successfully used since the GI election 2019 [Be19; Be21], and formally verified parts of the system's software [Ki22]. For the 2023 elections, the GI decided to additionally use the individual verifiability mechanism provided by Polyas to achieve full end-to-end verifiability. We describe this project in Sect. 2.3.

[6] https://www.bsi.bund.de/SharedDocs/Zertifikate_CC/PP/Archiv/PP_0037.html

2.3 Project Timeline

For the 2023 elections, the GI envisioned providing independent individual verifiability tools. Fig. 1 illustrates the most important milestones throughout the different phases of the project, which we briefly explain in the following.

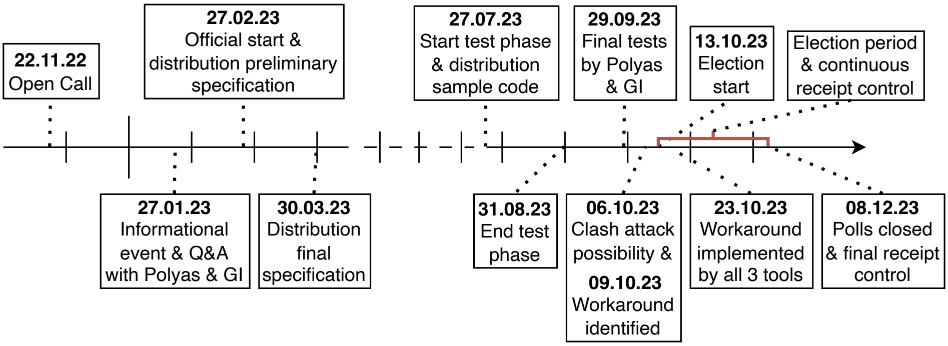

22.11.22
Open Call
27.02.23
Official start & distribution preliminary specification
27.07.23
Start test phase & distribution sample code
29.09.23
Final tests by Polyas & GI
13.10.23
Election start
Election period & continuous receipt control
27.01.23
Informational event & Q&A with Polyas & GI
30.03.23
Distribution final specification
31.08.23
End test phase
06.10.23
Clash attack possibility &
09.10.23
Workaround identified
23.10.23
Workaround implemented by all 3 tools
08.12.23
Polls closed & final receipt control

Fig. 1: Timeline of the most relevant milestones of the project.

Development Phase. In November 2022, the GI distributed a call[7] to develop an independent individual verifiability tool. The workload was described as appropriate for a thesis or a semester project for a computer science student. In January 2023, seven interested parties met with the GI and Polyas to outline tasks and schedules. The official start of the project was in February 2023, when the participants settled on a time schedule, and received the preliminary specification. In July 2023, the tools were almost finished. To help resolve the remaining issues, Polyas provided updated specifications, additional test data and the source code to their own individual verifiability tool. Further, access to a test election on their main voting system was granted. In September 2023, the tools were finished, and Polyas provided additional test elections for the final tests.

Pre-Voting Phase. In September 2023, the Polyas voting system and the verifiability tools were released for use by the GI. Subsequently, the election information and credentials were sent to the voters by postal mail.

However, after sending the letters, but before the voting phase had begun, the GI was made aware that the individual verifiability as implemented is susceptible against a clash attack. While each ballot contains an internal voter ID which binds the ballot to a specific voter, this internal voter ID was not given to the voters, hence the voters could not check that they indeed verify their own ballots. The GI found a workaround without sending out new election letters: The individual verifiability tools additionally print the internal voter ID

[7] `https://gi.de/wahlen/verifikation-der-gi-wahlen-tools-gesucht`

on the receipt. When the voter sends this receipt to the GI, it is then verified by the GI whether the sender corresponds to the internal voter ID on the receipt. This process was communicated to the GI members via mail, together with the general explanation about the newly added cast-as-intended verifiability.

Voting Phase. The voting phase was held from October 13 until December 8, 2023. Three independent individual verifiability tools were available, plus the one provided by Polyas. The voters could freely choose which verifiability tool(s) they want to use. Out of the $2,759$ Internet voters, 689 (25%) used at least one of the available tools. 72 voters sent their verification receipt to the GI, from which 30 included the internal voter ID checked by the election board.

Tally Phase. After the end of the voting phase, the votes were tallied using the official counting service of Polyas. In this phase, the available universal verifiability tools first introduced in 2019 [Be19; Be21; Ki22] were reused, now extended to check that the ballots from the receipts are included in the tally.

3 Experience Report and Recommendations

We share our experiences and formulate recommendations to implement independent individual verifiability. We categorize in implementation, testing, quality, and responsibility topics, each consisting of several sub-topics. Each (sub-)topic is introduced, and corresponding recommendations are given. While the recommendations are given independently to the GI project, to motivate and give context, we interleave — in separate paragraphs — with concrete experiences made as part of the GI project.

Concerning the experiences made, while in this section we primarily elaborate on issues and potential areas for improvement, we note that overall, the GI project was successful. Despite the inherent complexity, several independent implementations were created, deployed, and used by voters in the real election.

3.1 Implementation

An individual verifiability tool needs to implement cryptographic algorithms, the message exchange between devices and servers, and a user interface.

Polyas provided two specifications:[8] The main specification [Tr23] describes the high-level protocol, intended security guarantees, algorithms, and serializations needed to

[8] The specifications [Tr23; Tr24] only partially describe the protocol of the voting system; e. g., it remains unspecified how the ballot is encrypted on the voting device.

implement the verification. The second device specification [Tr24] describes the second device protocol, including the requests sent and received, while referring for the algorithms and serializations to the main specification. Polyas further provided code of their own second device implementation.

Algorithms. For Internet voting, cryptographic algorithms include domain-specific signatures, encryptions and zero-knowledge proofs, as well as auxiliary algorithms, e. g., pseudo-random generators, key-derivation functions, and message encoding. Available cryptographic libraries usually only cover parts of the necessary algorithms and often lack polish (e. g., insufficient documentation, unintuitive APIs, bugs).[9] A high-quality specification is, therefore, indispensable.

In the Polyas specification, parameters and return values are specified for every algorithm, as well as the algorithm itself as pseudocode or precise text. Complex algorithms are decomposed into multiple algorithms building upon each other. For most algorithms, the specification gives their context of use and some examples with input and output. Overall, the notation is consistent, and the presentation is compact. All these factors together contributed, such that in this part of the project, only few issues were encountered.

Algorithms

Clarity. Make parameters and return values explicit for each algorithm, and specify their implementation very clearly or use pseudo code.

Decomposition. Divide complex algorithms into smaller algorithms with a clear functionality, which can then build upon each other.

Context. Give context on the algorithm's usage, in particular by illustrating examples for input and output.

Serialization. Serialization and deserialization, respectively, define the conversion of an in-memory object (e. g., a public key) into a format to store it to a file, or send it over the network (e. g., as JSON). For interoperability of libraries and languages, this format needs to be precisely described, which should follow a clearly defined and ubiquitous standard.

In the main specification, Polyas specifies general serialization formats (e. g., the compressed form of ANSI X9.62 4.3.6 for representing elliptic curve points), which are implicitly used in the second device specification. Besides the specification, the source code by Polyas also provides insights for implementers. In this project, resolving serialization issues was

[9] For example, one of the implementers found and fixed an issue in PHP itself within one of the second device implementations (parsing a public key of a particular format produced an error, even if successfully parsed, see `https://github.com/php/php-src/pull/11055`). Finding issues in the programming language itself is very unusual and indicates that such APIs are rarely used.

responsible for up to half of the total effort of implementing the cryptography.[10] We give some examples here:

- **Java's `BigInteger` Encoding.** The specification describes the conversion of large integers to bytes by calling the function `toByteArray()` of Java's `BigInteger`. `BigInteger` considers integers as *signed*, with the most-significant bit indicating whether a number is positive or negative. However, natural serializations in JavaScript and PHP implementations omit this bit.[11]
- **Url-Safe Base64 Encoding.** The second device specification references a `decodeBase64()` function, but the sample data contains characters that are not part of a standard base64 encoding. Inspecting the Polyas source code revealed the usage of a URL-safe variant, which replaces `+/` by `-_`.
- **PDF Receipt Encoding.** The second device implementation generates a PDF receipt with cryptographic values, which are read out again automatically during the tally phase for verification. However, it was not clearly defined how these values need to be written to the PDF file, and different PDF writers generally use different approaches.[12] As a result, many receipts could not be read out automatically by all tools during the tally phase.

Serialization

Fit-for-Purpose. Chosen serialization and deserialization need to be suitable for their respective use-case.

Standards. Every serialization and deserialization needs to specify which technology-independent and widespread standard is employed.

Protocol. The protocol defines how algorithms are executed and messages are exchanged to achieve the security guarantees under the chosen adversary model. For individual verifiability tools, this typically involves the voter, who exchanges messages with their voting device or with the election server. The protocol, the security guarantees, and the adversary model need to be clearly described to the implementers for ensuring their implementation does not sabotage the protocol's goals (e. g., by accidentally publishing values assumed to remain secret).

[10] The diagnosis and repair of serialization and deserialization issues is time-consuming and difficult. Both usually involve extensive engagement with code, library and language, as well as studying appropriate standards.

[11] See, e. g., the PHP replication of the Java implementation, which prepends another zero-byte after the PHP-natural serialization in order to simulate the most-significant signed bit: `https://github.com/famoser/polyas-verification/blob/v1.3.2/src/Crypto/POLYAS/Utils/Serialization.php#L27`.

[12] In PDF, text is usually printed using the `Tj` function. By extracting the arguments to this function (e. g., using a regular expression), one can retrieve the printed text (e. g., `https://github.com/famoser/polyas-verification/blob/v1.3.2/src/Workflow/VerifyReceipt.php#L77`). However, this approach does not work in general, for example some PDF writers use compression or a custom encoding.

The Polyas second device specification describes the protocol, but it does not clarify the security claims and the adversary model. Instead, the specification references an academic publication [MT23] that describes a cast-as-intended mechanism which is the basis for the second device protocol. Neither having the full protocol specification of the system nor a security analysis with the adversary model and its security claims, we encountered the following issues:

- **Inconsistency of Specification with Published Protocol.** The second device specification is inconsistent with the referenced academic publication: In the implemented Polyas system, a performance optimization for big ballots is applied on the voter's device, which requires structural changes to the initially-published mechanism. Only one team discovered this discrepancy, notably *after* the election. At implementation time, the second device implementers were not fully aware of the protocol behavior, which could have resulted in implementation mistakes.
- **Unclear Receipts Handling.** To prevent a potentially dishonest Polyas server from dropping ballots, the second device implementations allowed voters to download a receipt (which they could then send to the election organizers, see Sect. 2.1). However, the exact process for collecting receipts is not described in the specification. The implementations then handled this functionality differently regarding how prominently the download was recommended and how well the whole process was explained (i. e., that the voter needs to send the receipt to the election organizers). Further, one of the implementations even deviated from the intended user flow, as it additionally offered voters to deliver the receipt to the election organizers within the implementation itself (thereby also unknowingly circumventing the clash attack mitigation, see Sect. 2.3).

Protocol

Complete. Specify the full protocol (in terms of involved parties, algorithms and message exchanges).

Security Claims. State the protocol's precise security claims (e. g., using a formal definition), possibly together with appropriate proofs.

Security Model. Define the adversary model and trust assumptions under which the security claims are achieved.

Intuition. Provide an intuition of the purpose of the executed algorithms and exchanged messages to prevent accidental mistakes.

User Interface. As individual verifiability directly involves the voter, the second device implementation needs to also provide a user interface. This interface needs to guide the voter to execute their part of the protocol efficiently and effectively (i. e., for detecting when their vote was not cast as intended). For the voter, the individual verifiability tool is part of their overall voting process. Correspondingly, the tool needs to be aligned with the main system, notably concerning terminology and general usage patterns.

Within the second device specification, Polyas included an example of how a ballot is displayed. However, the specification does not recommend specific wording, layout or interaction patterns, which, e. g., led to the following issues:

- **Inconsistent Wording.** When Polyas tested the second device implementations, they noticed that the wording differs from their main system. This included basic terminology such as "ballot" or "receipt", but also the ballot layout. While most deviations were unintentional, notably for the ballot layout, some implementations initially chose to use a different layout than in the Polyas system on purpose, e. g., in an attempt for ballots which are easier to understand for the voter. For consistency, all second device implementations aligned wording and layout to the Polyas system.
- **Unclear Voter Instructions.** Second device implementations were often unclear in how a voter is expected to behave, both in normal circumstances (e. g., checking both the ballot and the voter ID), and in exceptional cases (e. g., error messages which do not properly advise the voter how to proceed).

User Interface

Consistency. Define wording, layout, and interaction to be used consistently throughout all implementations for reducing friction for the users.

User Interaction. Specify the high-level interaction of the user with the system (e. g., executed checks) for achieving the security guarantees.

Error States. Describe error states that the user might experience and how the second device implementation needs to advise the user to proceed.

3.2 Testing

During and after the implementation, the tools need to be tested. In general, tests cannot use the exact election setting (as testing in the actual election is likely impossible), and monitoring needs to be kept to a minimum to safeguard the users' privacy. At the same time, mistakes are costly; they happen directly at the voter's side and need an individual investigation. Moreover, any issue or corresponding investigation may prevent successful verification or break vote secrecy. Further, while false positives may undermine trust in the system, false negatives may even lead to missing potential attacks. Therefore, proper testing is vital.

Samples. For checking the implementation's correctness, it is very useful to have execution samples of a presumed correct implementation. When given the same input as in the samples, an implementation needs to return the corresponding output. Samples may be given at different abstraction layers; e. g., both for some particular part of an algorithm and even for a full protocol execution.

Within the second device specification, Polyas uses an election as a running example, and for this election, samples for the algorithms are provided. Further, the main specification provides additional in- and output samples for some of the lower-level algorithms. The given samples focus on input and expected output, but do not provide intermediate values. We give examples of where we encountered issues with the given samples:

- **Limited Sample Set.** Samples for only a single election were given. Hence, the samples did not cover the full range of possible scenarios. Notably, this election only used a single ballot, while two ballots were used during the real election. Further, some behavior may only be observable when given many samples (e. g., the particular behavior of Java's `BigInteger` serialization), or some particularly-crafted samples (e. g., craft an empty ballot to discover crashes due to off-by-one errors).
- **Missing Intermediate Values.** The protocol also involves computing a ballot's hash value over its serialization. While the final hash value is given, the intermediate serialization value is not. During development, when the hash turns out to be wrong, the developer only knows that the input of the hash function was incorrect, but misses any efficient way to learn *which part* of the implementation was wrong.

Samples

Diversity. Ensure samples are numerous, structurally diverse, cover both realistic scenarios and edge-cases, and both successful and failing cases.

Abstraction. Provide samples for all levels of abstraction (serialization, algorithms and protocol) to enable targeted testing and diagnostic.

Intermediate Values. Include intermediate values for complex algorithms to ensure that the debugging of (partially) wrong results remains efficient.

Self-service. Support implementers in crafting their own samples to check for surprising behaviors under particular edge cases.

System Testing. Besides testing (parts of) the implementation from samples, the second device implementations also require tests that involve the user interface in order to include the user's point of view. Moreover, such tests must take the larger user interaction with the main system into account, e. g., by also considering the ballot casting before verification.

The main system developed and operated by Polyas is proprietary, and the implementers hence neither had access to the code nor could they operate or configure the system themselves. Thus, the system tests build on test elections that Polyas had set up for the implementers. We give some examples of where this setup led to issues:

- **Communication Overhead.** When implementers wanted to perform system tests, they needed to ask Polyas to set up a test election and deliver the corresponding election configuration (i. e., the voter credentials and second device configuration). The test elections then ran on Polyas infrastructure, hence in case they had to investigate any issues (e. g., error messages from the server), implementers needed to consult Polyas (e. g., for accessing their server logs in order to find the exact error). Polyas was generous with their time, but the communication overhead had a chilling effect on running system tests; e. g., many edge cases such as empty or very big ballots were not tested.
- **Missing Independence to Mitigate Clash Attack.** For preventing clash attacks, the implementations needed to integrate a fix on short notice (see Sect. 2.3). Once the fix was applied, a careful re-test that involved system testing was necessary. As no test election was available at this time, Polyas had to be asked to set up a new test election. This delayed rolling out the fix but also shifted the system's theoretical trust assumptions: If Polyas were a dishonest system provider and had been actively performing clash attacks, the request for setting up a new test election would have made Polyas aware that "something was up" and could have changed their tactic.
- **Complications in Demos and User Studies.** For gathering feedback on the second device implementations, performing user studies and demonstrations before (expert) audiences is useful. Ensuring that a test election was available when needed did require keeping Polyas in the loop. Besides the accompanying planning and communication overhead, misunderstandings were inevitable, and, e. g., resulted in a forced pause of a user study until its test election was again made available. Some tools instead opted to simulate the main system, and thereby gained flexibility, but accepted additional effort and complexity of developing such a simulation.

System Testing

Availability. Provide system tests continuously before, during, and after the elections to enable maintenance and reproducibility of tool behaviors.

Independence. Ensure that performing system tests is independent of the system provider for an efficient process and to prevent interference.

Self-service. Permit implementers to freely configure the system and observe its behavior (e. g., by monitoring the logs or the database).

3.3 Quality

Besides implementation and testing, many other factors contribute to the project's success. As the individual verifiability tools are integrated into the Internet voting system, they thus need to be a good fit to avoid becoming the system's weak link. This holds particular importance for security-critical aspects, as the tools might otherwise compromise the voter's privacy or verifiability.

Quality of Instructions. For a successful implementation of the tool, implementation instructions need to be precise without being overwhelming. This may include written specification, source code and specific answers to individual questions. High-quality instructions optimize work efficiency by freeing up resources that are better spent on other aspects of the project. For this matter, providing the source code of the underlying system may give a substantial orientation, as it serves as a definitive interpretation of all other instructions. If implementers can navigate the code freely, run tests against it, or inspect it with a debugger, they can resolve ambiguities and misunderstandings independently. Further, they may be inspired to improve their own implementation or identify bugs in the provided code to report to the provider.

Polyas provided the main system specification focusing on algorithms and the separate second device specification focusing on the second device protocol. Further, Polyas responded to clarification requests per email and updated the specifications to resolve identified gaps and ambiguities throughout the project. Since resolving issues on the cryptographic layer was time-consuming without access to the source code, Polyas later-on additionally distributed the source code and tests of their own second device implementation. Below are some examples of how instructions could have been improved:

- **Informal API Definitions.** Around half of the pages in the second device specification describe its API. However, describing and implementing API endpoints is rather mechanic and can be largely automated from formal API specification (e. g., using OpenAPI), which also avoids ambiguities and gaps. In one of the tools, insufficient API implementation led to a broken display of the election description (fixed shortly after the election started).
- **Clarifications via Email.** Describing and answering highly technical issues in an email is difficult and time-intensive. Further, the results of such discussions are impractical to share with other parties.

Quality of Instructions

Dogfooding. Use as a system provider the published specification as the basis of their own implementations to reduce gaps and ambiguities.

Formalizations. Define formally where reasonable (e. g., OpenAPI) to enable automation (e. g., code generation) and reduce gaps and ambiguities.

Efficient Q&A. Provide an efficient process to communicate issues and solutions, and, if necessary, update the specification to clarify.

Source Code. Provide source code and tests of the specification to efficiently resolve even highly technical issues.

Quality of Implementation, Maintenance, and Operation. The correctness of implementations for functional requirements can be mostly covered by tests. However, tools also need to fulfill non-functional requirements, which include, e. g., understandability of source code, compliance with security best practices, and usability and accessibility of the tool. Well-prioritized non-functional requirements lead to a system that performs well even in exceptional circumstances. After implementation, the tool needs to be provided to the end user. If it is executed in the browser, this requires to securely operate a server. If it is a smartphone app, this needs publication in the app stores. Both options come with individual challenges, e. g., proper configuration of the web server, or the publication process of the app stores.

In the GI project, implementers were mostly students with little industry experience in software development. The project's focus was primarily on the cryptographic part, where non-functional requirements such as usability were of lower priority. Further, some implementers were initially unaware that operating their own server would be part of the project (no implementer chose to publish in the app stores). We encountered the following challenges:

- **Low UI/UX Polish.** Not all second device implementations invested much effort in polishing their user interface (e. g., choosing appropriate colors, animations, or margins) or the user experience (e. g., giving clear instructions to the voters). Hence, voters might be unable to use the tools appropriately and, therefore, not achieve the intended security guarantees.
- **Dependency with a Vulnerability in the Active Election.** During the election, the widely-used *axios* JavaScript dependency disclosed a vulnerability. The vulnerability did not impact the second device implementations, as it was only relevant when using a *XSRF-TOKEN* cookie.[13] Still, as it is good practice to not use dependencies with known vulnerabilities, all second device implementations were patched nonetheless.

[13] For details, see `https://nvd.nist.gov/vuln/detail/CVE-2023-45857`.

- **Improper Header Configuration.** One of the implementations was operated using weak `http` header configurations, e. g., a header which helps to enforce `https` named `Strict-Transport-Security` remained unset. While assessing the impact of omitting this header is hard, such best-practice oversights should not happen.

Quality of Implementation, Maintenance and Operation

Non-functional Requirements. Appropriately identify and prioritize non-functional requirements (e. g., maintainability and usability).

Continuous Maintenance. Continuously observe for open flaws, and deliver the resolution to the voters in a timely manner.

Secure Operation. Provide the second device implementation in such a way that the voters can access it securely (e. g., no code manipulation).

3.4 Responsibility

Together, the main system provider, the election organizers, and the individual verifiability tool implementers hold the election. They rely on each other for a successful and trustworthy election, while their interests and capabilities differ. In consequence, they need to clarify, which party takes responsibility for which aspect early during the planning phase.

In the GI project, there was no contract or other explicit agreement between the involved parties. Responsibilities were not clearly defined and assigned. Instead, parties trusted the other parties to proactively contribute towards a successful election.

Responsibilities of the Main System Provider. The system provider can be expected to have deep domain knowledge in Internet voting and, in particular, know their own system best. As such, the provider needs to take the main responsibility to hold a secure and successful election using their system. In the context of the implementation of independent individual verifiability tools, this responsibility includes enabling the implementers to succeed in correctly and securely interfacing with their main system. It is in the main system provider's best interest that the implementers deliver their best possible work: A failure in the individual verifiability tools reflects poorly on the system as a whole, even if it is a false-positive. Further, well-constructed individual verifiability tools may uncover issues or potential for improvement in the main system.

Polyas provided sufficient specification, testing opportunities, and technical support to successfully implement several second device implementations. Polyas further tested the tools before the election and monitored them for availability during the election. Still, we identified the following issues:

- **Missing Quality Control.** To our knowledge, while Polyas tested the tools from the point of view of a voter, they did not perform other forms of quality control (e. g., code review, penetration testing). This would have likely reduced the observed quality issues (see Sect. 3.3).
- **Unclear Conditions of Endorsement.** In the Polyas main system, for each second device application, a corresponding QR code was shown after casting a ballot to start the verification. Polyas configured for which second device implementations such a QR code would be shown, while the conditions for which tools they selected to be included or not remained unclear. In the end, all tools were included, but the implicit possibility of exclusion gave Polyas negotiation power when requesting changes to the tools (e. g., concerning the wording used), jeopardizing the independence of the implementations.

Responsibilities of the Main System Provider

Technical Support. Provide appropriate resources to the implementers to enable them to deliver high-quality implementations in time.

Quality Assurance. Set clear quality thresholds over the implementations, e. g., based on code reviews, code scans, user testing, or penetration tests.

Responsibilities of the Election Organizers. While the main system provider has the general technical knowledge of the Internet election process, the election organizers know the specific context of the election, e. g., voter capabilities or the legal framework. Based on the context, the organizers decide on the process of the election, which includes both the choice of the Internet voting system (if any), and the particular system configuration. Both the system provider and the independent experts may contribute with their technical knowledge so that these decisions are taken in an informed manner in the given context.

In this project, the elections were organized by the GI. They were challenged in particular by the implementers' discovery of the possibility of a clash attack shortly before the election, and had to decide on a mitigation (see Sect. 2.3). We identified the following issues:

- **Unclear Project Target.** The purpose of the independent second device implementations was not clearly communicated.[14] Hence, the second device implementers could not consciously contribute towards fulfilling that purpose.
- **Ballot Validation.** The Polyas system does not prevent submitting invalid ballots (e. g., with too many candidates), but shows a corresponding warning to the voters when casting. However, the second device implementation was instead expected to

[14] The GI's election FAQ states that the verification's target was to "check the source code of Polyas" (see `https://gi.de/wahlen/faq`, visited on: 04/24/2024.).

display the ballot without a warning, even if it was invalid. The decision whether to show this warning or not should have been an active decision by the GI.

- **Receipts Not Checked Diligently.** In the tally phase, many receipts could not be read out automatically (see Sect. 3.1), and consequently were not validated. Further, around 30 receipts were directly stored by one of the second device implementations and also not validated. Although overall, much effort was spent on the receipts process, the validation of the receipts was not fully seen to the end.

Responsibilities of the Election Organizers

Communication. Ensure that all parties have timely access to all necessary information, especially in case of known flaws.

Purpose. Clearly communicate what is to be achieved by the independent implementations (e. g., more independence from the system provider).

Configuration. Take informed decisions about the employment of the security mechanisms, adjusted to the election context.

Supervision. Observe and enforce that decisions on the security mechanisms are respected and implemented by all parties.

Responsibilities of the Independent IV-Tool Implementers. The individual verifiability tool implementers' responsibility is that their tool works as intended. Depending on the project's purpose chosen by the election organizer, additional responsibilities need to be assumed, e. g., to vouch for the implemented protocol, or that the respective main system (respectively the observed interaction) is of high quality. With growing responsibilities, the remuneration of the implementers becomes necessary, which then needs to be carefully arranged so as not to jeopardize the intended independence.

In the GI project, the second device implementers were volunteers not paid by and without any contractual binding to the GI or to Polyas. Naturally, assigning responsibilities to or even holding implementers accountable for unmet expectations are both very difficult in this scenario, both ethically and practically. Nevertheless, the implementers were implicitly responsible that their part of the election could proceed as planned. Besides respectful and transparent communication, the implementers reliably provided (and improved on short notice) their independent tools while respecting the surrounding conditions by Polyas and the GI. In the following, we list some of the occurred issues nonetheless:

- **Aborted Implementations.** Out of seven initial parties, three provided a working implementation in time. In this project, the four aborted implementations did not lead to any issues, as all parties communicated openly about the state of their project ahead of time.

- **Limited Availability of Implementers.** In the first election week, the implementer of one of the second device implementations was unreachable. This absence was planned and communicated beforehand, but still impacted the election: This tool received the clash attack mitigation patch only after the other tools were already patched, notably during the active election.
- **Limited Tool Availability.** During the election, Polyas kept monitoring the running instances of the second device implementations to ensure availability. At least one instance was detected to be unavailable at some point, and the implementers needed to react to restore the availability of their tool.

Responsibilities of the Independent IV-Tool Implementers

Transparency. Communicate transparently about the tools (e. g., whether development is on track with respect to the election's timeline).

Availability. Implement necessary changes, and guarantee the availability of the tools, especially while the election is active.

Respect Surrounding Conditions. Consent to the surrounding conditions, e. g., concerning availability, functionality, or responsible disclosure.

4 Conclusion

To make independent individual verifiability tools a reality, detailed and exact documentation is necessary. We reported on our experiences of developing and deploying three independent individual verifiability tools for the 2023 GI elections with recommendations regarding implementation, testing, quality, and responsibilities.

Besides a precise description of the protocol and its algorithms, a precise treatment of how to interface with the main system (e. g., the serialization standards) and with the voter (e. g., consistent wording between both systems) is vital. Moreover, in order to reduce the effort of both the system provider and the independent implementers and prevent (too) late surprises, a diverse set of test data and self-service access to the main system are needed. For the system provider, the independent implementations not only benchmark whether the documentation is complete and accurate and whether their system behaves as expected, but also result in a very detailed review of the part of their system that interfaces with the individual verifiability tool. This may result in helpful feedback to improve their system; as demonstrated in the GI project by the discovery and mitigation of a potential clash attack.

All in all, successful employment of independent tools is more than just the provisioning of a functional implementation. The tools become part of the voters' voting experience and must not become the weak link, both from a security perspective and a user experience point of view. Further, the tools also need to fulfill the targets of the election organizers,

e. g., the distribution of trust away from the system provider. This needs to clearly set quality targets and assign responsibilities without jeopardizing the independence of the independent implementations.

Many of the recommendations appear common sense. Still, some aspects are easily forgotten, apparent in the GI project primarily in the quality and responsibility topics. While we reported on the experiences of the first project of this kind, we hope other projects will emerge, find the recommendations helpful, and report on the results.

References

[Be19] Beckert, B.; Brelle, A.; Grimm, R.; Huber, N.; Kirsten, M.; Küsters, R.; Müller-Quade, J.; Noppel, M.; Reinhard, K.; Schwab, J.; Schwerdt, R.; Truderung, T.; Volkamer, M.; Winter, C.: GI Elections with POLYAS: a Road to End-to-End Verifiable Elections. In: Fourth International Joint Conference on Electronic Voting (E-Vote-ID 2019), Lochau / Bregenz, Austria Oct. 1–4, 2019. TalTech Press, pp. 293–294, 2019, URL: https://digi.lib.ttu.ee/i/?13563.

[Be21] Beckert, B.; Budurushi, J.; Grunwald, A.; Krimmer, R.; Kulyk, O.; Küsters, R.; Mayer, A.; Müller-Quade, J.; Neumann, S.; Volkamer, M.: Aktuelle Entwicklungen im Kontext von Online-Wahlen und digitalen Abstimmungen, tech. rep., Karlsruhe Institute of Technology (KIT), 2021, DOI: 10.5445/IR/1000137300.

[Be24] Benaloh, J.; Naehrig, M.; Pereira, O.; Wallach, D.: ElectionGuard: a Cryptographic Toolkit to Enable Verifiable Elections. In: 33rd USENIX Security Symposium (USENIX Security 2024), Philadelphia, PA, USA Aug. 14–16, 2024. https://eprint.iacr.org/2024/955, USENIX Association, 2024, IACR: 2024/955, URL: https://www.usenix.org/conference/usenixsecurity24/presentation/benaloh.

[Be87] Benaloh, J.: Verifiable Secret-Ballot Elections, PhD thesis, Yale University, 1987, URL: https://www.microsoft.com/en-us/research/publication/verifiable-secret-ballot-elections.

[BNV14] Buchmann, J.; Neumann, S.; Volkamer, M.: Tauglichkeit von Common Criteria-Schutzprofilen für Internetwahlen in Deutschland. Datenschutz und Datensicherheit-DuD 38 (2), pp. 98–102, 2014, DOI: 10.1007/s11623-014-0040-x.

[CGG19] Cortier, V.; Gaudry, P.; Glondu, S.: Belenios: A Simple Private and Verifiable Electronic Voting System. In: Foundations of Security, Protocols, and Equational Reasoning - Essays Dedicated to Catherine A. Meadows. Vol. 11565. Lecture Notes in Computer Science, Springer, pp. 214–238, 2019, DOI: 10.1007/978-3-030-19052-1_14.

[Co23] Cortier, V.; Gaudry, P.; Glondu, S.; Ruhault, S.: French 2022 legislatives elections: a verifiability experiment. In: 8th International Joint Conference on Electronic Voting (E-Vote-ID 2023), Luxembourg City, Luxembourg Oct. 3–6, 2023. Lecture Notes in Informatics, Gesellschaft für Informatik (GI) e.V., 2023, URL: https://inria.hal.science/hal-04205615.

[Ha20] Haenni, R.; Dubuis, E.; Koenig, R.; Locher, P.: CHVote: Sixteen Best Practices and Lessons Learned. In: 5th International Joint Conference on Electronic Voting (E-Vote-ID 2020), Bregenz, Austria Oct. 6–9, 2020. Vol. 12455. Lecture Notes in Computer Science, Springer, pp. 95–111, 2020, DOI: 10.1007/978-3-030-60347-2_7.

[Ha23] Haenni, R.; Koenig, R.; Locher, P.; Dubuis, E.: CHVote Protocol Specification. IACR Cryptology ePrint Archive, 2023, IACR: 2017/325, URL: https://eprint.iacr.org/2017/325.

[HR21] Haines, T.; Rønne, P.: New Standards for E-Voting Systems: Reflections on Source Code Examinations. In: International Workshops on Financial Cryptography and Data Security (FC 2021), Revised Selected Papers, Virtual Event Mar. 5–5, 2021. Vol. 12676. Lecture Notes in Computer Science, Springer, pp. 279–289, 2021, DOI: 10.1007/978-3-662-63958-0_24.

[Ki22] Kirsten, M.: Formal Methods for Trustworthy Voting Systems: From Trusted Components to Reliable Software, PhD thesis, Karlsruhe Institute of Technology (KIT), 2022, DOI: 10.5445/IR/1000155115.

[Ko23] Koppenhöfer, O.: Individual verifiability in the Polyas e-voting system, Bachelor's Thesis, Institute of Information Security, University of Stuttgart, 2023, DOI: 10.18419/opus-13838.

[MT23] Müller, J.; Truderung, T.: CAISED: A Protocol for Cast-as-Intended Verifiability with a Second Device. In: 8th International Joint Conference on Electronic Voting (E-Vote-ID 2023), Luxembourg City, Luxembourg Oct. 3–6, 2023. Vol. 14230. Lecture Notes in Computer Science, Springer, pp. 123–139, 2023, DOI: 10.1007/978-3-031-43756-4_8.

[Ol12] Olembo, M.; Kahlert, A.; Neumann, S.; Volkamer, M.: Partial Verifiability in POLYAS for the GI Elections. In: 5th International Conference on Electronic Voting 2012 (EVOTE2012). Vol. P-205. Lecture Notes in Informatics, Gesellschaft für Informatik (GI) e.V., pp. 95–109, 2012, URL: https://dl.gi.de/handle/20.500.12116/18228.

[OSV11] Olembo, M.; Schmidt, P.; Volkamer, M.: Introducing Verifiability in the POLYAS Remote Electronic Voting System. In: Sixth International Conference on Availability, Reliability and Security (ARES 2011), Vienna, Austria Aug. 22–26, 2011. IEEE Computer Society, pp. 127–134, 2011, DOI: 10.1109/ares.2011.26.

[Sw24] Swiss Post Ltd.: Swiss Post Voting System – System Specification, tech. rep., version 1.4.1, Swiss Post Ltd., 2024, URL: https://gitlab.com/swisspost-evoting/e-voting/e-voting-documentation/-/blob/5a0987fb206510c5494312f505b7d56a5e2c1624/System/System_Specification.pdf, visited on: 06/24/2024.

[Tr23] Truderung, T.: POLYAS 3.0 Verifiable E-Voting System, tech. rep., version 1.3.2, Berlin, Germany: POLYAS GmbH, 2023, URL: https://github.com/kastel-security/polyas-core3-second-device-verification/blob/82a74d576469deadbcda600ded81f13f0de98f1c/doc/polyas3.0-verifiable-1.3.2.pdf, visited on: 07/31/2023.

[Tr24] Truderung, T.: Polyas-Core3 Second Device Protocol, tech. rep., version 1.1, Berlin, Germany: POLYAS GmbH, 2024, URL: https://github.com/famoser/polyas-verification/blob/7a0f708f41e471a0208c02c989ec727f236e349e/spec/second-device-spec-1.1.pdf, visited on: 07/24/2024.

Acknowledgement

This work was supported by funding from the topic Engineering Secure Systems, subtopic 46.23.01 Methods for Engineering Secure Systems, of the Helmholtz Association (HGF) and by KASTEL Security Research Labs.

The peasants are revoting, sire, and at random times

Enka Blanchard [1,2] and Peter Y. A. Ryan [3]

Abstract: Inspired by how coercion happens in practice in complex social networks where weak ties are more important than unilateral force, we propose a new mitigation mechanism based on revoting: adding for each voter a secret, extra revoting period of random duration. Not only does this add opportunities to resist coercion, it also adds friction to the system, increasing the costs for both coerced voter and coercer, and disincentivising coercion. We investigate generalisations and variants of this mechanism in different frameworks (such as with and without shareable credentials), considering the optimal strategies for the electoral authority, voters and coercer. We also propose an implementation of the mechanism in a simple setting.

Keywords: Coercion mitigation, Revoting, E2E verifiable voting

1 Introduction and central idea

Freedom from coercion is one of the fundamental requirements for free and fair elections. It is a stronger property than vote privacy[5] and comes with a stronger attacker model: rather than passively observing the unfolding of the election, the attacker may actively interfere with the voters and the system. Both coercion and counter-measures against it have long been studied in the election systems literature [HS19]. For in-person voting, coercion is typically countered by ensuring that voters cannot be observed as they make their selection and cast their ballot. Securing remote voting, whether by postal vote or through the internet, remains an open problem as there is no practical way to enforce such isolation at the time of casting, making coercion a serious risk. However, there is a wide range of possibilities for coercers and not all of them use force — with some instead relying on social ties, raising the possibilities of discouraging coercion by raising its cost.

One option that has been used in practice is to implement revoting, in which voters can cast a new ballot replacing their previous vote — especially when a vote cast in a polling place cancels any online vote cast, as in Estonia [Pe23]. More cryptographically involved counter-measures have also been proposed, with JCJ presenting a seminal approach in which

[1] Laboratoire d'Automatique, de Mécanique et d'Informatique industrielles et Humaines – UMR CNRS 8201, Université Polytechnique Hauts-de-France, enka.blanchard@cnrs.fr, https://orcid.org/0000-0003-1511-8864

[2] CNRS Center for Internet and Society, Paris, France, enka.blanchard@cnrs.fr, https://orcid.org/0000-0003-1511-8864

[3] University of Luxembourg, peter.ryan@uni.lu, https://orcid.org/0000-0002-1677-9034

[4] Authors are listed in alphabetical order as they made equivalent contributions. A longer version of this paper with complete proofs and more detailed social analysis is available at `https://hal.science/hal-04650731`,

[5] Although both are often discussed in similar ways, we will focus on coercion rather than vote buying, as the latter often assumes a weaker adversary with different strategies: only the coercer can use negative as well as positive incentives [FRS18].

doi:10.18420/e-vote-id2024_03

each voter can vote either with real credentials or fake credentials which are indistinguishable to the coercer [JCJ05], although full coercion resistance wasn't achieved in the original iteration [CGY24]. Tallying is performed such that, after anonymisation, votes with valid credentials are counted and those with invalid credentials or duplicates are discarded in a way that is verifiable but undetectable. The prior work closest to ours appears to be Caveat Coercitor, [Gr13]. This uses JCJ-like credentials but with the policy that when different votes are found to have been cast with a given credential, all votes cast with this credential are nullified and a coercion flag is raised.

Revoting, with the policy of counting that last vote cast for each voter, has been critiqued on the grounds that the coercer can simply wait to the last moment before voting closes to cast his vote. This leads us to propose blurring the deadline for (re-)voting to counter this coercer strategy. We present two contributions in this paper:

- A simple mechanism to mitigate coercion that should be adaptable to many voting systems: having an extra voting period of random duration.
- An investigation of this mechanism in different frameworks (with and without shareable credentials) and variants, considering the optimal strategies for the electoral authority (EA), voters and coercer.

In the different cases we investigate, we show that the mechanism makes the coercer's task harder by forcing them to invest more time and resources — sometimes in an unbounded fashion if they want to guarantee getting the coerced vote. Moreover, we show that it adds opportunities to detect coercion through network analysis, depending on the coercer's behaviour. Beyond the immediate impact, a second benefit is that it serves to disincentivise would-be coercers.

The central idea is to introduce a secret randomisation of the cutoff point for casting votes (e.g., with secret individual deadlines or by varying the policy regarding which vote is counted). In its simplest variant, instead of having the possibility to cast a new vote (replacing the previous one) until the end of the election, each voter is given a secret additional time during which they can continue revoting at will.

More formally, we consider a system with a conventional voting period from time T_0 to T_1 during which all votes are cast in the usual way. We introduce and extended voting period between T_1 and T_2 during which each voter will have a secret, random cutoff point t_i (with $T_1 < t_i < T_2$). Votes cast after t_i will be discarded. Uncoerced voters should of course cast their vote before T_1. This creates a game of strategy between the voter and the coercer, with the winner being the one to cast the vote that is counted. Thus the players can increase their odds of wining by casting more votes, but at the cost of investing greater effort.

2 Performance of various strategies

This section looks at bounds on the probability of the coercer successfully casting the vote that is ultimately tallied, depending on the choices of the EA. We will use proof schemes and refer the reader to the full version of the paper for complete proofs in both discrete and continuous cases.

2.1 Non-shareable credentials

If the voters need a physical token to cast a vote, the system becomes equivalent to a game where the coercer chooses a partition of the time (depending on who has the token) and wins iff t_i lies within their time segments. If we assume that the coercer's cost increases linearly with the total time and that they have a budget corresponding to a fraction b of the total time, they can guarantee a winning probability at least equal to b by choosing random time segments. The EA in turn can guarantee that it is at most b by taking t_i uniformly at random in the interval $\{T_1, T_2\}$.

2.2 Shareable credentials: unlimited votes

In the shareable credential model, a password can suffice to vote, which allows the coercer and voter to vote quasi-simultaneously as many times as they want — or with some negligible rate-limiting if the EA wants to prevent DOS attacks. The vote that gets tallied is the last one cast before t_i. Instead of the coercer's cost scaling with the total time the token is held, both voter and coercer have a small cost for each vote they cast. Moreover, although some voters will cast many votes for various reasons, they are generally a small minority [HPW15], and a pattern of frequent revotes in a constituency would indicate probable coercion. Thus, the coercer's interest is to find a balance between cost, probability of winning and detectability, while the voter wants to increase their chance of winning or increase the coercer's costs, while not spending too much effort. The question is then: which strategies are optimal for each agent — potentially depending on their knowledge? Relatedly, which models and parameters — which are presumably chosen by the EA — are most disadvantageous to the coercer?

We make two simplifying assumptions, both to the coercer's advantage. First, we assume that the voter strategies are generally simple, e.g. a voter could vote at regular times every day. If guessed accurately by the coercer, this could be countered for an equal cost with a probability of winning asymptotically equal to 1. In practice, we'll generally assume that voters follow a Poisson point process of rate λ_v.

Second, we assume that t_i is distributed uniformly at random between T_1 and T_2, and that this is publicly known — as lopsided distributions are generally to a coercer's advantage if they have at least as much information as the voters. Unlike in the non-shareable token model, we can immediately observe that the coercer cannot guarantee that the final vote will be cast in the way they desire, and their total cost is unbounded — even if the voter is restricted to a single vote. A coercer with a budget for k votes will in fact have a probability at most $e^{-\lambda_v \frac{T_2-T_1}{k}}$ of casting the last vote.

If the coercer also adopts a Poisson point process with rate λ_c, we can merge the two processes, with the coercer winning with probability $\frac{\lambda_c}{\lambda_c+\lambda_v}$. This is not optimal and in fact worse than voting at regular intervals. Indeed, if the coercer votes at the start of each time slot of length $\frac{1}{\lambda_c}$, they win if the voter does not cast a vote between the start of the last time slot and t_i. Integrating over the distribution of t_i in the final time slot, this becomes:

$$\lambda_c \int_0^{1/\lambda_c} e^{-\lambda_v t} dt = \frac{\lambda_c}{\lambda_v}(1 - e^{-\lambda_v/\lambda_c}) \geq \frac{\lambda_c}{\lambda_c + \lambda_v}$$

With the current assumptions and a fixed budget λ_c, this strategy is in fact optimal, which can be shown by taking any two adjacent time slots and showing that the coercer maximises their probability of winning by having them of equal length. By concavity of $x \mapsto x - e^{-1/x}$, one can also show that this remains true if the coercer's budget only has expected value λ_c.

We can provide a generalisation of this if we assume that the strategies of both voter and coercer are offset-invariant (which is the case for regular voting and Poisson point processes). In such a case, the optimal strategy for both agents is to vote at regular intervals with a random initial offset. Assuming that $\lambda_c \geq \lambda_v$, the coercer's winning probability is $1 - \frac{\lambda_v}{2\lambda_c}$, and $\frac{\lambda_c}{2\lambda_v}$ otherwise.

3 Extensions and discussion

The results above can be extended to many other variants, such as having a random set of time periods each admitting at most k votes. One must be careful as some variants can give an advantage to the coercer, such as taking the most frequent vote cast or taking a ballot at random. Questions of information and channel security also matter: what if one could communicate t_i to the voter? One could also consider accomplices or other actors, non-linear costs or logistical aspects. As a final extension, it would be interesting to implement this kind of scheme in E2E verifiable systems. This seems particularly difficult as we are dealing with potentially quite complex policies, involving randomisation, for selecting the votes to be included in the tally. To implement such policies in a verifiable way will involve zero-knowledge protocols, verifiable sources of randomness, distributed computation etc. Implementation of such variants is left to future work, as is the formalisation of the required properties and associated proofs.

All of these considerations bring us to a central trade-off between the complexity of the system, its understandability and its perceived legitimacy. Having an extra voting period could make voters afraid that their vote would be discarded, or that they need to vote frequently for their vote to be recorded. More generally, the scheme could be criticised for potentially infringing on principles of democratic equality, with different voting periods for various sets of voters being seen as unfair. The answer is complex but relies on three elements. First, as voters have an equal opportunity to vote in the initial period the only ones affected by this system are the potentially coerced voters. Second, this is mostly a question of degree and not nature: in systems without revoting, voters who go to the polling office early renounce their rights to change their minds later — and some electoral systems already have non-overlapping voting periods for different populations. Finally, whether voting early or late is an advantage is context-dependent. Although late voters have more information, if early voters can commit to their choices (as in a Stackelberg game), they can force the other voters to change their voting patterns.

4 Acknowledgements

This paper was supported by the Luxembourg National Research Fund (FNR) under the CORE project EquiVox (C19/IS/13643617/EquiVox/Ryan). The authors wish to thank Lê Thành Dũng (Tito) Nguyễn and Édouard Thomas for their mathematical assistance.

Bibliography

[CGY24] Cortier, Véronique; Gaudry, Pierrick; Yang, Quentin: Is the JCJ voting system really coercion-resistant? In: 37th IEEE Computer Security Foundations Symposium (CSF). CSF 2024, IEEE, Enschede, Netherlands, 2024. This is the long version of the paper published at CSF 2024.

[FRS18] Frye, Timothy; Reuter, Ora John; Szakonyi, David: Hitting Them With Carrots: Voter Intimidation and Vote Buying in Russia. British Journal of Political Science, pp. 1–25, 2018.

[Gr13] Grewal, Gurchetan S; Ryan, Mark D; Bursuc, Sergiu; Ryan, Peter YA: Caveat coercitor: Coercion-evidence in electronic voting. In: 2013 IEEE Symposium on Security and Privacy. IEEE, pp. 367–381, 2013.

[HPW15] Heiberg, Sven; Parsovs, Arnis; Willemson, Jan: Log analysis of Estonian internet voting 2013–2014. In: International Conference on E-Voting and Identity. Springer, pp. 19–34, 2015.

[HS19] Haines, Thomas; Smyth, Ben: Surveying definitions of coercion resistance. Cryptology ePrint Archive, Paper 2019/822, 2019. https://eprint.iacr.org/2019/822.

[JCJ05] Juels, Ari; Catalano, Dario; Jakobsson, Markus: Coercion-resistant electronic elections. In: Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society. pp. 61–70, 2005.

[Pe23] Pereira, Olivier: Individual Verifiability and Revoting in the Estonian Internet Voting System. In: Financial Cryptography and Data Security. FC 2022 International Workshops: CoDecFin, DeFi, Voting, WTSC, Grenada, May 6, 2022, Revised Selected Papers. Springer-Verlag, Berlin, Heidelberg, p. 315–324, 2023.

Is Benaloh Challenge Suitable for Polling Station Elections ?

Roberto Araujo [1], Marcos Simplicio [2], Eduardo Cominetti [2], Paulo Matias [3], and Jacques Traoré [4]

Abstract: A few voting schemes for polling station elections are end-to-end (E2E) verifiable, allowing voters to check if their choices were cast-as-intended and recorded-as-cast. Some solutions rely on the Benaloh challenge, through which voters can spoil their ballots to verify if they were encrypted properly. In this paper, we discuss two potential issues and mitigations when this technique is used for in person voting with direct-recording electronic (DRE) machines: (1) leakage of voters' intentions in some scenarios; and (2) malicious DRE machines gathering statistical data aiming to trick challengers.

Keywords: Electronic Voting, E2E, Benaloh challenge, DRE

1 Introduction

Electronic voting is continuously evolving. An important milestone of this advance was the introduction of end-to-end verifiability (E2E), adopted in solutions like: Prêt-à-Voter [CRS05], which uses pre-made paper ballots with candidates and ciphertexts; and ElectionGuard [BN22], which allows voters to spoil their choices to verify if they were correctly encrypted (the so-called "Benaloh challenge" [Be87]).

While E2E-verifiability can improve elections auditability, such systems have found little adoption in polling station elections. Consequently, identifying potential security issues that are not strictly technical, but related to human aspects, is a challenging task. Indeed, since Benaloh introduced the idea of challenging voting machines by auditing committed ciphertexts in [Be87], and introduced a scheme separating vote creation from vote casting [Be06], a few drawbacks and improvements have been identified [Be07]. For example, voters might be coerced to make a challenge based on the voting machine's commitment [Ch11, Be07]. Paper-based E2E systems, such as Prêt-a-Voter [CRS05] and Punchscan [PH10], are also affected by analogous coercion attacks, for example, using a scratch-off card [Ke10].

In this paper, we discuss two possible issues and migigations when Benaloh challenges are employed in real elections – which (to the best of our knowledge) have not been discussed in the literature. Similarly to attacks described in [Ch11, Be07, Ke10], those hereby presented exploit commitments made during the voting procedure.

[1] Universidade Federal do Pará, Brazil, rsa@ufpa.br, https://orcid.org/0000-0002-3691-4299

[2] Universidade de São Paulo, Brazil, msimplicio@usp.br, https://orcid.org/0000-0001-5227-7165; ecominetti@larc.usp.br, https://orcid.org/0000-0003-0434-0013

[3] Universidade Federal de São Carlos, Brazil, matias@ufscar.br, https://orcid.org/0000-0002-6504-5141

[4] Orange Innovation, France, jacques.traore@orange.com, https://orcid.org/0000-0002-2993-5526

doi:10.18420/e-vote-id2024_04

2 Benaloh challenge in polling station elections

The Benaloh challenge [Be87] is a versatile idea that improves auditability in both online (e.g., as done in Helios voting [Ad08]) and polling station elections (e.g., as in Wombat [Be12]). In what follows, we consider its application to a generic polling station scenario, comprising a DRE voting machine placed in a controlled environment. The voting procedure is, thus, as follows: (1) the voter uses the DRE to selects the desired candidates; (2) the DRE then encrypts the voter's choices, and commits to this encryption (e.g., by printing it); (3) the voter chooses to audit the encrypted vote (going to step 4a) or to cast it (going to step 5); (4a) for vote auditing, the DRE reveals the input used for computing the ciphertext, including the user's choices in plaintext and any secrets employed (e.g., a nonce); (4b) the voter uses an independent device to verify that the (now spoiled) vote was correctly encrypted, and goes back to step 1 to cast a new vote; (5) when the voter decides to cast a vote, the DRE stores the committed ciphertext, and may publish it on a public bulletin board that is used as the input for the tallying process. In step 4, we assume that voters may audit (spoil) votes either freely or with the help of election officials.

In this setting, we consider three different auditing approaches: (A) voters can freely choose the candidates in the challenge, and then use their own devices (e.g., smartphones) for auditing; (B) voters can freely choose the challenge contents, but auditing must use special devices provided by election authorities; or (C) candidates in challenges are defined by election officials instead of voters, and the audit is done in any independent device.

3 Issue 1: Spoiled votes can leak voters' true intentions

In the scenario described in Section 2, vote secrecy depends on how and where spoiled votes are audited. Specifically, in the auditing procedure of case (C), voters might be free to keep or to distribute their spoiled votes. Cases (A) and (B), however, require additional attention, since one would require a special auditing device that verifies spoiled votes without revealing any information about their contents. If those conditions are not met, information about the voter's actual preferences might leak in at least one relevant real-world scenario: when voters choose the *same* candidates while casting and auditing votes. We argue that this is a quite likely scenario, since the whole goal behind the Benaloh challenge is for voters to gain confidence that *their* choices are correctly recorded (rather than simply "any choice").

This concern is particularly problematic when adversaries can get access to spoiled votes. For example, suppose the voting machine and the auditing device are placed in different controlled places (e.g., in separate rooms) in scenario (B). In this case, voters might be coerced into showing their spoiled votes, carrying their choices in plaintext, when moving between places. Similarly, in scenario (A), spyware might be installed in the devices employed by voters, thus revealing the chosen candidates to attackers. Printed spoiled votes might also be thrown in the trash by unaware voters, and then obtained by adversaries.

One straightforward mitigation for this issue is to rely on voters' awareness. After all, a voter fearing coercion can always choose candidates that do not correspond to their actual choices when spoiling (some) votes. Unfortunately, however, the required degree of awareness may not necessarily easy to ensure in practice given the extra time taken by vote auditing – one of the probable reasons why few voters spoil their votes in real world experiments [Te21]. Furthermore, encouraging voters to only cast random choices leads to other issues (see Section 4). Hence, extra measures may be necessary in practice. For example, some deployments might forbid (or at least strongly discourage) voters from keeping copies of spoiled votes. Instead, they might leave the printed commitment and corresponding plaintext in a "spoiled votes box", so the challenges could be later verified by election officials (analogously to ThreeBallot [Ri06]). Obviously, to prevent the very same attacks this approach aims to address, a chain of custody for those spoiled votes is required.

Another promising mitigation is to promote overseen challenges as the main election auditing procedure (or even disallow non-overseen options). In this case, the election would focus on the auditing Approach (C), where the challenges do not involve voters' real intentions – and, hence, cannot leak them. For example, an election authority could enter the waiting queue from time to time, and randomly pick a card describing the auditing procedure to be performed: number of spoils, candidates in each challenge, etc. The auditing procedure should then be supervised by voters and third parties, for added transparency. This approach, if properly implemented, has the added benefit of ensuring a minimum number of spoils are made during the election, which is itself a challenging issue [Te21].

4 Issue 2: Statistical information leakage

The more voters spoils ballots using Benaloh challenges, the higher is the chance to identify a malicious DRE machine and, hence, the better the confidence in the corresponding election system. Although this is desirable for auditing purposes, more spoiled votes also means more information potentially learned by the DRE, as discussed in the following sub-sections.

As before, we assume the DRE can be trusted for protecting vote secrecy. After all, since DREs receive votes cast, they might be able to link choices to the corresponding voters in some setups (e.g., if the DRE itself is responsible for user authentication). Conversely, we do not assume the DRE is trustworthy for integrity: it may try to create encrypted ballots whose selections do not match the voters' choices. We also assume that auditing devices are trustworthy, i.e., they correctly verify the ciphertext and plaintext data correspondence.

Using Statistical Information to Change Votes. As described in Section 2, we consider a voting system where the DRE receives a vote, encrypts it, and reveals the plaintext selections if challenged by the voter. During this procedure, it might try to gather statistical information aiming to mount attacks.

Suppose that the first 60% voters audit at most one vote. The DRE might then assume that some external factor (e.g., long queues) is preventing voters from challenging it twice and, thus, that recording the wrong choices after the voter spoils a single ballot is likely to go unnoticed. Even though such misbehavior would be detected if the DRE is challenged twice by the same voter, the statistical information gathered gives some extra confidence that it might adopt such a malicious behavior. As the election progresses, the machine may decide whether or not it has enough information to violate vote integrity for some voters.

Races with Many Candidates. In races with many candidates, extra statistical information can be gathered by a malicious DRE depending on how challenges are conceived. In particular, challenges should not comprise only randomly selected candidates, or they can be easily detected.

To illustrate this concern, consider the example of Brazil, where a unique DRE receives roughly 500 votes per race, and some races (e.g., for state deputy) may have more than 2000 candidates. In this scenario, some candidates will never be chosen (since the set of candidates is larger than the set of voters), while some candidates may naturally receive more votes than others. Hence, a malicious DRE could gather statistical data about the initial (say, 50%) votes to learn which are common patterns and which are not Ballots with randomly selected candidates are likely to fall into the second category and, hence, hint the DRE that it should behave honestly when encrypting it. Conversely, ballots in the first category are more likely to be cast without spoiling, especially considering that voters seem to rarely spoil votes in real-world election [Te21]. Hence, a malicious behavior from the DRE would go unnoticed with higher probability when changing such common votes – with the additional advantage that voters would not be alarmed, since the candidate of their choice would still have at least one vote in the tally.

Mitigations. Once again, promoting overseen challenges, made by election authorities and supervised by interested parties, is a promising mitigation approach for such concerns. One requirement, though, is that challenges must include a non-biased number of spoils and candidates, e.g., via randomly picked cards describing the procedure to be followed for each auditing. Those cards should include not only random candidate selections, but also selections following probable distributions – e.g., based on voting intention surveys, as well as suggestions from running candidates' parties. For added transparency, it may be even possible to build to-be-spoiled ballots with inputs from local observers – for example, some of those ballots could have a few columns with different candidate names, and observers should choose one (without revealing their whether or not that choice match their preferred candidates). Similar choices may be given for the number of spoils to be performed.

Another possibility, which also helps with protecting vote secrecy, consists in separating the system's commitment and challenge procedures [Be06]. In this setting, voters could ask a *voting device* to encrypt as many votes as desired, including different candidate;

the ciphertext containing the voter's actual choices is then cast in a *registration device*, while other ciphertexts are spoiled. Implementing this approach in practice involves some challenges, though, including the overall logistics of the election procedures and ensuring that there are no direct nor subliminal communication channels between such devices.

5 Conclusions and Future Work

Enhancing DREs using E2E ideas is not an easy task. In principle, these ideas improve elections verifiability and help to identify integrity problems. However, as we pointed out, security issues may arise. We leave tests to confirm them as future work.

Bibliography

[Ad08] Adida, B.: Helios: Web-based Open-Audit Voting. In: 17th USENIX Security Symposium. USENIX Association, pp. 335–348, 2008.

[Be87] Benaloh, J.: Verifiable Secret-Ballot Elections. PhD thesis, Yale University, 1987.

[Be06] Benaloh, J.: Simple Verifiable Elections. In: Electronic Voting Technology Workshop (EVT'06). USENIX Association, 2006.

[Be07] Benaloh, J.: Ballot Casting Assurance via Voter-Initiated Poll Station Auditing. In: Electronic Voting Technology Workshop (EVT'07). USENIX Association, 2007.

[Be12] Ben-Nun, J.; Farhi, N.; Llewellyn, M.; Riva, B.; Rosen, A.; Ta-Shma, A.; Wikström, D.: A new implementation of a dual (paper and cryptographic) voting system. In: 5th Int. Conf. on Electronic Voting 2012, EVOTE 2012. pp. 315–329, 2012.

[BN22] Benaloh, J.; Naehrig, M.: ElectionGuard Spec 1.1. Technical report, Microsoft Research, Redmond, WA, United States, 2022. `https://www.electionguard.vote/spec/`.

[Ch11] Chaum, D.; Florescu, A.; Nandi, M.; Popoveniuc, S.; Rubio, J.; Vora, P.; Zagórski, F.: Paperless Independently-Verifiable Voting. In: 3rd Int. Conf. on E-Voting and Identity (VoteID'11). volume 7187 of LNCS. Springer, pp. 140–157, 2011.

[CRS05] Chaum, D.; Ryan, P.; Schneider, S.: A Practical Voter-Verifiable Election Scheme. In: 10th European Symposium on Research in Computer Security (ESORICS). volume 3679 of LNCS. Springer, pp. 118–139, 2005.

[Ke10] Kelsey, J.; Regenscheid, A.; Moran, T.; Chaum, D.: Attacking Paper-Based E2E Voting Systems. In: Towards Trustworthy Elections, New Directions in Electronic Voting. volume 6000 of Lecture Notes in Computer Science. Springer, pp. 370–387, 2010.

[PH10] Popoveniuc, S.; Hosp, B.: An Introduction to PunchScan. In: Towards Trustworthy Elections, New Directions in Electronic Voting. Springer, pp. 242–259, 2010.

[Ri06] Rivest, R.: The ThreeBallot Voting System. Available at: `http://people.csail.mit.edu/rivest/Rivest-TheThreeBallotVotingSystem.pdf`, October 2006.

[Te21] Teague, V.: Which e-voting problems do we need to solve? Crypto Invited Talk, 2021.

Click and Cast: Assessing the Usability of Vote App

Laura Cristiano [1], Riccardo Longo [1], and Chiara Spadafora [2]

Abstract: This article provides a comprehensive examination, from the usability perspective, of Vote App, a coercion-resistant electronic voting system. We report on the results of the usability tests conducted, where participants had to cast a vote in an artificial election while using the anti-coercion mechanisms of Vote App. To evaluate the results, we follow the three usability metrics of effectiveness, efficiency and satisfaction.

Keywords: User-Centered Design, Usability Metrics, SUS, E-Voting

1 Introduction

In recent years, electronic voting (e-voting) systems emerged as a significant innovation for democracy. However, the success of an e-voting system depends on its usability - how easily, effectively and satisfactorily voters can use it.

In this article we report the results of a usability assessment of a mobile voting application called *Vote App* (see Sect. 3 and [Lo22, Bi23]), which was developed with the specific aim of guaranteeing coercion-resistance. Over 5 million Italians abroad currently have the right to a postal vote, with well-known logistic and security issues: coercion due to organized crime is a historically well-documented threat [De22]. Informally, a voting protocol is coercion-resistant if voters cannot prove whether or how they voted, even if they can interact with the adversary while voting. Vote App achieves this property using the mechanism of *fake credentials* [JCJ10, CGY22]. These credentials allow voters under the influence of a coercer to express their true votes while pretending to comply with the coercer's demands. When the voter is, or fears to be, subject to a coercion attack, they can autonomously create a ruse credential indistinguishable from a real one. This credential will not validate the corresponding ballot when votes are tallied.

To evaluate Vote App, we first conducted a pilot test before extending it to a larger event (see Sect. 4), allowing us to reach a broader sample. This approach offers several advantages, including the opportunity to refine the application based on the initial feedback. In this paper, we focus mainly on evaluating the usability of Vote App, rather than on user experience (UX), which was partially analyzed in [CS24]. As this work is ongoing, we plan to conduct iterative testing of both UX and usability in the future.

[1] Fondazione Bruno Kessler, Center for Cybersecurity, 38123 Povo, Trento, IT,
l.cristiano@fbk.eu, https://orcid.org/0000-0003-0895-3466;
rlongo@fbk.eu, https://orcid.org/0000-0002-8739-3091

[2] University of Trento, Department of Mathematics, 38123 Povo, Trento, IT,
chiara.spadafora@unitn.it, https://orcid.org/0000-0003-3352-9210

doi:10.18420/e-vote-id2024_05

Ethics and Data Protection. The tests were designed to maintain anonymity, focusing exclusively on gathering feedback regarding Vote App, without collecting personal data. The pilot was conducted within our Research Center, and the Data Protection Officer advised that a privacy consent form was unnecessary. The main test took place at a public event, and also in this context a privacy consent form was deemed unnecessary.

1.1 Research Objectives

This paper addresses the following research question (RQ): *Is an anti-coercion electronic voting system usable by the general public in terms of effectiveness, efficiency, and user satisfaction?*

To address this, following the ISO 9241-11 [I.O18] (see Sect. 1.2) definition of usability and its three metrics, we derive these three hypotheses:

- *Effectiveness Hypothesis (H1)*: Vote App will demonstrate a high level of effectiveness, calculated by the completeness and accuracy of the voting process without user errors.
- *Efficiency Hypothesis (H2)*: Vote App system will demonstrate high efficiency, reflected by minimal effort and time required by users to complete the voting process.
- *Satisfaction Hypothesis (H3)*: Users will report high satisfaction levels in using Vote App, as assessed by their subjective feedback on the overall voting experience.

1.2 Usability metrics

According to ISO 9241-11 [I.O18], usability is defined as the *effectiveness*, *efficiency* and *satisfaction* with which users accomplish specific objectives within particular environments. The National Institute of Standards and Technology (NIST) recommends these three metrics as suitable for assessing e-voting systems [La04].

Effectiveness is defined as the completeness and accuracy with which a user can achieve a specific task [Ma19]. In electronic voting, effectiveness is the measure of voters' ability to successfully cast their vote for their intended candidate without encountering errors [Bu16, GBE06]. To determine effectiveness, the examiner can observe participants while they perform the task, use visual recording, or ask participants to self-report their progress. Additionally, Thinking Aloud [Ni12] can be employed. This method allows participants to vocalize their thoughts as they engage with the system [Ni12]. Error rates are also an effective way for verifying effectiveness, especially in the case of e-voting, as they are linked to the voter's intention and the actual outcome [Ma19].

Efficiency assesses whether participants achieve their goals without using excessive resources, striking a balance between effort and time [Ma19, GBE06]. For e-voting, it is represented by the time voters spend casting their vote and the time required by the user to complete verification [Ma19, Bu16].

Satisfaction refers to how content and at ease users feel during their interaction with a system or process [Bu16, GBE06]. Satisfaction is the only subjective usability metric recommended by NIST. In e-voting, satisfaction reflects how pleased or fulfilled voters feel while participating in the election [Bu16]. Satisfaction can be measured with standard methods like the System Usability Scale (SUS) questionnaire [GBE06] or it can also be measured using a non-standardized questionnaire developed by the examiner, tailored to the specific purpose of the study [Ma19].

The SUS [GBE06] is a 10-item survey designed to collect subjective usability assessments from participants in a usability test, evaluating aspects like efficiency, intuitiveness, ease of use, and satisfaction. Participants express their agreement or disagreement with each statement using a five-point Likert scale: *strongly disagree* (1), *disagree* (2), *neutral* (3), *agree* (4) and *strongly agree* (5) [Ha20].

1.2.1 Sections Overview

In Sect. 2, we discuss related works concerning usability evaluation, providing an overview of existing studies. Sect. 3 presents briefly Vote App. Sect. 4 outlines the methodology we employed to conduct our usability tests. In Sect. 5 we analyze the results of the tests. Finally, in Sect. 6, we conclude by summarizing the lessons learned from our study and discussing potential directions for future research and development.

2 Related Work

Given the sensitivity of e-voting systems, it is particularly crucial to emphasize the importance of conducting usability testing with potential users [Zo19, He06] and to assess the usability of the system [Ac22, Di19, FR12, Ku17, Ma21]. As highlighted by [YM07], voters' perception of voting technology significantly influences their intention to use it. Additionally, usability issues, as noted by [Ma20], have the potential to lead to incorrect election outcomes. Acemyan et al. [Ac14] compare the usability of three voting systems: Helios, Prêt à Voter, and Scantegrity II. Their evaluation reveals that only 58% of the participants were able to complete the voting process, with even lower completion rates observed during the verification phase. These usability findings underline the importance of not only guaranteeing the security of a system but also prioritizing usability. In [AM16] the authors assert that ensuring system usability is further complicated by the infrequency of elections. Voters are expected to cast their votes with nearly 100% success rates, despite often having no prior experience or training with the voting system.

In the study conducted by [Ma19], it is emphasized that human factors should be taken into account early in the design process of e-voting systems. In the article, the authors provide guidelines for assessing usability metrics of e-voting solutions.

3 The Tested E-Voting System

3.1 Description of Vote App

In this Section, we give a brief overview of the functionalities of Vote App (see Fig. 1). For more information about the protocol, the cryptographic primitives used and the implementation details, we refer the reader to [Bi23, Lo22].

Voters[3] authenticate themselves via their national digital identity to obtain a voting credential [JCJ10] from the responsible authorities in order to cast ballots. These credentials allow voters under the influence of a coercer to express their true votes while pretending to comply with the coercer's demands. When the voter is, or fears to be, subject to a coercion attack, they can autonomously create a ruse credential indistinguishable from a real one. This credential will not validate the corresponding ballot when votes are tallied. To enhance usability of the scheme, in [Bi23, Lo22] the credential is given to the voter in the form of a five-digit PIN. When inserted during the voting phase, this PIN unlocks the valid credential needed to cast a valid vote. To create a ruse credential, it is sufficient to set up a ruse PIN. The correctness of the PIN can be verified via a Designated Verifier Non-Interactive Zero-Knowledge Proof (DVNIZKP), which proves the correctness of the associated credential. To enhance the security of the system, $\mathcal{V}$ can also use an external verification service, which ideally should be a pre-installed application with no internet connection, to check the correctness of the data managed by Vote App. For coercion resistance, we assume that the voter has control over their voting device. This assumption is valid since Vote App can be used from any device. Furthermore, everyone can access a Web Bulletin Board (WBB), from any device, in which all the public data regarding the election is published.

The voting protocol can be divided into three steps.

3.1.1 Setup and Registration

The voter $\mathcal{V}$ downloads Vote App from an official application store and installs it on their device. Once Vote App is installed, $\mathcal{V}$ can start the registration process:

- $\mathcal{V}$ opens Vote App and reads the high-level information displayed.
- $\mathcal{V}$ logs in with their digital identity.

[3] The target population of Vote App is composed by Italian citizens eligible to vote in Italian elections.

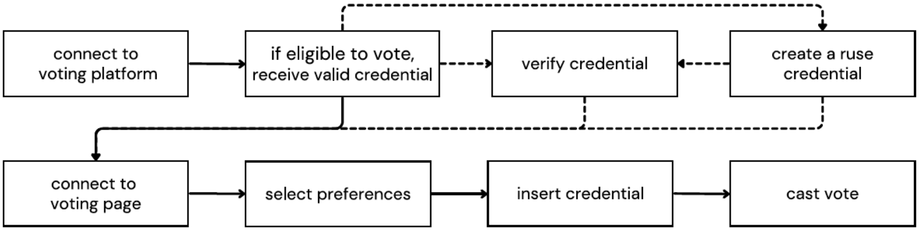


Fig. 1: User Flow of Vote App with and without coercion.

- If the login is successful, Vote App displays to $\mathcal{V}$ the procedure to retrieve the PIN and to activate Vote App from another device.
- After a random waiting period, $\mathcal{V}$ receives a push notification alerting them that the PIN is ready to be retrieved.
- $\mathcal{V}$ logs in to Vote App and receives their PIN along with its visual representation, encoded as a string of emojis called "Private PIN Emojis". $\mathcal{V}$ is warned to save both the PIN and its visual representation.
- After a random waiting period, $\mathcal{V}$ receives a push notification alerting them that from now on they can verify, via the DVNIZKP, as many times as needed, the correctness of the PIN.
- At any time, $\mathcal{V}$ can use the external verification service to check the correctness of the emoji string composing "Private PIN Emojis".

3.1.2 PIN Management

In order to resist coercion, $\mathcal{V}$ can set up one (or more) ruse PIN with which they can simulate a vote[4]. Moreover, a forged verification proof is made available to Vote App, so that the voter can deceive the coercer by pretending that this ruse PIN is a valid one. The procedure is the following:

- $\mathcal{V}$ opens Vote App and goes to the "app management" screen, in the tab "ruse PIN".
- $\mathcal{V}$ types in the ruse PIN they want to set up.
- After a random waiting period, $\mathcal{V}$ receives a push notification alerting them that the ruse PIN is ready to be retrieved. Note that this notification is exactly the same as the

[4] During the voting phase, $\mathcal{V}$ can simulate a vote that will not be counted just by casting a vote with a random PIN. However this does not guarantee coercion-resistance, since, by doing this, they cannot produce any proof of PIN reception or verification.

one sent during Registration, but in this case the ruse PIN is retrieved instead of the real one.

- $\mathcal{V}$ logs in to Vote App and receives their (ruse) PIN along with its visual representation, encoded as a string of emojis called "Private PIN Emojis". $\mathcal{V}$ is warned to save both the PIN and its visual representation. This screen is identical to the one $\mathcal{V}$ has seen during the registration step.
- After a random waiting period, $\mathcal{V}$ receives a push notification alerting them that from now on they can verify the PIN.

Differently from the previous step, in which the PIN that verifies the proof is the valid one, after the ruse PIN request, this proof is verified only by the ruse PIN. If $\mathcal{V}$ inserts the valid PIN, the verification will not be not successful. Nevertheless, the valid PIN remains the only one that allows to cast a valid vote.

In this step $\mathcal{V}$ can request a reminder of their valid PIN. In this case the procedure is as follows:

- $\mathcal{V}$ opens the App and goes to the "app management" screen, in the tab "PIN reminder".
- $\mathcal{V}$ confirms their choice.
- The issuance of the PIN follows the same procedure as in the previous cases. From now on, until a new ruse PIN request is made, only the valid PIN will verify the DVNIZKP.

3.1.3 Ballot Casting and Individual Verification

When the voting period starts, $\mathcal{V}$ can cast a vote with the following procedure:

- $\mathcal{V}$ opens Vote App and from its Homepage clicks on the "Vote" button.
- $\mathcal{V}$ expresses and confirms their preference.
- Vote App advises $\mathcal{V}$ that their preference is being encrypted.
- Vote App shows to $\mathcal{V}$ a string of emojis called "Ballot Emojis" that will be needed in the verification phase and asks $\mathcal{V}$ to enter a PIN.
- After PIN insertion, Vote App advises $\mathcal{V}$ that their vote is being sent, while also showing the "Private PIN Emojis". If the sequence differs from the expected one, $\mathcal{V}$ can report a problem via the "Report a Problem" button.
- $\mathcal{V}$ chooses one out of two values, called *control numbers*, shown by Vote App in a confirmation page to confirm their vote.
- Vote App displays a confirmation message to $\mathcal{V}$ that the voting has been cast and received.

- Vote App displays the hash of the vote and all the information needed by $\mathcal{V}$ to check that their vote has been correctly registered and that it has not been tampered with.
- Using these information, $\mathcal{V}$ connects to the WBB and checks that:
  - their vote has been correctly registered and confirmed, by checking that the hash given by Vote App is included in the list of registered hashes, along with a tick acknowledging correct confirmation.
  - their vote has not been tampered with by checking that the control number displayed is the same as the one previously chosen.

Vote App allows for re-voting: only the last vote cast, per PIN, will be kept and, if cast with the valid PIN, counted.

When the voting period ends, the WBB shows to every user the final tally. and everyone can download the proofs to check its correctness.

Almost every interface of Vote App includes a magnifying lens icon with more detailed information about the current step. To enhance usability, $\mathcal{V}$ can always access the User Manual of Vote App and, whenever $\mathcal{V}$ is advised to store some data, a "Share button" is included near the data to be saved.

4 Study design and procedure

Relying on the usability metrics outlined in Sect. 1.2, we conducted a pilot test and a main test, where the participants were asked to cast a vote in a fictional election using Vote App.

To evaluate the *effectiveness*, we assessed the task performance by counting how many participants successfully completed the task and the number of attempts required.

Additionally, we encouraged participants to share their thoughts verbally during the test following the *Thinking Aloud* method [Ma19].

For measuring *efficiency*, we recorded the time participants spent on each task, with a special focus on ballot completion.

To evaluate *satisfaction*, we administered the System Usability Scale (SUS) questionnaire at the end of each test session, tailored to our e-voting context (e.g., *"I think that I would like to use this system frequently"* became *"I think that I would use Vote App to vote in a real election"*). The final SUS score ranges from 0 to 100, with a score above 68 considered good [Des]. We decided to use the SUS questionnaire because it is quick and easy to administer (taking only a few minutes to complete), and it has been widely used in e-voting user studies [Ma19].

As suggested by [Ma19], during the pilot test, we also asked participants four additional questions to gain deeper insights into their experience with Vote App.

The questions were the following: (i) *In your opinion, how was the voting process?*; (ii) *What do you think about the PIN feature?*; (iii) *Did you notice the presence of the emojis during the voting process? What do you think about them?*; (iv) *What is your final opinion about Vote App?*.

4.1 Context

We conducted the pilot test inside a Research Center for Cybersecurity. The Center's objective is to enhance cyber risk management, with a particular focus on digital identity and the quality of online services. The test was conducted by two cryptographers with mathematical background and a UX/UI expert with sociological background.

The main test was conducted during 2023 European Researchers' Night, an EU initiative launched in 2005, that aims to bridge the gap between science and the general public. The event spanned from 5 pm to midnight, providing enough time to conduct multiple test sessions. Two information security and mobile security experts, and a jurist joined the group conducting the test. The different backgrounds allowed us to discuss the complexities of e-voting from various angles and dispel any possible doubts.

4.2 Test organization

During each test session, participants had to complete a task, specifically to cast a vote in a fictional election. After completing the voting process, participants were invited to check that their vote was recorded and to verify the accuracy of the information received through Vote App. These features were available in the main test through the Verification Service and the WBB (see Sect. 3), but omitted during the pilot due to ongoing development work. Also, in order to mitigate the lack of information experienced during the pilot test, for the main test we prepared two explanatory posters. The first one provided a simplified explanation of the cryptography behind Vote App, specifically focusing on the creation and verification of voting credentials, the second one contained a step-by-step guide of Vote App.

4.2.1 Fictional Election

In the pilot test, users were asked to vote for their preferred city in Italy to host EXPO 2030. Initially, participants were prompted to select a list and subsequently choose a candidate from their chosen list. The lists were: *Northern Italy*, *Central Italy* and *Southern Italy and Islands*. The candidate cities were: (i) for Northern Italy: *Trento, Milan, Turin, Venice*; (ii) for Central Italy: *Rome, Florence, Ancona, Perugia*; and (iii) for Southern Italy and Islands: *Naples, Bari, Tropea, Palermo*.

In the main test, to speed up the voting process, the election was a referendum. Six different questions were created for participants to choose from (e.g., Q: “Do you prefer cream-flavored or fruit-flavored ice cream?”).

4.2.2 Scenario

For the pilot test, we designed two distinct scenarios: one coercion-free and one with a coercer[5]. Six participants were pre-assigned to the coercion-free scenario, while the remaining four were assigned to the scenario with the coercer. In the scenario with the coercer, participants were approached by a fictional criminal who demanded that they vote for a specific city. The criminal’s motivation was financial gain, as he and his gang stood to profit substantially if the chosen city emerged victorious in the election. Participants were required to initially cast a vote for the chosen city using the anti-coercion functionalities of Vote App, thereby sending an invalid vote. Subsequently, they could send a valid vote for their preferred city. This scenario was intentionally crafted to evaluate the effectiveness of the anti-coercion features integrated into Vote App.

In the main test, scenarios were not included to allow participation by multiple individuals simultaneously in a noisy environment. Each voter was given complete freedom to build their own scenario, choosing independently how many times to vote, whether to verify the PIN, and whether to set and vote with a ruse PIN. Almost 75% of the participants tested the anti-coercion functionalities.

4.2.3 User Authentication

The focus of the test was not on the authentication, so this step was simulated in the tests. In the pilot, participants had to press on the button *Login with digital identity* but, instead of inserting their credentials, they were automatically logged in. In the main test, authentication was necessary in order to correctly assign voting credentials, thus voters were required to choose a pseudonymous identity from a pre-generated set of options. Each identity was represented by a drawn portrait of an animal, blatantly fictional first and last names, and a code for logging into Vote App.

[5] The coercer is an attacker who tries to influence or force the voter to follow their demands. Possible demands range from voting for a specific candidate or not voting at all.

4.3 Test session

4.3.1 Participants and Recruitment

For the pilot, participants were recruited via email. We sent invitations to all Italian individuals working within the Center eligible for voting in Italian elections. We decided to exclusively involve Italian citizens. The reason behind this decision was twofold: firstly, Vote App was developed in Italian, and secondly, they possess a familiarity with the electoral process in Italy, which was what we wanted to test. The group taking part in the test was composed of 10 people with expertise that ranges from information technology and cybersecurity to cryptography. Approximately 30% possessed a Bachelor's or Master's degree, while 70% also held a PhD. In terms of age distribution, roughly 30% of the sample fell between the birth years of 1975 and 1990, and the remaining 70% were born between 1991 and 2001. It is important to acknowledge the potential for biases as their perspectives and experiences might not fully represent the broader spectrum of voters' capabilities.

For the main test, due to the nature of the event, participants were not formally recruited. Instead, we approached individuals who came to our stand and invited them to try the application and provide feedback.

We were able to gain insights into the participant demographics through the official data obtained from the event organizers[6]. Approximately 25% did not hold a university degree, while 50% possessed a Bachelor's or Master's degree. Approximately 7% of participants held a PhD, with the remaining 18% having an educational qualification below the level of an middle-school diploma. In terms of age distribution, roughly 50% of the sample fell between the birth years of 1997 and 2003, 25% were born between 1991 and 1997, 13% were born between 2003 and 2009, and the remaining 12% were born between 1961 and 1991. Regarding employment status, the 67% of participants were students, while the remaining 33% were engaged in permanent employment (without further classification of occupation types). Approximately, 80 people participated in the main test.

4.3.2 Equipment.

The tests have been conducted using Android smartphones provided by the Center, with Vote App already installed and opened on the login page. To facilitate note-taking, we gave each participant a writing support.

For the main test, two desktops were set up to ensure visibility of the WBB. Additionally, we hung the previously prepared posters near our stand, so that they were easily accessible for reference in case of any doubts. The list of potential referendum questions was printed

[6] For confidentiality reasons, this data was not publicly available but was provided to us by the organizers exclusively for the purpose of this article.

on a card accessible by all. For the pseudonymous identities, we adopted a similar approach by printing them on paperboard and folding them in half, creating a resemblance to real ID cards. We even included a QR code on the back of the identity cards, which directed participants to a web page where the posters were uploaded.

4.3.3 Location.

The pilot test was conducted in one of the meeting rooms of the Center. The environment was quiet and the participants had no external distractions. For the main test, our stand was clearly marked with signage. The environment was busy and noisy, with people exploring the exhibition stands and wondering around.

4.3.4 Test.

We conducted multiple test sessions. In the pilot, each session involved only one participant, with approximate length between 20 to 30 minutes. In the main test, each session involved from 3 to 5 participants, with approximate length between 10 to 15 minutes.

1. *Introduction.* We offered the participant a brief overview of the project. Subsequently, we presented the test instructions and we recommended to maintain a good flow of comments while interacting with the system, adhering to the Thinking Aloud (see Sect. 1.2).
2. *Election Context.* In the pilot we read and explained the assigned scenario to the participant, giving them a printed copy for reference. In the main test there was no scenario, instead we gave to the participating group a list of referendum questions, from which they could choose which one to vote on.
3. *Authentication.* Each participant simulated the authentication step as explained before.
4. *PIN reception.* Vote App provided to each participant a PIN which they could write down.
5. *Voting.* Each participant accessed the ballot and cast their vote. In the main test, they were also able to view the registration of the vote on the WBB.
6. *Re-Voting, Ruse PIN and Verification.* Participants could set up a ruse PIN, re-vote with any PIN, verify a PIN. Only in the coercion scenario during the pilot re-voting with a ruse PIN was mandatory.
7. *Tabulation.* In the main test, upon completion of the voting phase by all the participants, we started the tabulation and displayed the results on the WBB.

8. *Post-Test Questionnaire.* At the end of each test session, we asked the participant to fill out the System Usability Scale (SUS) questionnaire. In the pilot this was mandatory, and we also asked participants to answer four additional questions (see Sect. 4).

5 Results evaluation

After both tests, we performed an in-depth analysis of the data gathered. This analysis included the evaluation of the three usability metrics established by ISO 9241-11 [I.O18]. Additionally, we reviewed the feedback provided by participants during their interactions with Vote App, aiming to gain valuable insights into their user experience and identify areas for improvement.

5.1 Pilot test

5.1.1 Quantitative results

Effectiveness (H1). In both scenarios, every test participant successfully completed the required task. In the coercion-free scenario, each test participant accomplished the required task on the first attempt and without any errors. Conversely, in the scenario with coercion, only two participants managed to complete the task on their first attempt; the others required at least two more attempts to succeed. As suggested in [Ma19], the Thinking Aloud method is not really reliable for determining effectiveness, but it can provide valuable insights into participants' feelings. Therefore, we collected users' comments while interacting with the prototype in the qualitative results. In terms of error rates, participants in the coercion-free scenario completed the task without any errors. Conversely, only half of the participants in the scenario with coercion managed to complete the task. The first error arose from a misunderstanding of the coercer's request (i.e., the participant did not understand that they had to vote with the ruse PIN for the forced choice to evade the coercer's request, see Sect. 4.2), while in the other case, the participant entered a random PIN instead of the ruse one, demonstrating a lack of understanding of the distinction.

Efficiency (H2). Due to some issues encountered with the (then still under development) prototype of Vote App, it was impossible to collect data on the time users spent on the task. We noted, however, that users in the scenario with coercion required more time to both comprehend and complete the task.

Satisfaction (H3).

The total (average) score of the SUS questionnaire (see Sect. 1.2), in the coercion-free scenario, was 82.9/100, ranging from 60/100 as the lowest to 95/100 as the highest (see Fig. 2). Conversely, in the scenario with coercion, the total score was of 69.3/100, with

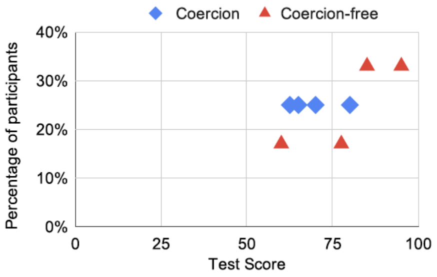


Fig. 2: SUS Results of the Pilot Test.

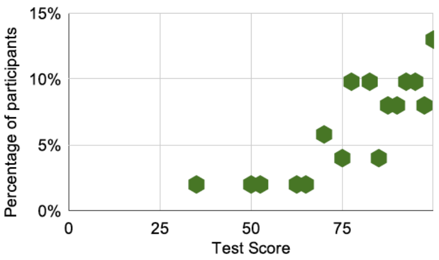


Fig. 3: SUS Results for the Main Test.

scores ranging from 62.5 to 80 out of 100[7]. From these results, it is clear that participants in the scenario with coercion provided lower scores (see Fig. 2), which was also highlighted by the feedback gathered during the test. This could be attributed to either an ineffective presentation of the task, or to an ineffective app design concerning the anti-coercion features.

5.1.2 Qualitative Results

The qualitative data presented in this subsection has been acquired during each test session by employing the Thinking Aloud method, as well as through participant's answers to the questions we posed at the conclusion of the test (see Sect. 4). Additionally, we carefully observed how participants behaved while interacting with Vote App, to obtain more insights on the app.

Here we summarize our analysis on the collected comments, observations and behaviours.

Positive feedback. The voting process was perceived as straightforward and efficient by all participants in both scenarios, who expressed satisfaction with the initial page providing a concise overview of the app. They also appreciated the clarity of the voting instructions and the option to consult electoral programs. Regarding the PIN, participants liked its brevity (only 5 digits) and found the PIN recovery and verification features to be highly beneficial and practical. Moreover, the setup mode for the ruse PIN proved to be intuitive. Participants highly valued the possibility to capture images and screenshots of Vote App. Although there were some negative comments regarding the usage of emojis, many participants verified that the "Private PIN Emojis" displayed alongside the PIN at the beginning of the protocol matched those shown during the voting process.

[7] Note that the lowest score is considered good, see [Des].

Negative feedback. After an in-depth analysis of the qualitative data collected, it became evident that the majority of negative feedback came from a lack of comprehensive explanations regarding the key features of Vote App. Initially, during the design phase of the test, we chose not to include extensive descriptions of the app in order to prevent the test from becoming too difficult. Given that this application requires complex functionalities that diverge significantly from standard apps, this decision ultimately proved ineffective. For example, it was not evident to participants that Vote App assigns a unique, valid PIN for each user, which remains unchanged. Additionally, they were unaware that they could vote multiple times during the voting process. The purpose of the ruse PIN functionality was also not fully understood, as some questioned the necessity of setting a ruse PIN to submit an invalid vote when they could simply enter a random one. Furthermore, when participants did use a ruse PIN, they often chose easily guessable combinations, such as "12345".

The emojis created confusion among some participants. Users that were familiar with emojis being used as a security measure, such as on platforms like Telegram, the messaging app, found this feature helpful and did not experience interface confusion. On the other hand, participants accustomed to encountering emojis primarily in messaging contexts found it challenging to understand their purpose within Vote App. Some perceived them as a distraction or mistook them for application errors, and a few remarked that they were difficult to remember. Recognizing this issue, a participant suggested to add the option of entering the PIN twice. This proposed solution aims to provide an additional layer of assurance for users who may not pay close attention to emojis or struggle to comprehend their function. By allowing users to input their PIN twice, they would have greater control and confidence in ensuring the accuracy of their PIN entry. Another feature that caused confusion were the control numbers. In fact, upon encountering them, many participants wondered, *"What am I supposed to do now?"* or *"Why am I being asked to select these numbers?"*, which strengthens our initial thesis about a lack of app explanation.

Other suggestions. We also collected some suggestions regarding how to improve the user interface design of Vote App. During the voting process, it was not clear when and where the emojis will appear and a participant suggested to introduce an element signaling their arrival. Another feedback we received pertains to text descriptions: it would be best to keep them as neutral as possible, avoiding exclamation marks and words like "attention" or "remember".

Post Test Improvements. Based on our analysis on the pilot conducted, we have made several updates to improve user experience and security. These include modifying interfaces to clarify PIN and emoji information and providing more explicit instructions for setting a non-trivial ruse PIN.

5.2 Main test

5.2.1 Quantitative Results

Effectiveness (H1). All the participants managed to successfully complete the voting process and send the vote to the WBB. Regarding the other features of the app (e.g., set up a ruse PIN, verify the PIN) we were not able to collect all the data as we did in the pilot due to the test context. Nevertheless, from the partial data gathered, we can state that participants were able to understand and use the verification service and to handle the ruse PIN request with lower error rates with respect to the pilot.

Efficiency (H2). All the participants managed to complete the test in 10 minutes, which was the time we expected for each test session to last.

Satisfaction (H3). In total, 50 out of the 80 participants in our demo compiled the SUS questionnaire. The total score was 84.9/100, with scores ranging from 35 to 100 out of 100 (see Fig. 3).

5.2.2 Qualitative Results

As for the pilot test, we present a brief summary of the qualitative data acquired via the Thinking Aloud method and our observations on the participants behaviours.

Positive Feedback. Based on participants' behavior, we observed that everyone wrote down their PIN, indicating the importance they attached to it, and the difficulty of remembering it. When we asked the participants who tested the anti-coercion functionalities to provide feedback on the ruse PIN, they did not have any negative or positive feedback. This suggests that we made a step towards improving explanations on Vote App, by communicating more efficiently the anti-coercion functionalities. Moreover multiple participants expressed positive feedback on emojis, stating that they were an interesting introduction and did not cause confusion. Moreover, their use in the voting process was well understood. The WBB introduction was well-received and participants found it to be useful and easy to use.

Negative Feedback. Some participants did not understand that the PIN could be retrieved if forgotten. Another participant mentioned that it was not clear that the PIN had to be entered each time voting. In terms of the use of emojis in general, one participant found them confusing since they were designed differently between the two different devices (i.e., the smartphone and the screen where the WBB was displayed). Regarding control numbers, participants thought that they were not well-explained and that it was unclear what they were for. Participants also questioned why they had to select one value, instead of letting the app doing it automatically.

Other suggestions. Some participants were confused by the usage of emojis to visualize the PIN and believed that the PIN was represented by them. The reason behind that is because the emojis of “Private PIN Emojis” were centrally placed on the PIN reception interface.

6 Conclusions

The goal of this study was to evaluate the usability of our proposed design for a coercion-resistant electronic voting system (RQ). The results obtained in this study highlight the importance of having an intuitive and user-centric design in an e-voting application. By providing clear instructions and ensuring accessibility, voters were able to make informed decisions and actively participate in the voting process. Moreover, the ability to view the registration of their vote on the WBB is an important step towards having a transparent voting procedure. The optional questions we posed to the willing voters, allowed us to gather valuable insights, further improving the overall usability.

The most evident result from our evaluation is the need to balance the amount of information being provided to the users in each interface. While some participants felt the explanations of the functionalities to be overwhelming, others found them useful and essential for a voting application, even if long. This shows that we still have a lot of work to do to find right balance, ensuring users can comprehend all aspects of the app without giving them excessive information. Regarding our initial research hypotheses, the results showed that participants successfully completed the voting task on their first attempt, although difficulties arose in the scenario with coercion during the pilot and insufficient feedback was collected in the main test regarding the anti-coercion features (H1). Both the pilot and main test demonstrated high efficiency, with participants completing the voting process within a few minutes (H2), and high satisfaction levels, with participants appreciating the system’s ease of use and security features (H3). This approach will help alleviate confusion and improve user experience, ultimately enhancing the effectiveness and trustworthiness of the voting system. As future works, we intend to address critical aspects of both Vote App and this Usability tests. Our study primarily involved students, which may limit the representativeness of our findings. Future research should expand the sample to include a more diverse population, such as different age groups, professions, and backgrounds. While our exploratory study lays the groundwork, it is difficult to properly assess a user reaction in a situation which involves fear. We plan to develop further tests to assess the anti-coercion functionality and to verify its effectiveness and reliability. Moreover we plan to provide more concise and user-friendly instructions to voters.

6.0.1 Acknowledgment

This work has been partially supported by a joint laboratory between Fondazione Bruno Kessler and the Italian Government Printing Office and Mint and by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

Bibliography

[Ac14] Acemyan, Claudia Z.; Kortum, Philip; Byrne, Michael D; Wallach, Dan S.: Usability of Voter Verifiable, End-to-end Voting Systems: Baseline Data for Helios, Prêt à Voter, and Scantegrity II. USENIX Journal of Election Technology and Systems (JETS), pp. 26–56, August 2014.

[Ac22] Acemyan, Claudia Ziegler; Kortum, Philip; Byrne, Michael D; Wallach, Dan 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. 866–889, 2022.

[AM16] Ali, Syed Taha; Murray, Judy: An Overview of End-to-End Verifiable Voting Systems. Real-World Electronic Voting: Design, Analysis and Deployment, pp. 171–218, 2016.

[Bi23] Bitussi, Matteo; Longo, Riccardo; Marino, Francesco Antonio; Morelli, Umberto; Sharif, Amir; Spadafora, Chiara; Tomasi, Alessandro: Coercion-resistant i-voting with short PIN and OAuth 2.0. In: E-Vote-ID 2023. Lecture Notes in Informatics (LNI). Gesellschaft für Informatik, Bonn, 2023. to appear.

[Bu16] Budurushi, Jurlind; Renaud, Karen; Volkamer, Melanie; Woide, Marcel: An investigation into the usability of electronic voting systems for complex elections. Annals of Telecommunications, 71, 04 2016.

[CGY22] Cortier, Véronique; Gaudry, Pierrick; Yang, Quentin: Is the JCJ voting system really coercion-resistant? Cryptology ePrint Archive, Paper 2022/430, 2022.

[CS24] Cristiano, Laura; Spadafora, Chiara: Enhancing Usability in E-Voting Systems: Balancing Security and Human Factors with the HC3 Framework. In: International Conference on Human-Computer Interaction. Springer, pp. 33–42, 2024.

[De22] Desantis, Vincenzo: Il voto degli italiani all'estero: nuove criticità e vecchi problemi nella prospettiva del superamento del voto per corrispondenza. Federalismi.it, 22:31–51, 2022.

[Des] Designer Italia 2023. Manuale Operativo di Design: test di usabilità.

[Di19] Distler, Verena; Zollinger, Marie-Laure; Lallemand, Carine; Roenne, Peter B.; Ryan, Peter Y. A.; Koenig, Vincent: Security - Visible, Yet Unseen? In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. CHI '19, Association for Computing Machinery, New York, NY, USA, p. 1–13, 2019.

[FR12] Fuglerud, Kristin Skeide; RØssvoll, Till Halbach: An evaluation of web-based voting usability and accessibility. Univers. Access Inf. Soc., 11(4):359–373, nov 2012.

[GBE06] Greene, Kristen K.; Byrne, Michael D.; Everett, Sarah P.: A Comparison of Usability Between Voting Methods. In: USENIX Workshop on Accurate Electronic Voting Technology. 2006.

[Ha20] Hao, Feng; Wang, Shen; Bag, Samiran; Procter, Rob; Shahandashti, Siamak F.; Mehrnezhad, Maryam; Toreini, Ehsan; Metere, Roberto; Liu, Lana Y.J.: End-to-End Verifiable E-Voting Trial for Polling Station Voting. IEEE Security & Privacy, 18(6):6–13, 2020.

[He06] Herrnson, Paul. S; Niemi, Richard G.; Hanmer, Michael J.; Bederson, Benjamin B.; Conrad, Frederick G.; Traugott, Michael: The Importance of Usability Testing of Voting Systems. In: 2006 USENIX/ACCURATE Electronic Voting Technology Workshop (EVT 06). USENIX Association, Vancouver, B.C., August 2006.

[I.O18] International Organization for Standardization. ISO 9241-11:2018, Ergonomics of human-system interaction. Part 11: Usability: Definitions and Concepts, 2018.

[JCJ10] Juels, Ari; Catalano, Dario; Jakobsson, Markus: Coercion-resistant electronic elections. In: Towards Trustworthy Elections. Springer, 2010.

[Ku17] Kulyk, Oksana; Neumann, Stephan; Budurushi, Jurlind; Volkamer, Melanie: Nothing Comes for Free: How Much Usability Can You Sacrifice for Security? IEEE Security & Privacy, PP:1–1, 06 2017.

[La04] Laskowski, Sharon; Yen, James; Autry, M; Cugini, John; Killam, W: Improving the Usability and Accessibility of Voting Systems and Products, 2004-04-01 2004.

[Lo22] Longo, Riccardo; Morelli, Umberto; Spadafora, Chiara; Tomasi, Alessandro: Adaptation of an i-voting scheme to Italian Elections for Citizens Abroad. In: E-Vote-ID 2022. 10 2022.

[Ma19] Marky, Karola; Zollinger, Marie-Laure; Funk, Markus; Ryan, Peter Y. A.; Mühlhäuser, M.: How to Assess the Usability Metrics of E-Voting Schemes. In: Financial Cryptography Workshops. 2019.

[Ma20] Marky, Karola; Zimmermann, Verena; Funk, Markus; Daubert, Jörg; Bleck, Kira; Mühlhäuser, Max: Improving the usability and ux of the swiss internet voting interface. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. pp. 1–13, 2020.

[Ma21] Marky, Karola; Zollinger, Marie-Laure; Roenne, Peter; Ryan, Peter Y. A.; Grube, Tim; Kunze, Kai: Investigating Usability and User Experience of Individually Verifiable Internet Voting Schemes. ACM Trans. Comput.-Hum. Interact., 28(5), sep 2021.

[Ni12] Nielsen, Jakob: Thinking Aloud: The #1 Usability Tool, 2012.

[YM07] Yao, Yurong; Murphy, Lisa: Remote electronic voting systems: An exploration of voters' perceptions and intention to use. EJIS, 16:106–120, 04 2007.

[Zo19] Zollinger, Marie-Laure; Distler, Verena; Rønne, Peter; Ryan, Peter; Lallemand, Carine; Koenig, Vincent: User Experience Design for E-Voting: How mental models align with security mechanisms. E-Vote-ID 2019 TalTech Proceedings, 2019.

Design and Evaluation of Verifiable Voting Systems Based on Tracking Code Verification

Christina Nissen[1], Oksana Kulyk[1], Melanie Volkamer[2], Lara Elisabeth Fredrich[3] und Helena Hermansen[3]

Abstract: To mitigate security risks of Internet voting, techniques for verifiability have been developed, allowing the voters to verify that their cast vote has not been manipulated. One such technique is the use of tracking codes, which does not rely on complex cryptographic mechanisms, and therefore is often assumed to be more intuitive for the voters. However, no systematic evaluation of the usability, verification efficacy, and perceived trustworthiness of these systems has yet been conducted. Our contribution evaluates two variants of a tracking code-based system in a user study ($N = 306$), testing both of the variants in the absence of vote manipulations as well as using different simulated tactics of vote manipulation. We conclude that both of our proposed variants are perceived as easy to use, transparent, and trustworthy in the absence of vote manipulation. However, we found varying rates of verification efficacy based on the vote manipulation tactic, manipulation with detection rates ranging from 0% to 76%. We conclude that using tracking codes can be a viable approach. At the same time, more awareness of potential manipulations needs to be raised to achieve the required security level, irrespective of the verification technique in place.

Keywords: tracking code-based verification, usability, manipulation detection, end-to-end verifiability, user study

1 Introduction

The concept of end-to-end verifiability has been introduced to reduce risks of large-scale vote manipulations in internet voting and has been accepted as the gold standard in designing secure voting systems. As such, a number of techniques for *cast-as-intended*, *recorded-as-cast* and *tallied-as-stored* verifiability has been proposed, with *cast-as-intended* techniques presenting a particular challenge from the usability point of view. As such, since many of them rely on complex cryptographic techniques and non-intuitive procedures, this can lead to voters being unable to properly apply the verification steps [KV18].

One of these cast-as-intended techniques relies on the use of so called *tracking codes* [Kü16a; RRI15; RRR21]. During vote casting, a tracking code is either automatically assigned to voters by the voting system or (fully or partially) self-chosen by the voter is assigned. The tracking code is then stored next to the voter's cast vote. At the end of the election, the cast (plain text) votes are published together with their associated tracking codes, so that the voter can verify that the vote next to their code is indeed their intended choice.

[1] IT University of Copenhagen, Denmark, chfn@itu.dk; okku@itu.dk
[2] Karlsruhe Instutute for Technology, Germany, melanie.volkamer@kit.edu
[3] No affiliation, elisabeth@elisabethfredrich.eu; helena.hermansen@outlook.com

doi:10.18420/e-vote-id2024_06

Due to the simplicity of the verification procedure, verification using tracking codes has been previously suggested as a promising way to implement voter verification in a usable way [Kü16b; Ma22]. It has furthermore been used in real-world elections, in particular, in the internal party elections of the Christian Democratic Union (CDU) party in Germany in 2020 to decide whom the party should nominate as chancellor for the next general parliamentary elections [Be21]. However, no systematic evaluation of these systems and effectiveness of their verification techniques has been conducted yet. Our contribution in this work is therefore the answer to the following research questions:

RQ1 How effectively can voters detect various types of simulated manipulations in tracking code-based systems?

RQ2 How usable and trusted are the tracking code-based systems perceived?

Answering these research questions, we conduct an online study ($N = 306$). In particular, to study **RQ1**, we simulate three different types of vote manipulations, relying on the approach from previous research [Vo22a]. Our study reveals that the detection rate of manipulated votes differs depending on the manipulation tactic chosen by the adversary, with 76% of voters being able to detect manipulation in the simplest tactic, yet none of the voters detecting the manipulation with the most complex one. For studying **RQ2**, we test both of the systems without presence of manipulations. We use the System Usability Scale [BKM09] to evaluate the usability of the system and the Trust in Voting Systems (TVS) [AKO22] questionnaire to evaluate the perceived trust in both systems. We find that both systems are perceived in a similar way: both achieved high scores in usability and a moderate-high level of perceived trust.

Overall, our work concludes that the tracking code-based systems are indeed promising in terms of being easy to use, perceived as transparent and trusted by the voters. However, we also show that an attacker controlling the voting client can use deceptive voter interface modifications to prevent voters from detecting vote manipulations, which is contrary to the goal of achieving verifiability via tracking codes. We therefore conclude that while using tracking codes for verification can indeed be a promising direction, more research is required into protection against the manipulation tactics we investigated.

2 Related Work

The usability evaluation of end-to-end verifiable voting systems has been a subject of several studies. In particular, the ability of voters to detect various types of manipulations during vote casting is investigated, e.g. in [Ac14; Ka11; Ku20; Ma18; Vo22a]. There are those manipulations in which only the chosen candidate is altered before sending the encrypted vote to the server (simple manipulation types) and those in which the user interface of the voting client is altered in a way that any information related to verification is removed (deceptive manipulation). In particular, the deceptive manipulation show high efficacy of such attacks; i.e. only a minority of voters being able to detect such manipulations, see, e.g. [Ku20; Vo22a]. None of these studies, simulating the deceptive manipulation, however, studied verification using tracking codes, but studied other types of verification techniques.

Systems relying on tracking code verification have, however, been a subject of empirical studies not involving deceptive manipulation of user interfaces – while studying manipulation detection efficacy and/or how the system was perceived, e.g. from usability point of view: Marky et al. [Ma21] have studied the ability of voters to detect simulated simple manipulations, with 84% of participants being able to detect that their vote was altered (while there was no change in the UI). The study furthermore demonstrated high user satisfaction as measured by System Usability Scale. In addition, in their study, only 44% of voters expressed trust in the system by answering that they are confident that they can verify the integrity of their votes. Other user studies focused on Selene [RRI15], a system that employs a variant of tracking code-based verification that distributes the tracking codes after the tally [AS20; Di19; Zo21]. Similar to the study by Marky et al [Ma21], the studies have demonstrated high usability of the evaluated system, yet low levels of perceived trustworthiness and lack of understanding of the verification process.

Overall, the related work on tracking codes is limited compared to the amount of research on the other type of verification approaches and the various ways to implement tracking codes (see Abschnitt 3). In particular, verification efficacy of such systems given an attacker capable of modifying user interfaces of the voting client has not been studied yet. However, those that exist show that such an approach can be promising in terms of ease of use. Therefore, our research contributes in studying two types of tracking codes regarding perceived usability and manipulation detection efficacy for three different attacks. In order to address the issues with perceived trustworthiness and understandability issues, identified by related work, we developed our proposals in an iterative way.

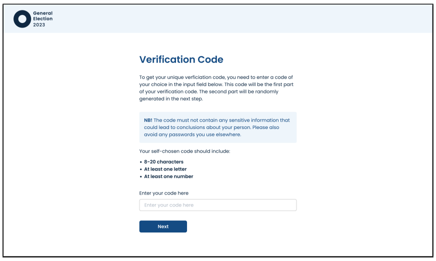
3 Proposed Voting Systems

In this section we describe the prototypes for two tracking code-based systems that we used in our evaluations[4]. As such, we chose to compare two methods of assigning a tracking code to the voter: *partly self-chosen*, where the voter chooses half of the code themselves and the other half is generated by the system similarly to [Kü16b] (*System-Self*, see Figure 1) and *auto-generated* where the entire code is generated by the system (*System-Auto*, see Figure 2) [5]. We furthermore decided to assign tracking codes to the voters before voting in order to mitigate clash attacks [KTV12], because adversaries cannot be sure which option a voter is going to choose, unlike if tracking codes were assigned after vote casting. After being assigned the tracking code and voting, the voter in both systems gets an option to download their tracking code as a PDF file, also containing the tracking code encoded in a

[4] For a more detailed description of the design process of both systems, see the extended version of our paper here [Ni]

[5] We decided against implementing fully self-generated codes due to usability issues and concerns regarding voters' ability to generate unique codes [Bo12]

QR code to facilitate later verification with a mobile device for ensuring better usability and device independence. After the voting phase is finished, the voters are given an option of either visiting the bulletin board website listing all the votes with tracking codes, or scanning the QR code that forwards them to a website showing only the vote assigned to their code [6]. The screenshots showing all the steps in both *System-Self* and *System-Auto* are provided in the extended version of our paper [Ni].



Verification Code

To get your unique verficiation code, you need to enter a code of your choice in the input field below. This code will be the first part of your verification code. The second part will be randomly generated in the next step.

NB! The code must not contain any sensitive information that could lead to conclusions about your person. Please also avoid any passwords you use elsewhere.

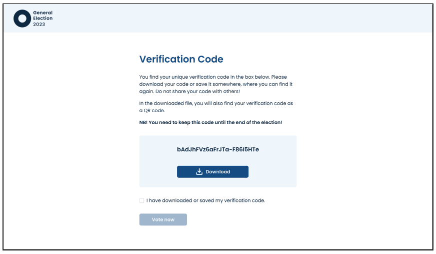
Your self-chosen code should include:

- **8-20 characters**
- **At least one letter**
- **At least one number**

Enter your code here

Enter your code here

Abb. 1: System-Self



Verification Code

You find your unique verification code in the box below. Please download your code or save it somewhere, where you can find it again. Do not share your code with others!

In the downloaded file, you will also find your verification code as a QR code.

NB! You need to keep this code until the end of the election!

bAdJhFVz6aFrJTa-F86I5HTe

□ I have downloaded or saved my verification code.

Abb. 2: System-Auto

After designing initial prototypes of *System-Self* and *System-Auto*, we conducted two rounds of lab usability testing ($N = 4$ and $N = 6$ respectively) and one round of remote moderated testing using video and chat ($N = 4$). Following the feedback from the testing, we made several enhancements to functionality and usability, such as relaxing the requirements for partly self-chosen tracking codes.

4 Methodology for the evaluation of the actual system

We describe the manipulation tactics we simulated to study **RQ1**, the measurements we used for answering both **RQ1** and **RQ2** and the study conducted to answer these research questions.

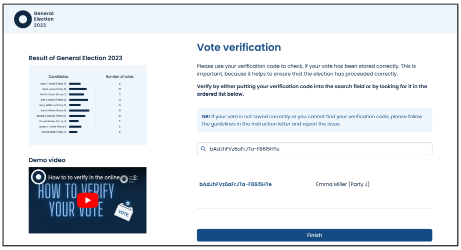
4.1 Manipulation tactics

We consider an attacker whose goal is to either change the vote cast by the voter into a vote for another candidate of attacker's choosing or to nullify the vote, making sure that it is not included in the tally. We assume that the attacker is capable of at least partial control over the voting client, being able to modify the cast votes as they are being sent to the voting system, as well as make changes to the user interface of the voting client. In order to prevent the voter from detecting a manipulation, such an attacker can therefore be assumed to attempt various tactics to prevent the voter from verifying their vote. Following previous research [Ku20; Vo22b], we study the following tactics.

[6] Note that in the latter case, the device scanning the QR code needs to be trusted for vote secrecy; on the other hand, the voter can potentially let it scan a QR code with a tracking code of another voter in case of concerns

Replace-Vote In the simplest tactic, the attacker does not attempt to do anything beyond manipulating the cast vote, so that the voter still gets their tracking code assigned, and the tracking code is published next to the vote sent by attacker. Once the voter attempts to verify their vote, they would therefore find their tracking code next to "Emma Miller", which is different to their intended vote for Sarah Wilson (see Abb. 3). Detecting the manipulation would then require noticing this mismatch.

Remove-Vote In the next tactic, the attacker removes the voter's vote from the bulletin board. Note, this tactic can be applicable in an attack where the adversary replaces the vote and alters the voting client UI so that the manipulated vote is not displayed, or an attack where the adversary removes or blocks the transmission of the vote entirely. Detecting this attack would require voter to understand that the absence of their code on a bulletin board is a sign of a problem with verification (see Abb. 4).



Result of General Election 2023

Candidates	Number of votes
Alice T. Smith (Party A)	8
Mark Jones (Party B)	13
Martin Taylor (Party C)	11
Ann K. Brown (Party D)	14
Mary Williams (Party E)	8
Sarah Wilson (Party F)	16
James R. Davies (Party G)	12
Daniel Walker (Party H)	7
Jacob S. Turner (Party I)	9
Emma Miller (Party J)	6

Vote verification

Please use your verification code to check, if your vote has been stored correctly. This is important, because it helps to ensure that the election has proceeded correctly.

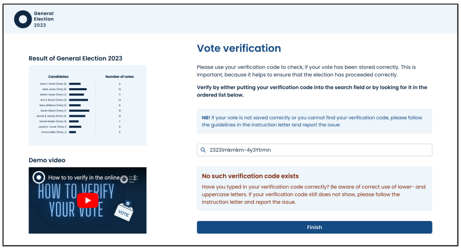
Verify by either putting your verification code into the search field or by looking for it in the ordered list below.

NB! If your vote is not saved correctly or you cannot find your verification code, please follow the guidelines in the instruction letter and report the issue.

bAdJhFVz6aFrJTa-F86I5HTe

bAdJhFVz6aFrJTa-F86I5HTe	Emma Miller (Party J)

Abb. 3: Replace-Vote



Result of General Election 2023

Candidates	Number of votes
Alice T. Smith (Party A)	8
Mark Jones (Party B)	13
Martin Taylor (Party C)	11
Ann K. Brown (Party D)	14
Mary Williams (Party E)	8
Sarah Wilson (Party F)	15
James R. Davies (Party G)	12
Daniel Walker (Party H)	7
Jacob S. Turner (Party I)	9
Emma Miller (Party J)	5

Vote verification

Please use your verification code to check, if your vote has been stored correctly. This is important, because it helps to ensure that the election has proceeded correctly.

Verify by either putting your verification code into the search field or by looking for it in the ordered list below.

NB! If your vote is not saved correctly or you cannot find your verification code, please follow the guidelines in the instruction letter and report the issue.

2323Imkmkm-4y3Ytrmn

No such verification code exists

Have you typed in your verification code correctly? Be aware of correct use of lower- and uppercase letters. If your verification code still does not show, please follow the instruction letter and report the issue.

Abb. 4: Remove-Vote

Remove-Process The third tactic involves a comprehensive manipulation of the UI with attackers removing everything related to the verification process. Thus, the voter does not receive any tracking code at all and is directed to vote casting straight after being welcomed by the voting system. The texts and graphics on the voting website are furthermore altered to remove any mentioning of the verification. Detecting such a manipulation requires the voter to be aware of the verification possibility – e.g. from thorough reading of the information materials related to the election – and therefore noticing that necessary steps are missing in the voting system.

The manipulation tactics require different levels of involvement on behalf of the attacker – as such, while the *Replace-Vote* tactic does not require any additional action from the attacker aside from manipulating the cast vote, the *Remove-Process* tactic requires a much more thorough modification of the user interface. Hence, depending on the level of access the attacker has to the voting client, some of the tactics might be much more challenging for them to perform. On the other hand, applying the tactics *Replace-Vote* and *Remove-Vote* has an advantage for the attacker in that the manipulation would only be detected after the election (i.e. when the voter either sees the wrong vote next to their code or cannot find their code at all in the list of all published votes), at which point it can become much more damaging to the election integrity. The *Remove-Process* manipulation, on the other hand, can be detected already during vote casting, which can lead to a more timely mitigation on behalf of election authorities.

4.2 Measurements

In order to answer our research questions, we collect the following measurements.

Manipulation detection (RQ1) In order to answer RQ1, we evaluate the manipulation detection rates as the share of participants who reported detecting manipulation when interacting with both systems with all three manipulations tactics.

Usability and trust (RQ2) We measure both usability and trust in the evaluated systems, given the scenario where the voters were not subjected to any manipulations. For measuring usability of the system, we rely on the System Usability Scale. For measuring trust, we use the Trust in Voting Systems questionnaire [AKO22]. In order to get further insights on trust perceptions of both systems, we furthermore use the TDIV scale [Ag23] to measure perceived transparency of the systems.

4.3 Study Procedure

We conducted a between-subject online experiment, resulting in overall seven groups of participants. Of these groups, five were subjected to one out of three manipulation tactics with one out of two systems, allowing us to investigate RQ1. Note, since the *Remove-Process* manipulation removes any references to vote verification from the system, the resulting user interface is indistinguishable between *System-Auto* and *System-Self*; hence, we decided to form one group of participants interacting with this manipulation tactic. For each of the remaining two manipulation tactics (that is, *Remove-Vote* and *Replace-Vote*), two groups of participants were formed, interacting with *System-Auto* and *System-Self* for the corresponding manipulation tactic. Two more participant groups interacted with either *System-Auto* or *System-Self* system without vote manipulation, allowing us to investigate RQ2.

The data collection and recruitment in two phases. In the first recruitment phase (conducted in April 2023), the participants were randomly assigned into one out of four groups, that is, being assigned either *System-Auto* with *Replace-Vote* manipulation, *System-Auto* with *Remove-Vote* manipulation, *System-Self* with *Replace-Vote* manipulation and *System-Self* with *Remove-Vote manipulation*. In the second recruitment phase (conducted in December 2023), participants were randomly assigned to one out of three groups, that is, being assigned either the *Remove-Process* manipulation, *System-Auto* without manipulation or *System-Self* without manipulation. For an overview of the study procedure, see Abb. 5.

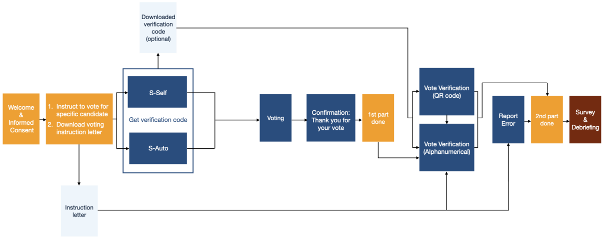

Downloaded verification code (optional)
Welcome & Informed Consent
1. Instruct to vote for specific candidate
2. Download voting instruction letter
S-Self
Get verification code
S-Auto
Voting
Confirmation: Thank you for your vote
1st part done
Vote Verification (QR code)
Vote Verification (Alphanumerical)
Report Error
2nd part done
Survey & Debriefing
Instruction letter

Abb. 5: Study procedure

The study consisted of two parts. In the first part, the participants were instructed to interact with their assigned voting system and cast a vote for a fictional candidate [7]. After completing the vote casting, the participants in groups with verification process intact were forwarded to the mock results page, where they were able to view the final tally of the election as well as verify that their own vote has been recorded correctly. For the participants in a group where the verification process was removed (that is, simulating the *Remove-Process* manipulation tactic), this step was omitted. For the second part, after the voting (for participants in groups subjected to *Remove-Process* manipulation) or after the verification (for participants in all of the other groups), the participants were forwarded to the study questionnaire. We explain the individual study steps in more details below.

Welcome and informed consent: Upon landing on the initial page redirected from the recruiting system, participants were prompted to provide informed consent. They were briefed on the study's purpose, focusing on exploring their user experience and perceived trustworthiness of an online voting system designed for general elections. Following this, participants were randomly directed to one of the four prototypes in the first run and one of the three prototypes in the second run.

Voting: Before entering the actual voting process of the prototype, participants were guided to an information page, urging them to vote for the candidate "Sarah Wilson" and confirm their commitment. Simultaneously, participants were instructed to download an election instruction letter, mimicking how they would receive it by letter or email in a real-world election [8]. The letter, among other instructions, furthermore included a link to a

[7] In order to make it clear for the participants to distinguish between study instructions and parts of the mock voting system, we employed two distinct color schemes for these two types of pages.

[8] The full text of the letters is provided in the extended version of our paper [Ni]

website the participants could use to report any problems they experience with the system. After downloading the letter, they were forwarded to their corresponding voting system where they cast their vote.

After participants completed the voting, the participants who interacted with prototypes that left the verification process intact (that is, participants in groups not subjected to manipulation as well as participants in groups subjected to manipulation tactics *Replace-Vote* and *Remove-Vote*) were directed to another information page. This page explained that, in a real election, the verification phase would only begin after the voting and tallying phases are complete. However, as a part of the study, they could verify their vote immediately after voting. Once the participants verified their vote, they were guided to the study questionnaire. to As the participants in groups subjected to *Remove-Process* were not subjected to any verification process within the voting system, they were forwarded directly to the questionnaire after they completed the voting.

Questionnaire: In the final questionnaire, the participants were asked questions about their experience with the system, including SUS questionnaire for evaluating usability, the “overall trust” part of the Trust in Voting Systems [AKO22] questionnaire for evaluating perceived trust and the Transparency Dimensions in Internet Voting questionnaire [Ag23] for evaluating perceived transparency. The participants in groups subjected to manipulations were asked whether they have experienced any problems with the voting system and asked to elaborate on the problems they experienced via an open-ended question. They were afterwards debriefed about the manipulation and asked whether they have detected the manipulation and whether they reported it; if they answered that they detected the manipulation but did not report it, they were furthermore asked about their reasons for not reporting via a multiple-choice question of (pre-selected reasons based on previous research) and an open-ended “Other” option. The questionnaire furthermore included an attention check, as an item “This question is not part of the survey and just helps us to detect bots and automated scripts. To confirm that you are a human, please choose ‘Disagree’ here”.

4.4 Recruitment and ethics

We used Prolific platform for recruitment, using the gender-balanced option. While Prolific samples are known to be biased to a younger, educated and digitally savvy population, previous research shows that the platform can provide sufficient validity for studies related to security and privacy [TBL22]. We paid participants 4.5 GBP (April 2023) for the data collection as conducted in April for an estimated 30 minutes of participation, which is above the hourly rate recommended by Prolific. As the actual median duration of the study ended up being lower than expected, the reimbursement for the study in December has been lowered to 3 GBP for an expected duration of 20 minutes, keeping the hourly rate the same. Moreover, we had a filter that only recruited participants who were fluent in English.

While there is no mandatory ethical review process at our institutions, measures were taken to ensure participants' informed consent and confidentiality. Before accessing the voting system, participants signed a consent form providing details on the study's purpose, withdrawal options, and data handling. Researchers' contact details were also disclosed for inquiries. Given the sensitivity of political beliefs, all parties and candidates were fictional, a fact communicated at the study's outset. As the study involved deception with regards to vote manipulation, the participants subjected to manipulation were debriefed in the final study questionnaire about the real purpose of the study and the reason for deception. No personal identifying information about participants has been collected within the study.

5 Results of the user study

After excluding participants who either voted for a candidate other than Sarah Wilson (as they were instructed) or failed an attention check within the survey, a total of 306 participants were included in the data analysis. 149 of the participants (49%) identified as female, 155 as male, one as non-binary and one preferred not to answer. Most of the participants (195, 63%) were between 18 and 30 years old, and most (209, 69%) had at least a Bachelor's degree [9]. Table Tab. 1 shows the distribution of participants depending on the system and manipulation type they interacted with, including groups that interacted with a system that did not include a vote manipulation.

System	Manipulation	N
System-Auto	*Remove-Vote*	25
System-Auto	*Replace-Vote*	25
NA	*Remove-Process*	25
System-Self	*Remove-Vote*	25
System-Self	*Replace-Vote*	25
System-Self	None	92
System-Auto	None	89

Tab. 1: Distribution of participants by groups depending on system/manipulation. Note that since both *System-Auto* and *System-Self* look the same under the *Remove-Process* manipulation, we do not mention a specific system for this manipulation tactic.

5.1 RQ1 - Manipulation Detection

A total of 56 participants (44%) who were subjected to a manipulation reported it using the form on the website referenced in the study instructions. In the survey, 61 participants (49%) reported having problems with verifying their vote when asked a question about experiencing any issues with the system before debriefing. When asked whether they detected the manipulation after debriefing, 71 participants (57%) responded that they

[9] The full breakdown of the demographics is included in the extended version of our paper [Ni]

detected and reported the manipulation, while 31 (25%) more answered that they detected the manipulation but did not report it. Tab. 2 shows a breakdown of manipulation detection rates using different metrics by the system and manipulation tactic.

System	Manipulation	Voting System	Survey before Debriefing	Survey after Debriefing
System-Auto	*Remove-Vote*	13 (52%)	16 (64%)	21 (84%)
System-Auto	*Replace-Vote*	15 (60%)	16 (64%)	25 (100%)
System-Self/System-Auto	*Remove-Process*	0 (0%)	0 (0%)	16 (64%)
System-Self	*Remove-Vote*	9 (36%)	16 (64%)	17 (68%)
System-Self	*Replace-Vote*	19 (76%)	13 (52%)	23 (92%)
	Total	56 (45%)	61 (49%)	102 (82%)

Tab. 2: Overview of manipulation detection rates reported either via the voting system, survey before debriefing, or survey after debriefing (including participants answering that they detected the manipulation but did not report it within the system), separated by system/manipulation.

In order to understand the effect of either system or manipulation tactic on the detection rate, we decided to use the rate of participants reporting the manipulation during voting as our main measurement, in order to avoid the inaccuracies resulting from self-reporting within the survey. Figure 6 shows 95% confidence intervals [10] for all combinations of system/manipulation tactic. As such, our results show that while detection rates were similar between the two systems for each manipulation, the differences between manipulations are more pronounced, with *Remove-Process* manipulation being particularly hard to detect.

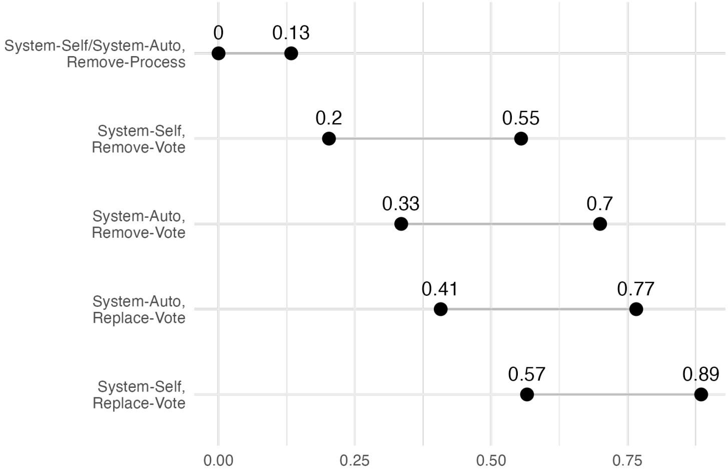


Abb. 6: 95% confidence intervals for manipulation detection rate, for all combinations of system and manipulation tactic.

[10] For all calculations of the confidence intervals, we used R package “DescTools”

5.2 RQ2 - Usability and Trust

Tab. 3 provides an overview of the mean scores for usability (using the SUS scale, ranging from 0 to 100) and trust (using the TVS "overall trust" scale, ranging from 1 to 7) for both *System-Self* and *System-Auto* (among the participants that were not subjected to any manipulation). Both of the systems fall within the range of "Good" to "Excellent" grades for usability [BKM09] and elicit a moderate to high level of trust.

	System-Auto	System-Self
Usability	M = 81.01	M = 83.07
	SD = 13.54	SD = 14.22
	CI = 78.23; 84.02	CI = 80.11; 85.98
Trust	M = 5.02	M = 5.35
	SD = 1.41	SD = 1.46
	CI = 4.72; 5.31	CI = 5.06; 5.67

Tab. 3: Overview of usability and trust scores for each system (M: mean, SD: standard deviation, CI: 95% confidence interval)

We furthermore calculated the scores from the scales for the perceived information availability, understandability, verifiability and general transparency of the systems, see Tab. 4, resulting in a moderate to high level of fulfillment across all the dimensions for both systems. While both of the systems had similar scores for perceived information availability, understandability and general transparency of the systems, the difference in scores for participants' perceived capability of the system to enable verification of cast votes was more pronounced, hinting at the need for further investigation in this direction.

	System-Auto	*System-Self*
Information availability	M = 4.92	M = 4.97
	SD = 1.1	SD = 1.44
	CI = 4.7; 5.15	CI = 4.7; 5.25
Understandability	M = 5.7	M = 5.76
	SD = 0.78	SD = 0.98
	CI = 5.54; 5.87	CI = 5.59; 5.96
Verifiability	M = 5.72	M = 5.98
	SD = 0.77	SD = 0.73
	CI = 5.56; 5.88	CI = 5.83; 6.14
General transparency	M = 5.4	M = 5.69
	SD = 1.27	SD = 1.28
	CI = 5.16; 5.68	CI = 5.45; 5.95

Tab. 4: Overview of scores TVID Dimensions for *System-Self* and *System-Auto* (M: mean, SD: standard deviation, CI: 95% confidence interval). The scores are calculated as mean of Likert scale items, ranging from 1 (low) to 7 (high). The 95% confidence intervals are calculated using the bootstrapping method, due to non-normal distribution of data.

6 Discussion

Study limitations Our research was carried out among participants, most of whom lacked familiarity with internet voting systems. Consequently, it raises the question of how applicable our results are to different demographics. This includes voters in nations like Estonia or Switzerland where internet voting is established, and the representation of various demographics beyond the younger, educated Prolific sample. Furthermore, as the data collection has been conducted in two phases, it is not clear whether this separation has contributed to a difference between *Remove-Process* and the other two manipulation tactics. Furthermore, in our study, the participants voted for a fictional candidate, minimizing personal data collection and promoting broader participation. While this approach reduced potential dropout due to privacy concerns, it's essential to note that using fictional candidates may impact participant seriousness in reporting.

Manipulation detection (RQ1) Our study shows relatively low verification detection rates for our systems. As such, none of the voters reported detecting manipulation using the link in the election instructions for the *Remove-Process* manipulation. Even if self-reporting of manipulation detection is taken into account (that is, assuming that the participants are telling the truth about detecting the manipulation but not reporting it), the rate of undetected manipulations remains up to 64% depending on the manipulation tactic. Furthermore, the verification process using tracking codes is optional – that is, the voter has to actively choose to verify their vote after the election is over. This can lead to low verification rates, as demonstrated by Internet voting in Estonia, where only around 5% of voters choose to verify their votes [Es24] [11]. Combined with low manipulation detection rates, the risk of undetected manipulation can be high if only a small percentage of voters chooses to verify their votes, and among the ones that do, a large share does not detect their votes being manipulated.

Our findings show that the *Remove-Process* manipulation was by far the hardest to detect. Since this manipulation implies that no hints are given to the voter via the voting client user interface, it is critical to communicate the need to verify one's vote via alternative channels (e.g. media campaigns) and provide instructions outside of the voting system specifying how to verify one's vote. Since similar studies of other verification techniques show improvement in verification rates given properly designed instructions available to voters as paper materials [Vo22a], designing such instructions for tracking code-based verification is an important direction of future work.

Overall, all three types of simulated attacks can become threats in a real election. Therefore, additional education is needed on a societal level outside the voting system, as especially instruction letters are not sufficient to detect different kind of manipulations. Additionally, ensuring the existence of reliable reporting channels is essential for the voters to report

[11] Note that the Estonian voting system relies on a different verification approach.

manipulations. Investigating most appropriate ways to establish and communicate such channels, as well as processes for handling reported manipulations (in particular, also accounting for voters who might lie about verification failures, aiming to create distrust in the election result) is an important direction of future work.

Our study furthermore has shown a large discrepancy between voters detecting and reporting the manipulation during or directly after interacting with the voting system, and self-reporting detecting the manipulation after being debriefed about its presence. Such discrepancy can be explained by several reasons, such as social desirability bias (i.e. participants not wanting to admit that they missed the manipulation), participants not feeling necessary to report a manipulation in a study setting, or not being able to find a link to the reporting form. Nonetheless, the real manipulation detection rates can be only roughly estimated, and a consistent metric for such an estimation has to be applied for further studies on manipulation detection rates, to ensure that the results of the studies are comparable with each other.

Usability and trust (RQ2) Our findings show moderate to high scores of usability and trust in both of our two proposed prototypes (with no manipulations). However, our proposed prototypes do not provide any information about how such types of systems are secured, nor do they inform voters of remaining risks such as voter coercion or violations of eligibility due to insufficiently secure voter authentication. Consequently, it remains an open question how trust will be affected if we notify people about the potential security risks related to electronic voting, including manipulations. Furthermore, our findings show a difference in perceived verifiability, with the prototype having a partly self-chosen tracking code perceived as more verifiable, as indicated by voters being more likely to agree with statements such as “I can confirm that the voting system accurately recorded my vote”. One possible explanation could be voters feeling a greater sense of control over their tracking code, thus becoming more engaged in the verification process. Nonetheless, this disparity did not result in a notable variance in perceived overall transparency or trust. As studies indicate that there are additional factors affecting both transparency and trust that were not addressed in this research [Ag22], further investigation is needed.

Further limitations of tracking code-based verification Our proposed voting systems, as well as systems relying on tracking code-based verification, do not address several critical risks that can be an issue in Internet voting. As such, while both *System-Self* and *System-Auto* offer an option to use a second device for verification using QR codes, such an option is not enforced, and the voters are furthermore not informed about the risks of using the same device for voting and verifying. Furthermore, as verification can only be done after tallying, vote manipulations would not be detected during vote casting, which might make it more challenging to support the voters (e.g. by telling them to cast their vote via an alternative voting channel in case of failed verification, as is the common practice in countries implementing Internet voting) or otherwise address them after the fact. This issue might become critical even in absence of actual vote manipulations, if there are voters who falsely claim verification failures. Finally, our study do not address the issue of coercion attacks which tracking code-based systems in general are prone to. While such attacks are addressed within the Selene voting system, the resulting system relies on complex cryptographic techniques, thus making the verification process potentially more difficult to understand (see Abschnitt 2). Therefore, future studies need to investigate how to address the risk of coercion attacks when developing tracking code-based systems.

7 Conclusion

Using tracking codes in Internet voting systems can provide an easy and intuitive way for voters to verify the integrity of their cast votes. Our study shows that systems relying on such verification can achieve a high level of usability and trust on behalf of the voters. However, we demonstrate that verification efficacy of tracking code-based systems can be lacking if the attacker can be assumed to have full or partial control over the voting client (e.g. the voting website) – an adversarial capability that cast-as-intended verification approaches, including tracking codes, were specifically designed to protect against. Depending on the specific manipulation tactic such an adversary can apply – such as removing all references of the verification process from the voting client user interface – the rate of voters who detect such a manipulation can be critically low, as shown by our study where none of the voters have noticed such a manipulation. Other kinds of manipulation tactics, involving attacker who is able to hide the vote from the voter's view (either by modifying the user interface or by blocking the vote from reaching the voting system, e.g. by interfering in the network communications), achieve a higher verification efficacy. However, even with such tactics, up to 64% of participants in our study failed to detect and report vote manipulation. Our findings imply that tracking code-based verification can potentially be used for low-stake elections where vote manipulation is not a critical issue. However, in more high-risk environments, additional care must be done to ensure that the verification efficacy of the implemented systems remains sufficiently high, e.g. with the use of properly designed and evaluated information materials educating voters on the verification process, as well as properly functioning and easy to use communication channels for reporting detected manipulations. Future research into development of such information materials and reporting channels is therefore required.

Acknowledgements

This work was supported by a research grant (40948) from VILLUM FONDEN.

Literaturverzeichnis

[Ac14] Acemyan, C. Z.; Kortum, P. T.; 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. In: EVT/WOTE. 2014.

[Ag22] Agbesi, S.; Dalela, A.; Budurushi, J.; Kulyk, O.: What Will Make Me Trust or Not Trust Will Depend Upon How Secure the Technology Is": Factors Influencing Trust Perceptions of the Use of Election Technologies. In: Proceedings of Seventh International Joint Conference on Electronic Voting (E-Vote-ID 2022). Seventh International Joint Conference on Electronic Voting : E-Vote-ID 2022 ; Conference date: 04-10-2022 Through 07-10-2022, University of Tartu, 2022.

[Ag23] Agbesi, S.; Budurushi, J.; Dalela, A.; Kulyk, O.: Investigating Transparency Dimensions for Internet Voting. In: International Joint Conference on Electronic Voting. Springer Nature Switzerland Cham, S. 1–17, 2023.

[AKO22] Acemyan, C.; Kortum, P.; Oswald, F.: The Trust in Voting Systems (TVS) Measure. International Journal of Technology and Human Interaction 18, S. 1–23, 2022, DOI: 10.4018/IJTHI.293196.

[AS20] Alsadi, M.; Schneider, S.: Verify My Vote: Voter Experience. In. 2020.

[Be21] Beckert, B.; Budurushi, J.; Grunwald, A.; Krimmer, R.; Kulyk, O.; Küsters, R.; Mayer, A.; Müller-Quade, J.; Neumann, S.; Volkamer, M.: Aktuelle Entwicklungen im Kontext von Online-Wahlen und digitalen Abstimmungen, Techn. Ber., 46.23.01; LK 01, 2021, DOI: 10.5445/IR/1000137300.

[BKM09] Bangor, A.; Kortum, P.; Miller, J.: Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. J. Usability Stud. 4, S. 114–123, 2009.

[Bo12] Bonneau, J.: The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In: Proceedings - 2012 IEEE Symposium on Security and Privacy, S and P 2012. Proceedings - IEEE Symposium on Security and Privacy, S. 538–552, 2012, DOI: 10.1109/SP.2012.49.

[Di19] Distler, V.; Zollinger, M.-L.; Lallemand, C.; Rønne, P.; Ryan, P.; Koenig, V.: Security - Visible, Yet Unseen? How Displaying Security Mechanisms Impacts User Experience and Perceived Security. 2019, DOI: 10.1145/3290605.3300835.

[Es24] Estonian National Electoral Committee: Statistics about Internet Voting in Estonia, Accessed: 16th of February, 2024, 2024, URL: https://www.valimised.ee/en/archive/statistics-about-internet-voting-estonia.

[Ka11] Karayumak, F.; Olembo, M. M.; Kauer, M.; Volkamer, M.: Usability Analysis of Helios — An Open Source Verifiable Remote Electronic Voting System. In: 2011 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 11). USENIX Association, San Francisco, CA, 2011, URL: https://www.usenix.org/conference/evtwote-11/usability-analysis-helios-%7B%5Ctextemdash%7D-open-source-verifiable-remote-electronic-voting.

[KTV12] Kusters, R.; Truderung, T.; Vogt, A.: Clash Attacks on the Verifiability of EVoting Systems. Proceedings - IEEE Symposium on Security and Privacy, S. 395–409, 2012, DOI: 10.1109/SP.2012.32.

[Kü16a] Küsters, R.; Müller, J.; Scapin, E.; Truderung, T.: sElect: A Lightweight Verifiable Remote Voting System. In: 2016 IEEE 29th Computer Security Foundations Symposium (CSF). S. 341–354, 2016, DOI: 10.1109/CSF.2016.31.

[Kü16b] Küsters, R.; Müller, J.; Scapin, E.; Truderung, T.: sElect: A lightweight verifiable remote voting system. In: 2016 IEEE 29th Computer Security Foundations Symposium (CSF). IEEE, S. 341–354, 2016.

[Ku20] Kulyk, O.; Volkamer, M.; Müller, M.; Renaud, K.: Towards Improving the Efficacy of Code-Based Verification in Internet Voting. In. S. 291–309, 2020, ISBN: 978-3-030-54454-6, DOI: 10.1007/978-3-030-54455-3_21.

[KV18] Kulyk, O.; Volkamer, M.: Usability is not Enough: Lessons Learned from 'Human Factors in Security' Research for Verifiability, Cryptology ePrint Archive, Paper 2018/683, https://eprint.iacr.org/2018/683, 2018, URL: https://eprint.iacr.org/2018/683.

[Ma18] Marky, K.; Kulyk, O.; Renaud, K.; Volkamer, M.: What did I really vote for? On the usability of verifiable e-voting schemes. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. CHI Conference on Human Factors in Computing Systems : Engage with CHI, CHI 2018 ; Conference date: 21-04-2018 Through 26-04-2018, Association for Computing Machinery (ACM), United States, 2018, DOI: 10.1145/3173574.3173750, URL: %5Curl%20https://chi2018.acm.org/%22.

[Ma21] Marky, K.; Grube, T.; Kunze, K.; Zollinger, M.-L.; Roenne, P.; Ryan, P. Y. A.: Investigating Usability and User Experience of Individually Verifiable Internet Voting Schemes. ACM Trans. Comput.-Hum. Interact. 28, 2021.

[Ma22] Marky, K.; Gerber, P.; Günther, S.; Khamis, M.; Fries, M.; Mühlhäuser, M.: Investigating {State-of-the-Art} Practices for Fostering Subjective Trust in Online Voting through Interviews. In: 31st USENIX Security Symposium (USENIX Security 22). S. 4059–4076, 2022.

[Ni] Nissen, C.; Kulyk, O. K.; Volkamer, M.; Fredrich, L. E.; Hermansen, H.: Design and Evaluation of Verifiable Voting Systems Based on Tracking Code Verification. Extended version of the paper, URL: https://anonymous.4open.science/r/e-vote-id2024-0B62/.

[RRI15] Ryan, P. Y. A.; Roenne, P. B.; Iovino, V.: Selene: Voting with Transparent Verifiability and Coercion-Mitigation, Cryptology ePrint Archive, Paper 2015/1105, https://eprint.iacr.org/2015/1105, 2015.

[RRR21] Ryan, P. Y. A.; Rastikian, S.; Rønne, P. B.: Hyperion: An Enhanced Version of the Selene End-to-End Verifiable Voting Scheme. In: Proceedings of the Sixth International Joint Conference on Electronic Voting E-Vote-ID 2021. S. 285–287, 2021.

[TBL22] Tang, J.; Birrell, E.; Lerner, A.: Replication: How Well Do My Results Generalize Now? The External Validity of Online Privacy and Security Surveys. In: Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022). USENIX Association, Boston, MA, S. 367–385, 2022, ISBN: 978-1-939133-30-4, URL: https://www.usenix.org/conference/soups2022/presentation/tang.

[Vo22a] Volkamer, M.; Kulyk, O.; Ludwig, J.; Fuhrberg, N.: Increasing security without decreasing usability: A comparison of various verifiable voting systems. In: Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022). S. 233–252, 2022.

[Vo22b] Volkamer, M.; Kulyk, O.; Ludwig, J.; Fuhrberg, N.: Increasing security without decreasing usability: A comparison of various verifiable voting systems. In: Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022). USENIX Association, Boston, MA, S. 233–252, 2022, ISBN: 978-1-939133-30-4, URL: https://www.usenix.org/conference/soups2022/presentation/volkamer.

[Zo21] Zollinger, M.; Distler, V.; Rønne, P. B.; Ryan, P. Y. A.; Lallemand, C.; Koenig, V.: User Experience Design for E-Voting: How mental models align with security mechanisms. CoRR abs/2105.14901, 2021, URL: https://arxiv.org/abs/2105.14901.

Track 2: Governance Issues

Trust, distrust and failure: Revisiting the use of electronic voting in the Netherlands (2006/07)

David Duenas-Cid[1]

Abstract: This paper focuses on the complex dynamics of trust and distrust in digital government technologies by approaching the cancellation of machine voting in the Netherlands (2006-07). The analysis stresses how, although being a central component, technology's trustworthiness dialogues with the socio-technical context in which it is inserted. Overall, this paper contributes to understanding trust dynamics in digital government technologies, with implications for policymaking and technology adoption strategies.

Keywords: Trust, Distrust, Machine voting, Netherlands

1 Introduction

This research approaches the dynamics of creation of trust and distrust in electoral technologies, focusing on the Netherlands' experience with electronic voting. Trust is a critical factor influencing the adoption of new technologies, and its positive impacts and sources have been widely studied. But what happens when working technologies face a crisis of trust? This research examines a well-known case in the electronic voting literature—the machine voting abandonment in the Netherlands (2006-07) [JaPi09, Loeb16, Oost10]. By revisiting this case, the research analyzes the dynamics of trust erosion and the complex relation between trust and distrust, providing new knowledge on how distrust is created in electronic voting. This paper is a short version of the piece „Trust and distrust in electoral technologies: what can we learn from the failure of electronic voting in the Netherlands (2006/07)“ [Duen24].

2 Theoretical framework

Trust is a complex concept, allowing for various approaches from many disciplines. In e-governance, trust is important in technology adoption theories and models such as TAM or UTAUT. Differently, the concept of distrust has not attracted the same level of attention among scholars [SiLa23], and the relation between trust and distrust has often been seen as symmetrically opposite concepts. Following the insights of Luhmann [Luhm79] and Lewicki [LeMB98], distrust can be understood as a concept possessing its functioning

[1] Kozminski University, Pub/Tech Research Center, Jagiellońska 57, 03-301 Warsaw (Poland), dduenas@kozminski.edu.pl, https://orcid.org/0000-0002-0451-4514

doi:10.18420/e-vote-id2024_07

patterns, closely intertwined with but separate from trust: not trusting does not necessarily mean distrusting. Recognizing the nuanced distinction between trust and distrust allows understanding that individuals may simultaneously place trust in certain aspects of a system while distrusting others (e.g., distrusting politicians while trusting democracy [Szto03]), that trust and distrust can change over time, even towards the same entity, or that certain elements within a system may foster both trust and distrust at the same time.

This approach is still quite unexplored concerning election technologies, where trust is often approached as an isolated concept. This paper, aligned with previous research by the same author [DuCa23, Duen22, DuJK22], understands paper advocates for treating trust and distrust as distinct yet interconnected concepts.

As mentioned, this paper focuses on the use of voting machines in the Netherlands. Their adoption aimed to streamline the process of manual vote counting, enhance accuracy, and mitigate human errors in vote casting, which often resulted in invalidated ballots [Blan16]. The credibility of voting machines was challenged by the hacktivist group ”Wij vertrouwen stemcomputers niet” (Trad. We don't trust voting computers). The group demonstrated that the voting machines were susceptible to hacking and compromising the secrecy of the vote [GoHe07]. Their successful campaign exposed a set of vulnerabilities provoking the decision to discontinue voting machines in 2008, reverting to paper ballots—a decision that remains unchanged.

3 METHOD

This paper proposes a qualitative method, gathering and analyzing the discourses of key actors and identifying the primary elements influencing the creation of trust and distrust and their potential impact on public perception. The research methodology involves qualitative interviews with members of the hacktivist group, the electoral management board, the Ministry of Interior Affairs, researchers in IT and social sciences, electoral observers, and journalists—all directly involved in the crisis. While the number of interviews conducted is objectively modest for a qualitative study, the relevance of the interviewees and their direct involvement in the events analyzed mitigate this potential limitation. The interviews were coded and analyzed using NVivo software.

4 DISCUSSION

The Dutch case is relevant for different reasons. Firstly, the experience of the Netherlands created an important precedent in the community [Loeb16, Oost10], for example, highlighting the need to avoid black-box voting systems and promote transparency to build trust and help manage distrust. However, its impact goes beyond academia; for instance, the German constitutional court's ruling against voting machines and stressing the need for transparency in the electoral process without specialist technical knowledge.

Regarding trust and distrust, the fact that the system was very convenient (especially for electoral managers), the generalized lack of technical expertise, and a prevailing culture of trust in electoral authorities and institutions veiled the potential risks of using voting machines. It simply took a small but well-prepared group of hacktivists to disrupt the system entirely. When revisiting the case, it becomes evident that the biggest hack of the hacktivist group was not towards the machines that were easy to hack [GoHe07] but to the overall electoral management system. The actions of the hackers unveiled several problems in the adoption and management of voting machines. The rise of distrust did not only come from the machines themselves but also from their bad security management, the lack of legal provisions in cybersecurity, or how voting machines were physically safeguarded.

Based on the previous, we can shortlist a set of factors contributing to trust or distrust, distributed using the proposed theoretical differentiation between trust and distrust that allows detecting elements that simultaneously contribute to the creation of trust and distrust, as well as others having a negligible impact on either. In Table 1, the main elements in play are shortlisted, describing their impact on the creation of trust and/or distrust.

Factor (in alphabetical order)	Contributes to Trusting/ Distrusting	Reason
Expert debate	None	The complexity of the debate kept it far from most of the population. The lack of political controversy on the topic helped keep it largely unnoticed by a broad audience.
High level of trust in institutions	Trust	The high level of trust in institutions among the Dutch citizenry facilitated the acceptance and unquestioned adoption of voting machines but also mitigated the negative impact of the case.
Lack of critical approach towards Voting Machine	Both	The lack of criticism regarding voting machines fostered a false sense of security, wherein trust was posited in something not deserving of it, consequently seeding the roots of distrust.
Legal amendments after the crisis	Trust	Reforming legislation and ensuring transparency and openness in elections allowed the integrity of future electoral management and democracy.
Long-term reaction of the administration	Trust	Resolving the problem, withdrawing voting machines, allowed protecting the trustworthiness of electoral management and democracy.
Media and Hacktivist campaign	Distrust	The hacktivists' media communication strategy resulted in an effective control of the discourse, fueling the creation of distrust in the voting machines.
Outdated and inadequate legal framework	Both	The legal framework was unprepared to address the emerging problem as the security concept did not

		align with the actual threats. The legal framework legitimized the system to the public and became a source of distrust when proven inadequate.
Positive previous experience	Trust	The successful experience using voting machines prevented critical discourses from appearing.
Previous activities of the hacktivists	Both	The hacktivists' prior activities gave them legitimacy amongst certain key actors. While bringing distrust to the voting machines, they contributed to increasing the trustworthiness of the overall democracy.
Short-term reaction of the administration	Distrust	The administration's lack of capacity to effectively address the concerns raised by the hacktivists generated distrust in the system and cast a general doubt on the suitability of using voting machines.
Weaknesses of the voting machine	Distrust	The proven weaknesses of the voting machines were a legitimating factor for the hacktivist claims.
Withdrawal of the system	None	The withdrawal of voting machines did not significantly impact the democratic system itself.

Tab. 1: Factors contributing to trust and/or distrust.

5 Conclusions

This case leaves interesting takeaways on how trust and distrust intertwine when managing electoral technologies. First of all, the trustworthiness of technology is at the center of the problem. Still, it is inserted in a socio-technical environment, in a permanent dialogue with the cultural and administrative context in which it operates. Secondly, the case shows how trust and distrust can occur in parallel. The legal framework surrounding the Dutch voting machines exemplifies this duality: it initially helped trust in the reliability of the machines, yet when it failed to address the concerns raised by hacktivists, it became a source of distrust. Democracy heavily relies upon the existence of societal agreements that are secured by the performance of public administrations.

Thirdly, the dynamics of trust and distrust creation have different impacts across societal levels. While the discussions surrounding voting machines deeply affected the ministry's core, its impact on average citizens was limited [Loeb11]. This sets a dangerous precedent; people will use insecure systems if they feel or think they are secure [OoBe04], highlighting the relevance of social factors for the production of trust. Finally, convenience is crucial in creating trust in electoral technologies. A primary argument favoring technology adoption is its ability to simplify activities or processes. However, every technological adoption entails a series of trade-offs that must be carefully evaluated before implementation. These trade-offs typically manifest as humanly created risks that we accept in exchange for the benefits offered by the technology [Beck92]. In this case, the risk was the potential for vendors to manipulate election results, with no means of definitively proving such tampering. Therefore, convenience can be understood as a gateway to technology acceptance on the condition that overall experiences remain positive and risks are perceived as manageable.

Acknowledgements

Data collection for this paper was funded by the Polish National Research Center's grant Miniatura 3 - 2019/03/X/HS6/01688. The data analysis and development of the publication has been funded by the Polish National Research Center's grant OPUS-20 - 2020/39/B/HS5/01661 and EU H2020 MSCA Program, grant agreement no. 101038055.

Bibliography

[Beck92] Beck, Ulrich: RISK SOCIETY, Towards a New Modernity. London: SAGE Publications, 1992 — ISBN 0803983468

[Blan16] Blankesteijn, Herbert: Vertrouw ons nou maar: Opkomst en ondergang van de stemcomputer. Amsterdam : Boom Uitgevers, 2016

[DuCa23] Duenas-Cid, David ; Calzati, Stefano: Dis/Trust and data-driven technologies. In: Internet Policy Review Bd. 12 (2023), Nr. 4

[Duen22] Duenas-Cid, David: A theoretical framework for understanding trust and distrust in internet voting. In: Krimmer, R. ; Volkamer, M. ; Duenas-Cid, D. ; Germann, M. ; Glondu, S. ; Hofer, T. ; Krivonosova, I. ; Martin-Rozumilowicz, B. ; u. a. (Hrsg.): E-Vote-ID 2022 Proceedings. Tartu : University of Tartu Press, 2022, S. 57–62

[Duen24] Duenas-Cid, David: Trust and distrust in electoral technologies: what can we learn from the failure of electronic voting in the Netherlands (2006/07). In: Duenas-Cid, D. ; Liao, H.-C. ; Macadar, M. A. ; Bernardini, F. (Hrsg.): 25th Annual International Conference on Digital Government Research (DGO 2024), June 11–14, 2024, Taipei, Taiwan. Taipei : ACM , 2024, S. 1–9

[DuJK22] Duenas-Cid, David ; Janowski, Tomasz ; Krimmer, Robert: Trust and Distrust in e-Democracy. In: DG.O 2022: The 23rd Annual International Conference on Digital Government Research. New York, NY, USA : ACM, 2022 — ISBN 9781450397490, S. 486–488

[GoHe07] GONGGRIJP, ROP ; HENGEVELD, WILLEM-JAN: Studying the Nedap/Groenendaal ES3B Voting Computer: A Computer Security Perspective. In: Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology, EVT'07. USA : USENIX Association, 2007, S. 1

[JaPi09] JACOBS, BART ; PIETERS, WOLTER: Electronic Voting in the Netherlands: From Early Adoption to Early Abolishment. In: ALDINI, A. ; BARTHE, G. ; GORRIERI, R. (Hrsg.): Foundations of Security Analysis and Design. Berlin : LNCS Springer, 2009, S. 121–144

[LeMB98] LEWICKI, ROY J ; MCALLISTER, DANIEL J ; BIES, ROBERT J: Trust and Distrust : New Relationships and Realities. In: Academy of Management Review Bd. 23 (1998), Nr. 3, S. 438–458

[Loeb11] LOEBER, LEONTINE: Voter trust in the Netherlands between 2006 and 2010. In: PARYCEK, P. ; KRIPP, M. ; EDELMANN, N. (Hrsg.): CeDEM11 Proceedings of the International Conference for E-Democracy and Open Government. Krems : Edition Donau-Universität Krems, 2011, S. 323–335

[Loeb16] LOEBER, LEONTINE: E-Voting in the Netherlands; from General Acceptance to General Doubt in Two Years. In: 3rd International Conference on Electronic Voting 2008. Bregenz : Gesellschaft für Informatik, 2016 — ISBN 9783885792253, S. 21–30

[Luhm79] LUHMANN, NIKLAS: Trust and Power. Chichester : Wiley-Blackwell, 1979

[OoBe04] OOSTVEEN, ANNE-MARIE ; VAN DEN BESSELAAR, PETER: Security as belief User's perceptions on the security of electronic voting systems. In: Electronic Voting in Europe: Technology, Law, Politics and Society Bd. 47 (2004), Nr. May 2014, S. 73–82 — ISBN 3885793768

[Oost10] OOSTVEEN, ANNE-MARIE: Outsourcing Democracy: Losing Control of e-Voting in the Netherlands. In: Policy & Internet Bd. 2 (2010), Nr. 4, S. 196–215

[SiLa23] SIX, FRÉDÉRIQUE E. ; LATUSEK, DOMINIKA: Distrust: A critical review exploring a universal distrust sequence. In: Journal of Trust Research (2023), S. 1–23

[Szto03] SZTOMPKA, PIOTR: Trust: A Sociological Theory. Cambridge : Cambridge University Press, 2003

Challenging the idea that internet voting verification tools create trust: they serve as distrust mitigation tools

David Duenas-Cid[1], Vladimir Misev[2]

Abstract: Individual verification mechanisms in elections allow voters to ascertain that their votes are correctly cast and ensure overall electoral integrity by identifying possible issues such as large-scale attacks. Previous research shows that while some voters find these tools helpful, the usage rates of these mechanisms are generally low, and, therefore, their impact on building trust is less than expected. Despite this, there is consensus on the need to introduce verification mechanisms and the expectation that they help build trust. Based on interviews conducted in Estonia and New South Wales (Australia), we posit that verification mechanisms do not create trust but rather mitigate distrust by providing a security layer for detecting possible problems and reducing the appearance of possible distrust discourses.

Keywords: Verification, internet voting, trust, distrust.

1 Introduction and theoretical background

The use of electronic and internet voting has been surrounded from the very beginning by debates on how to increase the security of the system [VoSD11]. While the mechanics of paper voting are generally easy to understand, the mechanics of electronic and, especially, internet voting are more complex and require advanced levels of technical expertise to observe and assess. In some contexts, this generalized lack of capacity to understand the voting system's insights has been a major reason for introducing legal impediments to the use of the different forms of electronic voting [FiJö22]. Translating security features into something that citizens can understand has been one of the challenges for researchers in electronic voting. With this goal in mind, different solutions are being proposed, assuming that if voters understand better how the system works, they will trust it more.

The proposed solutions for solving this challenge regarding the lack of understandability of internet voting methods have generally followed two main streams. The first one includes solutions that either replicate, as much as possible, paper voting processes that are already familiar to voters (for example, using double envelope system metaphors mimicking postal voting [HMVW16] or allocating observation mechanisms for electoral technologies [Osce13, Osce24]) or follow easy-to-understand processes that resonate with the real world (shared decryption keys between a set of trusted citizens). The second stream includes solutions and mechanisms that are ad hoc created to benefit from the

[1] Kozminski University, Pub/Tech Research Center, Jagiellońska 57, 03-301 Warsaw (Poland), dduenas@kozminski.edu.pl, https://orcid.org/0000-0002-0451-4514
[2] International Election Expert, Warsaw (Poland), vladimir.misev@gmail.com

doi:10.18420/e-vote-id2024_08

properties of electoral technologies (such as multiple voting methods or auditing systems benefiting from the traceability of online systems) and that are not available in traditional paper-based voting methods.

Verification mechanisms are one of the proposed solutions belonging to this second stream. They can be defined as systems providing verifiable (cryptographic) proof allowing any particular voter to ascertain whether the vote has been included in the tally phase exactly as it was cast [AlSc20, S.280]. As such, vote verification mechanisms rely on the unique properties that digital technologies provide, being impossible to replicate in physical means since, once the vote is in the ballot box, it is detached from the voter and is impossible to retrieve to check its correctness [Will18]. Following Marky et al. [MZRR21], verification mechanisms are generally divided into three categories, depending on the scale of the verification: universal verification mechanisms (where anyone -especially externally designated electoral bodies or independent experts and observers - can verify that the result corresponds to published ballot, aiming to prevent a possible corrupt authority from faking the results [IRRR20]) and individual verification mechanisms (where voters can verify themselves that their ballots are cast and counted as intended); and voters eligibility verification. While universal and individual voter verification can be considered two different steps of the same process (individual allowing verifying that individual votes have been cast as intended and recorded as cast, and universal verification adding that the votes have been counted as recorded), voters eligibility verification allows proving that a voter the information that has been included in the register. The obvious goal of these three verification mechanisms is to increase trust in the system by providing elements that help identify problems, attacks, or malpractices that could damage the integrity of the electoral process.

This paper aims to focus on individual verification mechanisms and their impact on trust creation. While the theoretical foundations of the system are clear, and there is a general agreement on the need to use verification methods (an agreement that the authors of this paper subscribe to), the real impact of individual verification methods is not evident since they are not widely and systematically used by voters. Kulyk and Volkamer [KuVo18] defined the “five lacks” explanation to why voters are not actively using them: 1) lack of awareness of the risks of internet voting, 2) lack of concern about the real impact that those risks might entail, 3) lack of self-efficacy, the verification systems are not accessible enough for users, 4) lack of compulsion, the systems are not convenient enough, and voters do not make an effort to use them, and 5) lack of perseverance, voters interrupt its use after a certain number of elections. Expanding on these ideas, Solvak’s [Solv20] research concluded that verification mechanisms in Estonia are helping build confidence in risk-aware voting population, who are most likely using those systems not for voting verification purposes but to check against cyber threats. These findings are in accordance with the relatively low portion of voters verifying their votes election after election in Estonia, also referred to in Solvak’s work [Solv20], ranging between 3,4% in 2013 and 5,3% in 2019. To solve this situation, previous research focused on improving the information about verifiability systems received by voters [OlBV13] or the usability of verification systems [MKRV18] to increase their likelihood of using them. But still, the data persistently shows that citizens do not verify their votes.

Based on the previous, in this paper, we aim to challenge the assumption that individual verification mechanisms increase users' trust in i-voting. Instead, we posit that, in fact, verification mechanisms are distrust management systems. The aim of challenging this assumption is not to undermine the need to implement verification tools and promote their use but to discuss their impact on the overall creation process of trust and put in place some questions that can invite further reflection about them.

We sustain this discussion on the assumption that trust and distrust are concepts that, although closely related, are not symmetrically opposite ones, but they have their own dynamics and mechanisms of functioning. This idea is not new since it was theoretically already coined by Luhmann [Luhm79], expanded by Lewicki [LeMB98], and recently framed for its use in data-driven [DuCa23] and voting technologies [Duen22, Duen24]. We understand trust is a *social construct whose dynamics of creation cannot be fully grasped without also considering the dynamics of distrust* [DuCa23], understanding that trust and distrust have different patterns of functioning and can even coexist towards the same target. Norris, for example, claims that trust should go hand in hand with a certain level of distrust, leading to the need for rational and informed judgments and increased trustworthiness [Norr22].

Previous research assumed that verification mechanisms increase the existing trust but systematically reported the existing problems to justify such an assumption, heavily contrasting with the generalized belief that they serve as trust creators and necessary tools for improving the quality of elections. In contrast, we propose changing the angle and hypothesize that verification mechanisms are tools for managing distrust rather than creating trust directly. And we sustain this claim with two basic ideas. Distrust plays a central role in our societies and can be understood as a practice of active verification, oversight, and control of public institutions [Szto03]; in other words, distrust can be a valid mechanism for indirectly building trust [Duen24]. Internet voting verification mechanisms, hence, might follow this indirect path for trust creation by reducing the possible effect of distrust factors affecting a given system, following the principle that some of those who verify might be distrustful users and a positive impact of verification mechanism in reducing the capacity of raising distrust discourses against the system.

2 Method

This research employs a qualitative approach to explore verification mechanisms in Internet voting systems. The paper proposes an analysis of the narrations collected during the development of different case studies under the project ELECTRUST[3.] The data utilized for this research comprises a total of 31 semi-structured interviews with experts

[3] To know more: https://davidduenascid.cat/dynamics-of-trust-and-distrust-creation-in-internet-voting/

in the field, including cryptographers, cybersecurity experts, social scientists, politicians, vendors, and journalists, all with a high degree of specialization or knowledge of internet voting. This comes as a result of the fact that the interviews were conducted as part of the development of case studies in Estonia (16 interviews) in December 2022 and Australia (15 interviews) in July and August 2023, both countries having or having had internet voting systems in place. The number of interviews is over the average for this type of research, ensuring the validity of the opinions gathered by the number of participants and the position they represent. Unfortunately, due to privacy-related issues, the identity of the participants cannot be disclosed, and the reader will have to trust the best efforts of the researchers to gather the best possible informants for such research.

The interviews were audio recorded (around 34h of audio), transcribed, and further coded using N-Vivo. For this work, the codes related to verification processes were retrieved and analyzed.

	Australia	Estonia
Technical researcher	3	3
Politician		3
Vendor	1	2
Public Administration	3	3
Social researcher	6	3
Activist/journalist	2	2

Table 1. Interview profiles

3 Discussion

In this paper, we state that, following the findings in the current literature, individual verification methods do not have the expected impact on the generalized creation of trust in the electronic (internet) voting system. Differently, we defend that the effects on trust creation are indirect and targeted to specific collectives, and they occur via the mitigation of distrust.

To approach this statement, we created an ideal distribution of theoretical voters depending on two main variables: 1) do they trust or distrust the use of internet voting? 2) Do they verify or not their votes? This combination of factors allows creating an ideal typology of voters whose theoretical positions are described in Table 1. These theoretical positions are then compared with the discourses gathered in the interviews conducted in Estonia and Australia.

	Voters who trust i-voting	Voters who distrust i-voting

Voters who do verify	Concerned voters who trust and use internet voting and understand the need to verify their vote to ascertain that the vote was cast as intended. Using a verification mechanism will not change their trust in the system but reinforces their position, matching the “skeptical trust “profile proposed by Norris [Norr22]/	Concerned voters who use internet voting, even if distrusting it. They might use verification mechanisms searching for proof to justify their distrust in the system (e.g., problems with the verification tool or discrepancies with the vote cast), matching the profile described by Solvak [Solv20].
Voters who do not verify	Based on data on the usage of verification mechanisms, this group represents a vast majority of voters who, trusting and using internet voting, do not use verification systems. The reasons can span from the fact that they trust enough internet voting or the institutions providing the system, to not feeling the need to verify because they lack one or more of the “five lacks” proposed by Kulyk and Volkamer [KuVo18].	Voters who distrust i-voting and do not use verification mechanisms, either cause they are not using internet vote at all or because, in case they are i-voting, they do not want to use verification mechanisms.

Table 2. Ideal typologies of voters in relation to trust and verification

The interviews provided valuable references for the four ideal typologies of voters described above. Some extracts will be presented and commented on to validate them, serving as foundations for further discussion.

3.1 Trust and verify.

The interviews collected agree with the described typology that the voter who trusts and verifies is a person with a certain level of knowledge for whom the verification process makes sense from a technical perspective:

> *"I assume that the people who do verify are more tech-savvy anyway. And they anyway like trust technologies stuff more." (Est. 10)*

> *"we're talking here about people having a level of understanding that makes them feel comfortable that their vote is cast and counted as the vote that they actually put in" (Aus. 2)*

The use of verification systems is not adding trust to the already existing, and the use of verification is reduced to a way to test the technology while actually verifying the vote:

> *"The use of verification technology is very constant over time; it's not growing; it's being used by people who already have a trust level. So again, it doesn't seem to be actually picked up by people who have an issue. So, they don't use it to gain additional trust. But it's the other way around. High-trust people simply get another, I don't know, verification on top of that or simply play around with this technology." (Est. 8)*

In fact, the borders between the approach to verification mechanisms by the first and second groups of voters (either trusting or distrusting, but verifying) are pretty blurred. They might both be using verification mechanisms to check the tool's accuracy and if the system works as expected, just changing to the goal of this exploration. Still, most likely, the results obtained in the verification mechanism will not necessarily change their initial position regarding their trust or distrust in the system.

3.2 Distrust and verify.

This second ideal profile also includes voters with a high level of expertise who approach the verification critically to test if the verification system fails to provide the correct output. Paradoxically, this type of voter behavior does not preclude having a certain amount of trust in the technology. Still, it is sustained by a certain degree of distrust on the technological choices (the system is not good enough) or the management of the technology (those in charge are not skilled enough). The information collected in the interviews exemplifies the role of activists with a good/very good understanding of the technology.

> *"The server is basically a black box. And we know how the votes go in, we know that process until your votes go in, and then we know the result when they come up. What happens with the votes when they are being basically tabulated in the server? We have no way of knowing. So, to counter that problem, the Electoral Commission or the program sends you a verification notice that, yes, you voted for this person, and your vote has been taken and registered as a vote for this or*

> *that candidate. But, in fact, that proves nothing, because what you get is the program telling you that whom you voted for; it doesn't tell you how your vote came out or how it actually was counted. So that doesn't help us either. And then, as long as we don't know how those votes are being really tabulated or counted within the server, it's just an issue of faith." (Est. 5)*

> *"I'm pretty sure some of the activists said 'I can't trust the verification app because *Company* will make it', so maybe it always just tells you that your votes are right" (Aus. 6)*

> *"They didn't really fix the trust assumptions because you're still just using a *company* app to call something that's reading from *company* database. There was still this huge single point of trust failure, from one company providing all of the software for the verification process" (Aus. 1)*

3.3 Trust and do not verify.

The third ideal type of voter represents the vast majority of internet voting users, who do not feel appealed to use verification methods. The opinions collected in the expert interviews gather different perspectives with a common departure point: they trust the system, which relaxes their potential need for verification.

> *"I think that they are aware of the verification tools. Yes. I think that they are aware of them. I don't know how many people actually use them, but I think they know it's available. And if they don't use them properly, they trust the system. It's another short and small procedure. But, well, why waste time?" (Est. 4)*

> *"You said that it's just around 4-5% of people who are using it. Why do you think so little people are using? Maybe because of the trust they have in the system" (Est. 14)*

Another issue raised to justify the low usage rates relates to the added complexity of verification mechanisms, which entails a lower degree of convenience. This type of statement was often found in the experience with the internet voting system used in New South Wales, where the verification mechanism implied some more steps than the Estonian one.

> *"It's actually a little unclear why so little (people use it), because you could say that it is complicated, but is it really? No, you have to download an app and just aim it at the screen. I thought in 2021, 2022, or 2023, a lot of people would be able to do that." (Est. 7)*

> *"The verification system got a bit more complicated as time went on. In the 2019 election, it required scanning a QR code and having two different devices. And it did become a little more difficult to verify but it was still possible, and I think if the system had continued, we could have sort of made the verification process*

> *more lazier to use" (Aus. 19)*
>
> *"Verification was quite complex. When you voted, you got back a 32-digit receipt number and the way you could verify your vote would be to phone and enter that 32-digit receipt number, and if we got it correct, it would repeat back to you. But it was cumbersome. Hardly anybody used it and one of the problems with that is that it had very low… I think it's less than 5% verified. So, if you have 250,000 votes and you get less than 5% verified, it's quite difficult to actually make any form of statement around how well the system is performing" (Aus. 8)*

3.4 Distrust and do not verify.

The final group, also depicted in the interviews, gathers those voters who do not use internet voting and do not verify their votes. The main element highlighted by experts' interviews is that the problem relates to the general distrust in the system and institutions, and the use of verification mechanisms does not affect their perspective regarding trust in the system.

> *"if you add another technological measure, it will not affect the trust of those who distrust the system" (Est. 6)*
>
> *"the people who believe in conspiracy theories don't trust it or don't verify. But they don't verify because they don't trust technology" Est. 10*

This position is reinforced by the perspective of the same distrustful experts who use different arguments to legitimize their position. When relating to verification, the main element highlighted is the lack of capacity to directly observe the process and how that forces a trust transmission toward the management bodies dealing with the electoral process.

> *"I can't trust this; I can trust only things I can verify myself or I can send someone to verify (...) I don't know what the program actually does" (Est. 3)*
>
> *"Do you trust the Estonian internet voting system?*
>
> *No, I don't. Well, I don't think that voting can rest on belief. And as a matter of fact, we cannot observe or verify the process of e-voting in Estonia. So basically, what we're being told is the belief that it's honest. And would you believe paper voting if you could not observe and verify the counting process? I don't think anyone should really believe that the votes are being counted, honestly, we need to be able to see." (Est. 5)*

3.5 Then, for whom is the verification system?

The previously described ideal types of voters fit well in the narrations provided by experts, whose discourses reinforce the idea that the use of verification mechanisms does not directly influence increasing trust in any of the groups. Still, the connection between verification and trust seems solid in their narrations, *"even if, let's say, random citizens aren't verifying their votes (...) I like it; it definitely creates trust" (Est. 2).* This situation, accordingly, opens the door to questioning for whom the verification mechanisms are thought, and which is its actual impact. The information collected points out different directions.

First, verification mechanisms are helpful for electoral administration to ensure the integrity of the electoral process and the absence of massive attacks. Even with low usage rates, the data obtained allows for reporting mistakes and statistically ensures the process's correctness.

Report mistakes	*"A) We don't get any feedback as such, so if the result is not correct, then the voter will have to vote again or react.* *Q) And the reaction would be to report a mistake to you so that you could explore the reason for that?* *A) Yeah. All right." (Est. 7)* *"Some form of verification was, I understand, deemed to be important. So as to create trust somebody who's plugged their numbers into a keyboard. How can they be certain that the system has recorded their vote correctly? Well, they can go in again and check. And of course, if they've gone in again and checked and found that to their surprise, one in a million, the machine made a mistake, well, then they're able to change their vote" (Aus. 20)*
Statistical verification	*"But it's important that we have because this allows us to actually build trust by actually saying, conveying the message that no large-scale attacks were possible because we got actually mathematically 5% of the votes that were correct" (Est. 1)* *"This amount of people verify, they have a statistical certainty that they would then, through that detect large-scale attacks. So, which is actually giving all individuals some functionality, where the logic behind this is actually to observe on the aggregate level whether something is going on, completely detached from the individual motivations of using this, and that function would still stay there with these two different forms as*

	well" (Est. 8) *"Mathematically we are happy with 2% to detect large-scale attacks. That's the very reason for having that (...), large-scale attacks." Est. 13*

Table 3. Verification and Security

A second group of arguments relates to the need for the administration to have the system in place to avoid the problems that could arise if verification is not in place: 1) although most of the voters might not want to use it, verification needs to be in place for those who effectively demand it, 2) to avoid coercion related problems, 3) to create a generally more secure environment.

For concerned voters	*"It's not fixing that part where the problem is, what it simply ticks a box; we have verifiability, and if people want to, they can use this. So maybe it might have a positive effect simply by being there and not being used. But it definitely is not addressing this segment, where people actually have a problem (...) I'm pretty certain that the introduction of verifiability, individual verifiability, did not bring low trusting people into voting" (Est. 8)*
For coercion-related issues	*"The reason why we do not allow this is, again, coercion resistance, right? So, the second point is that, like 99.5% of people, if they verify, they do it during the first 20 minutes; we very clearly see this from the logs. So, for people who already want to verify, there is no reason to give a further time window. And the reason we restricted the time being there was that we consider this as an anti-coercion measure. So, for instance, you have less time to share your QR code with a coercer and stuff like that" (Est. 10)*
Security perception	*"I'm pretty certain that there are two aspects: they create real security, like how they create change perception. I think, real security-wise, they have a pretty strong effect. And they have some effect also in the space of perceived security. But certainly, the perceived effect is less than the actual effect" (Est. 12)* *"Most of the surveys of users suggest that they were quite*

	*satisfied with the process. It was slightly cumbersome because of the verification protocols which, you know, people like *activist* think are a good thing. You know, they are a good thing for security, but for an ordinary voter, they're sometimes cumbersome" (Aus. 9)*

Table 4. Verification and management of distrust

Finally, a last set of arguments relates to the perception of citizens regarding the use of the system, assuming that the use of verification mechanisms can be a long-lasting learning process in which the numbers will increase when voters learn about the need to verify the votes for their security and the overall integrity of the electoral process. For example, the sequence might be similar to accepting two-factor authentication in online banking.

> *"I believe that if people would understand the very meaning of verification, then definitely. So, we are saying that, hey, you can't trust a computer; there are means to check whether your computer behaves well. (...) I'm just the machine. I might be faulty; I might be infected; I might have the flu. Please check with me. And if you have the means to check it. For the voter itself, it should increase confidence in this way. The message has to be right: I'm just a machine. I'm sorry." (Est. 13)*

> *"I think it's a ripple effect (...) All this technology is slowly rippling out. Building trust with sophisticated things like voter verification and two-factor authentication and all this sort of stuff is slowly rippling out. Still, I'm not sure when it will reach someone like my grandmother, for example (...) Is slowly having an effect. Now, most banks require some sort of two factor or sort of authorization, right? They'll send, they'll send an SMS to your mobile phone and most users. Would now be probably a little bit suspicious if the online voting system did nothing, right?" (Aus. 12)*

4 Conclusions

The obvious goals of Individual verification mechanisms are twofold: 1) they aim to provide tranquility to the voter by giving an indicator that the vote that is finally counted is the same that they intended to cast, and 2) they aim to provide certainty that the overall electoral results are correct by ascertaining the lack of problems in the individual verification reports. Both goals logically relate to creating a more trustworthy electoral environment, assuming this means generating trust in the process. Under the assumption that research repeatedly discussed the efficacy of verification mechanisms for building trust amongst voters, we decided to look at the other side of the coin and approach verification mechanisms as tools that help mitigate distrust (which, indirectly, does contribute to creating a more trustworthy environment).

The qualitative data collected in our interviews reinforces such an idea. The opinions gathered from experts during the development of case studies in Estonia and New South Wales (Australia) coincide with the assessment that verification mechanisms are differently approached depending on the type of user. While for some of them, it is a valuable tool, most voters do not use verification mechanisms. Some of them do not use it as a result of their already existing trust in the system and might not understand the need for verifying; some others cause they distrust the system (and, to some extent, probably the overall institutional environment where it is immersed) and this distrust transitions to the elements supporting internet voting, in our case the verification mechanisms. Still, there is a general agreement, seemingly contradictory to the previous ideas, that verification mechanisms are building trust in general and, therefore, are necessary mechanisms.

Following our hypothesis, our research points out that some of the impacts of verification mechanisms are more related to mitigating possible sources of distrust than to the direct creation of trust. Verification mechanisms are described as elements providing security because the electoral administration can use the available data to ascertain the integrity of elections by allowing the detection of large-scale intrusions in the system. This positions verification systems as a background security measure that would be necessary in situations of emergency. One of our interviewees brilliantly compared it with a car's airbag in place: under normal conditions, it remains unnoticed, but in case of need, you would be happy to have it.

This connects with a second important distrust management element: verification tools, which are essential elements for those who can understand in detail the insights of how the system works: the expert community. A quick overview of the recent history of electronic and internet voting will point out that some of the main reasons why some systems have been discontinued in the past (e.g., in the Netherlands [Loeb16] or Switzerland [HaPT22]) relate to the presence of (legitimate) distrust discourses provided by the community of experts. This conclusion dialogues with the findings of Solvak [Solv20], who targets tech-savvy voters as those for whom verification mechanisms produce some trust-related impact. If we assume that, for many voters, expert discourses are an important source of trust creation, using an expert-approved verification mechanism reduces the possibility of facing distrust-related discourses that could spill over into the general voters' perception of the adequacy of internet voting.

We could speculate on how this situation would change if voters massively adopt verification mechanisms and expect that then it could serve to generate trust at a larger scale (under the assumption that voters understand the sense of verification). Still, in the current scenario, verification serves as a necessary tool for validating the integrity of elections and to build trust indirectly by helping manage possible distrust.

Finally, we would like to develop some policy suggestions stemming from the reflection proposed in this document. While it is commonly accepted that the concepts of trust, confidence, and integrity can bear different meanings and are subject to distinct academic definitions, they are often interchangeably used in election-related literature as well as national legislation. As we have discussed earlier in the paper, the necessity for the internet voting systems to contain mandatory verification mechanisms (both individual and universal) as a pre-condition for building trust is a widely established international standard and most often a legal requirement in the respective national legislation or regulation where i-voting systems are used[4]. As such, the broader interpretation of the notion of verification is not only related to the casting (individual verification) or counting and tabulation of votes (universal verification) but also independent or third-party verification (and for that reason, audit and certification) of any i-voting related systems and processes should be part of the relevant national provisions.

However, the concept of trust is significantly broader and depends on multiple aspects and layers, often unrelated to technological factors. The good national and international practice has shown that strategies and regulations that are holistic in nature and relate to broader aspects of the election process (for example, those addressing casting and counting of votes, the work of election management bodies, voter registration processes, transparency and accountability of institutions, campaign activities and behavior of electoral contestants, etc.) as well as target various groups of voters are necessary for building public trust - independently of the technology used for casting votes. By that token, the elements that can erode public trust in elections are multiple and continuously growing, and as such, the states' approaches used for building trust in other aspects, such as those dealing with misinformation narratives or raising public voter information campaigns or cybersecurity awareness, should be paired with those for managing distrust in the voting systems and continue actively promoting various verification opportunities (individual, universal or independent third party mechanism).

Acknowledgments

The work of David Duenas-Cid has been funded by the Polish National Research Center's grant OPUS-20 -2020/39/B/HS5/01661 and EU H2020 MSCA Program, grant agreement no. 101038055.

AI was used in this text to review grammar. The authors take full responsibility of the content of this document.

[4] See for example CoE Recommendation CM/Rec(2017)5 of the Committee of Ministers to member States on standards for e-voting (available at https://rm.coe.int/0900001680726f6f)

Bibliography

[AlSc20] ALSADI, MOHAMMED ; SCHNEIDER, STEVE: Verify My Vote: Voter Experience. In: KRIMMER, R. ; VOLKAMER, M. ; BECKERT, B. ; DRIZA MAURER, A. ; DUENAS-CID, D. ; GLONDU, S. ; KRIVONOSOVA, I. ; KULYK, O.; U. A. (Hrsg.): *Fifth International Joint Conference on Electronic Voting E-Vote-ID 2020*. Tallinn : Taltech Press, 2020, S. 280-296

[DuCa23] DUENAS-CID, DAVID ; CALZATI, STEFANO: Dis/Trust and data-driven technologies. In: *Internet Policy Review* Bd. 12 (2023), Nr. 4

[Duen22] DUENAS-CID, DAVID: A theoretical framework for understanding trust and distrust in internet voting. In: KRIMMER, R. ; VOLKAMER, M. ; DUENAS-CID, D. ; GERMANN, M. ; GLONDU, S. ; HOFER, T. ; KRIVONOSOVA, I. ; MARTIN-ROZUMILOWICZ, B. ; U. A. (Hrsg.): *E-Vote-ID 2022 Proceedings*. Tartu : University of Tartu Press, 2022, S. 57–62

[Duen24] DUENAS-CID, DAVID: Trust and distrust in electoral technologies: what can we learn from the failure of electronic voting in the Netherlands (2006/07). In: DUENAS-CID, D. ; LIAO, H.-C. ; MACADAR, M. A. ; BERNARDINI, F. (Hrsg.): *25th Annual International Conference on Digital Government Research (DGO 2024), June 11–14, 2024, Taipei, Taiwan*. Taipei : ACM , 2024, S. 1–9

[FiJö22] FITZPATRICK, JASMIN ; JÖST, PAULA: “The High Mass of Democracy” —Why Germany Remains Aloof to the Idea of Electronic Voting. In: *Frontiers in Political Science* Bd. 4 (2022)

[HaPT22] HAINES, THOMAS ; PEREIRA, OLIVIER ; TEAGUE, VANESSA: Running the Race: A Swiss Voting Story. In: KRIMMER, R. ; VOLKAMER, M. ; DUENAS-CID, D. ; RØNNE, P. ; GERMANN, M. (Hrsg.): *Electronic Voting. E-Vote-ID 2022*. Cham : Springer, 2022, S. 53–69

[HMVW16] HEIBERG, SVEN ; MARTENS, TARVI ; VINKEL, PRIIT ; WILLEMSON, JAN: Improving the Verifiability of the Estonian Internet Voting Scheme. In: *Electronic Voting. E-Vote-ID 2016.* : Springer, Cham, 2016, S. 92–107

[IRRR20] IOVINO, VINCENZO ; RIAL, ALFREDO ; RONNE, PETER B. ; RYAN, PETER Y. A.: Universal Unconditional Verifiability in E-Voting without Trusted Parties.

In: *2020 IEEE 33rd Computer Security Foundations Symposium (CSF)* : IEEE, 2020 — ISBN 978-1-7281-6572-1, S. 33–48

[KuVo18] KULYK, OKSANA ; VOLKAMER, MELANIE: Usability is not Enough: Lessons Learned from 'Human Factors in Security' Research for Verifiability. In: KRIMMER, R. ; VOLKAMER, M. (Hrsg.): *Third International Joint Conference on Electronic Voting (E-Vote-ID 2018)*. Bregenz : TUT Press, 2018, S. 66–81

[LeMB98] LEWICKI, ROY J ; MCALLISTER, DANIEL J ; BIES, ROBERT J: Trust and Distrust : New Relationships and Realities. In: *Academy of Management Review* Bd. 23 (1998), Nr. 3, S. 438–458

[Loeb16] LOEBER, LEONTINE: E-Voting in the Netherlands; from General Acceptance to General Doubt in Two Years. In: *3rd International Conference on Electronic Voting 2008*. Bregenz : Gesellschaft für Informatik, 2016 — ISBN 9783885792253, S. 21–30

[Luhm79] LUHMANN, NIKLAS: *Trust and Power*. Chichester : Wiley-Blackwell, 1979

[MKRV18] MARKY, KAROLA ; KULYK, OKSANA ; RENAUD, KAREN ; VOLKAMER, MELANIE: What Did I Really Vote For? On the Usability of Verifiable E -Voting Schemes. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA : ACM, 2018 — ISBN 9781450356206, S. 1–13

[MZRR21] MARKY, KAROLA ; ZOLLINGER, MARIE-LAURE ; ROENNE, PETER ; RYAN, PETER Y. A. ; GRUBE, TIM ; KUNZE, KAI: Investigating Usability and User Experience of Individually Verifiable Internet Voting Schemes. In: *ACM Transactions on Computer-Human Interaction* Bd. 28 (2021), Nr. 5, S. 1–36

[Norr22] NORRIS, PIPPA: *In Praise of Skepticism: Trust but Verify*. Oxford : Oxford University Press, 2022

[OlBV13] OLEMBO, MAINA M. ; BARTSCH, STEFFEN ; VOLKAMER, MELANIE: Mental Models of Verifiability in Voting. In: HEATHER, J. ; SCHNEIDER, S. ; TEAGUE, V. (Hrsg.): *E-Voting and Identity. 4th International Conference, Vote-ID 2013*. Guildford : Springer, 2013, S. 142–156

[Osce13] OSCE / ODIHR: *Handbook For the Observation of New Voting Technologies*, 2013 — ISBN 9789292348694

[Osce24] OSCE ODIHR: *Handbook for the Observation of Information and*

Communication Technologies (ICT) in Elections. Warsaw : OSCE-ODIHR, 2024

[Solv20] SOLVAK, MIHKEL: Does Vote Verification Work: Usage and Impact of Confidence Building Technology in Internet Voting. In: KRIMMER, R. ; VOLKAMER, M. ; BECKERT, B. ; KÜSTERS, R. ; KULYK, O. ; DUENAS-CID, D. ; SOLVAK, M. (Hrsg.): *Electronic Voting. E-Vote-ID 2020*. Cham : Springer, 2020, S. 213–228

[Szto03] SZTOMPKA, PIOTR: *Trust: A Sociological Theory*. Cambridge : Cambridge University Press, 2003

[VoSD11] VOLKAMER, MELANIE ; SPYCHER, OLIVER ; DUBUIS, ERIC: Measures to establish trust in internet voting. In: *ACM International Conference Proceeding Series* (2011), S. 1–10 — ISBN 9781450307468

[Will18] WILLEMSON, JAN: Bits or paper: Which should get to carry your vote? In: *Journal of Information Security and Applications* Bd. 38 (2018), S. 124–131

Track 3: Election and Practical Experiences

Online Voting: Social Elections in Germany

Carsten Schürmann[1]

Abstract: In 2023, Germany, for the first time in history, offered online voting over the Internet in form of a pilot project as an alternative to postal voting for the 2023 social elections (in German "Sozialwahl"). In this paper, we describe intentions, political processes, implementation, and also the lessons learned from this pilot project, which might be of interest for other election management bodies wishing to offer Internet Voting as an alternative voting channel.

1 Introduction

In 2023, more than 22 million voters were able to cast their vote online for the first time in Germany for the 2023 social elections. The online elections took place at five statutory health insurance funds. Social elections in Germany are traditionally held using postal voting only. For over two decades, there have been discussions about supplementing the social elections postal voting with the option of online voting. Voters should be able to decide whether they want to cast their vote by post or online. The demand for the introduction of online voting in social elections was raised primarily by the respective federal election commissioners and the social insurance institutions.

At first, the idea encountered widespread skepticism from politics in part because of the decision of the constitutional court [BV09]. Over the years, however, due to the success of Internet Voting pilot projects in other countries, the political establishment started warming up to the idea of online voting in Germany, although, moving forward, it is unlikely that online voting will be offered at political elections any time soon [ESBHL23]. The Federal Election Commissioner for the 2017 social elections and her deputy - Rita Pawelski and Klaus Wiesehügel - were ultimately able to convince the members of the German Bundestag to allow a pilot project for the 2023 social elections. As part of this pilot project, five statutory health insurance funds would be able to offer online voting as an additional voting channel to postal voting. This pilot project was carried out for the 2023 social elections and supported by the current Federal Election Commissioner for the 2023 social election and his deputy – Peter Weiß and Doris Barnett. This paper summarizes the author's experiences as an academic and independent expert, working with the Federal Election Commissioners to assess the verifiability of the election outcome.

This paper is organized as follows. In Section 2, we describe the organizational aspects of the social elections before we summarize some facts and statistical information in Section 3. Next, we go into the details of the design and implementation of the voting protocol in

[1] Center for Information Security and Trust , IT University of Copenhagen , Denmark,
carsten@itu.dk, https://orcid.org/0000-0003-4793-0099

doi:10.18420/e-vote-id2024_09

Section 4. In Section 5, we analyze the pilot project and describe lessons learned, before we conclude in Section 6.

2 Organizational Aspects

The legal basis for this pilot project was an amendment to the Fifth Book of the German Social Code [So88, § 194a to § 194d] and an online voting ordinance issued by the Federal Ministry of Health [Bu20]. These two documents form the legal basis for an in depth technical guideline issued by the Federal Office for Information Security (BSI) [Bu23]. An analysis of this technical guideline in the broader setting of election security can be found in [Be21].

Each statutory health insurance fund is self-governed by an elected administrative board consisting of representatives of the insured and their employers. The concept of the pilot project envisaged that the representatives of the insured - not the employers - could be elected through online voting in addition to postal voting. In contrast to political elections, individuals do not stand as candidates in social elections. Voters can only vote for candidate lists. All statutory health insurance funds that amended their statutes accordingly by September 30, 2020 were eligible to participate in the pilot project. The detailed requirements for the social elections are outlined in [Bu23].

Fifteen statutory health insurance funds decided to take part in the preparation of the online voting pilot project after the amendment to the statutes had been made in due time. The Fifth Book of the German Social Code [So88, § 194a] stipulates that the participating health insurance funds had to form a joint working group in order to prepare and conduct the online election. All fifteen health insurance funds prepared the pilot project, but ultimately only five conducted an online election: Techniker Krankenkasse, BARMER, DAK-Gesundheit, KKH and hkk. This meant that 22.44 million eligible voters were able to opt in to online voting. The working group commissioned two service providers to implement online voting for the social elections. the IT service company regio iT[2] and Smartmatic[3] to build and deliver the online voting system.

For the provision and operation of the online voting system, the common IT security standards of the German Federal Office for Information Security on information security management systems [Bu17a] had to be applied, including the guidelines for reporting security incidents [Bu21], the IT-Grundschutz methodology and risk management (BSI IT-Grundschutz) [Bu17b], as amended. All documents are available in the English language and can be downloaded from the webpage of the BSI.

The BSI issued a pilot-specific technical guideline for implementing the pilot project [Bu23] that the statuary health insurance funds and online service providers used as the de facto

[2] See https://www.regioit.de/

[3] See https://www.smartmatic.com/

standard when preparing and conducting the online social elections and also to verify independently the election result. The participating health insurance funds developed security and emergency response concepts for the online elections together with the online service providers. In accordance with the regulations, the health insurance funds reviewed the correct operations of the online voting system using a test case catalog and hired independent expert to conduct an external review. The online election administrators were appointed by the election committees of the respective health insurance funds, many from their respective IT departments.

The Federal Election Commissioner for social elections published a template for informing eligible voters about online elections. Important key points for this announcement (No. 14) were contained in [So88, § 194b]. The five electoral health insurance funds produced instructions on online voting, in which they presented all the necessary information.[4] Online ballot papers had to look identical to and behave like postal ballot papers. The use of hyper-links on the voting page were explicitly prohibited. If they chose to do so, voters were allowed to submit an *invalid* vote, where none or more than one candidate list was selected. No validity checks on the ballot was to be conducted before a ballot paper was submitted. Eligible voters could submit one and only one ballot. Re-voting was not permitted in contrast to Estonian Internet elections, where it is.[5] According to the technical guideline [Bu23], voters must have the opportunity to check whether their vote had been recorded-as-intended in the electronic ballot box.

The criteria to determine if the online voting solution is ready for use are described in Section 9 of the Online Election Ordinance [Bu20]. These criteria include that the digital ballot box is empty, ballot designs and other information are correct and immutable, logging is activated, and that the voting system is activated by two members of the respective election committee. Furthermore, the in Section 13, the ordinance defines that online votes overwrite postal votes: the online service provider is required to transmit the voter identification numbers of all online voters to postal voting authorities. In the case double voting, the postal vote is discarded and the online vote is counted [So88, §194b]. In accordance with [So88, §194b], the online election results and the postal vote results were determined separately and then combined to produce the overall result. After the election result is approved, any electronic voting data related to the election is destroyed at the same time as ballot papers and envelopes.

Online voters could choose between two authentication methods: either by the three number method, where voters entered three numbers from the front and back of the health insurance card as well as the election identification number printed on the ballot envelope to authenticate to the online voting system, or by using the electronic ID card (ID card app2) in conjunction with the election identification number.

[4] See, for example, https://www.dak.de/dak/unternehmen/sozialwahl-bei-der-dak-gesundheit/sozialwahl-2023-erstmals-auch-als-online-wahl_33916

[5] https://www.valimised.ee/en/internet-voting/documents-about-internet-voting

Health insurance company	online voters	Percentage of voters
Techniker Krankenkasse	200,080	9.96 %
BARMER	92,577	5.91 %
DAK-Gesundheit	22,208	2.42 %
KKH	10,267	3.79 %
hkk	9,034	5.90 %
TOTAL	334,166	6.74 %

Fig. 1: Proportion of online voters

Health Insurance	Valid Votes	Verifications	in %	Duplicates	Invalid Votes
KKH	10.267	1.048	10.2	0	114
HKK	9.034	1.274	14.1	1	53
TK	200.080	19.948	10.0	1	2.111
DAK	22.208	2.569	11.6	0	246
Barmer	92.577	6.658	7.2	0	2.013

Fig. 2: Summary of Votes Cast during the Social Elections 2023

The costs of the pilot project are still being determined as of May 2024. In accordance with [So88, §194a], the costs of this pilot project will be allocated to all statutory health insurance funds. The SHI Financial Stabilization Act of November 2022 specified the apportionment procedure. According to this procedure, in particular the costs for the tender, the commissioning of external services including the costs for scientific and technical advice as well as the material and personnel costs of the participating health insurance funds, if they carried out tasks in coordination with the working group, may be apportioned on the basis of the project planning (for the preparation and implementation of voting via online voting). The Federal Ministry of Labour and Social Affairs determines the apportionment amounts. All statutory health insurance funds must contribute to the costs of the pilot project in proportion to their share of the potentially eligible members.

3 Facts and Statistics

The online election began on April 11, 2023 and ended on May 31, 2023. The proportion of people voting online was 6.56 %, a detailed statistics for the different health insurance funds is depicted in Figure 1. Online voter participation ranged between 2.42% and 9.96%. The totals of votes cast and verified, including the number of votes marked as invalid, are depicted in Figure 2. The rate of voters who vote both online and by post was less than 1 percent. The pilot project was supported by the Federal Ministry of Health and is being evaluated in agreement with the Federal Ministry of Labor and Social Affairs. This evaluation will be completed in the Fall of 2024.

4 Technical Aspects

In this section, we elaborate on the technical aspects of the online voting system for the social elections and the consequences for determining the verifiability of the election result. The technical guideline TR-03162 [Bu23, Section 5.2] requires that the social elections are verifiable, in the following sense:

1. Recorded-as-Intended: Voters can verify that their votes were received and stored as intended (Requirement 1).
2. Counted-as-Recorded: Voters can verify that their *individual* votes were correctly included in the count (Requirement 2) and that *all* votes were included in the count (Requirement 3).

4.1 ElGamal Cryptosystem

The privacy of the vote is implemented using the ElGamal cryptosystem. A ballot consists of several vote options for different candidate lists and each vote option is either selected or not by a vote choice $v \in \{0, 1\}$. ElGamal is based on the choice of a multiplicative finite group $\mathbb{G}$ of order q and a generator element $g \in \mathbb{G}$, such that $\mathbb{G} = \langle g \rangle$.

The ElGamal crypto system consists of three algorithms, the generation of the election private key $sk = \langle \mathbb{G}, q, g, x \rangle$ uniformly chosen by from $\mathbb{Z}_q$ and the corresponding public key $pk = \langle \mathbb{G}, q, g, h \rangle$, where $h = g^x$, an encryption algorithm, $\mathsf{enc}_{pk}(m) = (g^r, h^r m)$, where r is some ephemeral random bit string, also called a random coin.

And finally there is a decryption algorithm $\mathsf{dec}_{sk}(a, b) = a^{-x} b = m$, which, in the setting of the social elections, is only applied to the final election tallies (for each candidate lists). Decryption is accompanied by a cryptographic proof of correct decryption δ, which allows us to verify with all but negligible certainty, that m is the correct decryption of (a, b), not knowing the secret key! More formally we write $\mathsf{verify}((a, b), m, \delta)$.

From a cryptographic point of view, it is important that r is ephemeral, meaning that it is generated for purpose of encryption, and forgotten right away. If r is not ephemeral, it is possible to decrypt a message: Let (a, b) the encryption of a vote m, where the random coin r is published and known. Then $h^{-r} b = h^{-r} h^r m = h^{r-r} m = h^0 v = m$, meaning that it is possible to decrypt the encrypted vote *without* knowing the secret key x. Note also that in this case ElGamal is no longer secure against chosen plaintext attacks, because a probabilistic polynomial-time adversary can distinguish between ciphertexts.

4.2 Threshold Encryption Scheme

The election keys were created using a three-out-of-five threshold scheme: Private and public key shares were generated and stored on five smart cards, respectively, and those five cards were given to five members of the election committee for safe-keeping. The election private key was never generated, instead, the partial decryptions of election result where executed on each smart card. The election public key was then computed jointly from the public key shares. A special hardware, called the stand-alone VIU-machine, was used to orchestrate the interactions between the smart cards. For decryption of the election result, it was sufficient to present only three-out-of-five smart cards. The key size was chosen to be consistent with the BSI guidelines [Bu23].

4.3 Electronic Ballot Box

The electronic ballot box, also referred to as the *bulletin-board* in the literature, is based on a Merkle-tree datastructure. There is a root hash for the empty ballot box, and appending a vote to the bulletin-board also means to append the hash of the current bulletin-board. The verifiability of the consistency of the ballot box is effectively reduced to the verifiability of the root hash value.

4.4 Recorded-as-Intended Verifiability

For the record-as-intended verifiability, the vendor provided a "Sozialwahl Verifier" app for mobile phones that voters' could use up to 30 minutes after voting. Once the vote was confirmed and transferred to the electronic ballot box, a QR code was displayed on the screen that could then be scanned with the Sozialwahl Verifier. The app downloads the vote from the electronic ballot box, decrypts it using the information contained within the QR code and displays the vote in plaintext in the app for user verification.

To achieve this, the implementation of the online voting solution releases the ephemeral random coin (see Section 4.1) that was used to compute the ElGamal ciphertext in form of a QR code, reducing the levels of its ephemerality. It is displayed on the voting device after the vote has been submitted ready to be scanned by the verifier app. The app retrieves the ballot (a, b) from the digital ballot box (bulletin board) and decrypts it using the election public key h by computing bh^{-r}. The narrative arguing for why security is preserved is based on two arguments: (1) the ephemeral random coin is not shared with third persons, only the voter, and (2) in the case that the ephemeral random coin has been leaked to a third person, the encrypted votes are stored securely in a database protecting using standard security principles and security policies. One policy is that voters have access to their votes for up to 30 minutes after the vote was cast. Neither argument is cryptographically satisfying. During the recorded-as-intended verification, only the content of the vote is displayed in the app, other information, such as the hash of the current state of the digital ballot box, remains hidden. Assuming that voters keeps track of their encrypted vote, in theory, the voter could verify that the vote was included in the count (cast-as-recorded), by participating in the public universal verification effort that we describe in Section 4.8.

4.5 Vote Representation

A ballot for the social elections consists of k candidate lists. The voter is asked to vote for at most one list but may submit an invalid ballot if so desired. To ensure the verifiability of the representation of the ballot, digital signatures and zero-knowledge proofs are used when storing encrypted ballots in the ballot box. Here pk is the election public key. Besides k encrypted vote options $c_1 = \mathsf{enc}_{pk}(v_1), \dots, c_k = \mathsf{enc}_{pk}(v_k)$ there are also k disjunctive zero-knowledge proofs [CDS94] $\pi_1, \dots, \pi_k$ included in the online ballot, as well as a range proof [Ch24] ϱ that attests that the sum of all vote choices $0 \leq \sum_{i-1}^{k} c_i \leq 1$. In the general, case, the online-ballot $b = (c_1 \dots, c_k, \pi_1 \dots \pi_k, \varrho)$ is submitted with a signature $\sigma = \mathsf{sign}_{sk'}(b)$, where sk' is the private signing key of each voter, generated by the online voting system on behalf of the voter, and shared during authentication.

4.6 Vote Collection

Voters can submit at most one vote. Votes $(c_1, \dots, c_k, \pi_1, \dots, \pi_k, \varrho)$ are stored in the electronic ballot box together with a digital signature σ and processed further after polling has closed. At this time, the signature and all zero-knowledge proofs are verified. If all zero-knowledge proofs are verified, including the range proof, the vote is considered valid and is admitted to the count. If the verification of the signature or any of the zero-knowledge proofs fails then the vote is discarded.

This stands in contrast to other countries, where voters may re-vote as many times as they wish and only the last vote counts, in order to mitigate voter coercion. Re-voting makes the tallying process more difficult, since the overwritten votes must be purged from the ballot box in a verifiable way as they must not be counted. This was a political choice, and to a certain extent also justified, because voter coercion for online voting is not worse than voter coercion for postal voting, which are the only two voting channels offered during the social elections.

4.7 Homomorphic Tallying

The election committee for each statuary health insurance fund conducted their own counting process. The count was implemented as a homomorphic tallying of all votes cast. As re-voting was not available to the voter, no duplicate votes had to be removed from the electronic ballot box before counting started. During counting, the invalid votes were disregarded in the count (see column invalid votes in Figure 2). The count resulted in a ciphertext for each candidate list encrypting the respective total number of votes cast.

The tallying in the voting protocol used in the social election is implemented using the homomorphic encryption property of ElGamal encryption, which says that multiplying n

encrypted vote options is the same as the encryption of the product of n encrypted vote option

$$\prod_{i=1}^{n} \mathsf{enc}_{pk}(m_i) = \mathsf{enc}_{pk}(\prod_{i=1}^{n} m_i).$$

A simply trick allows us to replace the product in the right hand side of the equation by a sum, which is needed to compute the result of an election: $m_i = g^{v_i}$, where v_i can take only two values: 0 or 1.

$$\mathsf{enc}_{pk}(\prod_{i=1}^{n} m_i) = \mathsf{enc}_{pk}(\prod_{i=1}^{n} g^{v_i}) = \mathsf{enc}_{pk}(g^{\sum_{i=1}^{n} v_i})$$

Given set of encrypted ballots B where each ballot has a valid digital signature, we can now compute the encrypted result d_j for each vote option $1 \leq j \leq k$, as follows

$$d_j = \prod_{\substack{(c_1,\ldots,c_k,\pi_1,\ldots,\pi_k,\varrho) \in B \\ \bigwedge_{i=1}^{k} \mathsf{verify}(c_i,\pi_i) \qquad (*) \\ \mathsf{verify}(c_1,\ldots,c_k,\varrho) \qquad (**)}} c_j \tag{1}$$

followed by a decryption step, we obtain

$$e_j = \mathsf{dec}_{sk}(d_j)$$

and the corresponding proof of correct decryption δ_j. and finally by computing the discrete logarithm, we obtain that the result for vote option j:

$$r_j = \mathsf{dlog}_g e_j.$$

In general, computing the discrete logarithm for a finite group is hard, but as the election result is bounded by the number of votes, and these numbers are much, much smaller than the order q of the group $\mathbb{G}$, it is possible to compute the result r_j by brute force.

4.8 Counted-as-Recorded Verifiability

Counted-as-recorded verifiability of the correct operation of the online voting system and the accuracy of the result was conducted. For this, the following data files were made available.

1. the standard log files of the web server that collects Internet votes,
2. the database of public voter keys,
3. the ballot box B of votes $(c_1, \ldots, c_k, \pi_1, \ldots, \pi_k, \varrho)$ and the corresponding digital signature σ, created with the voter's private key,
4. the hash-chain of the ballot box, meaning that every submitted ballot is stored with the hash value of the current state of the digital ballot box,
5. the encrypted results $(d_1, \ldots, d_k)$ of homomorphic tallying,
6. the unencrypted results $(r_1, \ldots, r_k)$,
7. and the proof of correct decryption $(\delta_1, \ldots, \delta_k)$.

To execute the *verifiability checklist*, we would need

1. to extract all of the "vote receivedëntries and related information from the log files to check that they coincide with the number of votes stored in the ballot box B,
2. to rebuild the hash-chain with independent software to verify that the ballot box was not modified between the time when it was constructed and the time when it was verified,
3. to verify for all $(c_1, \ldots, c_k, \pi_1, \ldots, \pi_k, \varrho) \in B$ that (*) and (**) holds,
4. to verify for each ballot in the ballot box B the digital signature σ: $\mathsf{verify}(b, \sigma)$,
5. to rerun the homomorphic tallying on the ballot box of all k races $(d'_1, \ldots, d'_k)$ and verify that the result matches $(d_1, \ldots, d_k)$,
6. to verify the proofs of correct decryption: For all $1 \leq j \leq k$: $\mathsf{verify}(d_j, g^{r_j}, \delta_j)$.

Some of the verifiability steps provide mathematical guarantees, such as checking a proof of correct decryption. These guarantees vouch for that once verified, there is only a negligible chance that the integrity of the vote was broken or the election outcome was not correct. Other steps of the verifiability argument require a verifying re-computation. For example, the rebuilding of the hash chain in Step 2 and re-running the homomorphic tally in Step 5.

The statuary health insurance funds made relevant technical and election data available for public inspection for a period of one month after the day of the public announcement of the final election result in order to provide some version of universal verifiability to interested parties. No one took advantage of this opportunity. It is not clear what efforts where undertaken to encourage the public to participate in this effort.

4.9 Privacy checking

Vote privacy depends on election committee members not colluding. It would be possible to decrypt individual votes with access to the list of encrypted votes and three-out-of-five smart cards.

5 Lessons Learned

In this section, we summarize the practical experiences of checking the verifiability of the social elections.

5.1 Recorded-as-Intended Verification

The legal framework requires that voters are allowed to cast invalid ballots, online and on paper. Recall that an *invalid* vote is a vote where either less or more than one candidate list is chosen. This is implemented in the online voting solution as votes whose range proofs fail. Unfortunately, the way invalid votes were represented in the online voting solution in combination with and oversight in the recorded-as-intended verification mechanism meant that invalid votes could not be distinguished from votes that were incorrectly recorded or maliciously modified in the electronic ballot box. A voter could submit and verify the vote using the recorded-as-intended verification mechanism, but still the vote could be flagged as invalid and not included in the count, just because the corresponding range proof ϱ does not verify. Faulty range proofs could be the result of programming mistakes or malicious attempts to render votes invalid. This means that invalid votes recorded in the right-most column of Figure 2 might have been intentionally submitted as valid but were eventually rejected because of a faulty range proof.

5.2 Concurrency

Modern distributed and concurrent implementations of software systems are difficult to get right. If a program is not hardened against concurrency issues unwanted behaviors might ensue, like race conditions, where by unlucky interleaving of different program threads, computation might result in wrong results. Although highly problematic, concurrency issues are difficult to eradicate.

The online voting system does not admit re-voting and therefore does not provide a mechanism for removing multiple votes of the same voter. During the execution of the online voting system for the social elections, two duplicate ballots were found in the electronic ballot boxes of two statuary health insurance funds, respectively, after the polls closed.

Their existence could allegedly be traced back to a concurrency issue. Two voters managed to submit two identical ballots within a short time span, possibly by double clicking the ßend"button, a program behavior that should have been ruled out by the specification. No malicious intent could be identified.

Although it is possible to identify and remove the duplicate ballot in the ballot box, it is not advisable to do so, because Step 2 of the verifiability checklist in Section 4.8 would inadvertently fail, since the hash value of the final ballot box would not align. A decision was made to let the duplicate ballot remain in the ballot box, hoping that it would not make a difference in the final election result.

5.3 Determinism

Recall that the design of the online voting system encrypts every vote option on a ballot individually. The tally for vote option j is denoted as d_j. As it turns out, Step 5 of the verifiability checklist could also not be verified, because re-computing the d_j's would lead to different results: The reason is that instead of implementing the formula depicted in Equation (1), the implementation of the voting solution initializes the d_j with an encryption of zero (empty) $z_j = \mathsf{enc}_{pk}(g^0)$ and then iteratively computes the product of all vote options for candidate list j.

$$d_j = z_j \cdot \prod_{(c_1,\ldots,c_k,\pi_1,\ldots,\pi_k,\varrho) \in B} c_j$$

This multiplication with an encryption of zero is tantamount to re-encrypting the tallies, a step that is not verifiable. Therefore, Step 5 of the verifiability checklist described in Section 4.8 will inadvertently fail with high probability.

Verifiability could still be satisfactorily demonstrated, because vendor, election committees and working groups worked together to rerun the entire decryption with an patched tool implementing Equation (1) correctly. The authenticity of all input files was verified using hash values, and the renewed decryption of the encrypted tallies required three-out-of-five smart cards according to the threshold scheme used. It could be shown, that the election results for the second decryptions of the five respective elections were the same as the ones computed on election night, and furthermore, it could now be shown that the newly generated evidence supported the correctness of the result.

5.4 Public Participation in the Verification Effort

The statistics for and recorded-as-intended verifiability are depicted in Figure 2 shows the number of valid votes, verifications, duplicates, and invalid votes for each respective social election. Individual verifiability was exercised by 7.2% to 14.1% of all voters, which was higher than in Estonian elections [6] where verification rates range between 3.4% and 6.7% for elections held during 2011 and 2023. The statistics for counted-as-recorded verifiability are bleak, as no one participated. For future elections more effort could be undertaken to involve the wider election security community.

6 Conclusion

This concludes the presentation of the organizational and technical details of the 2023 social elections in Germany. The implementation of the online elections pilot project as part of the 2023 social elections was considered a success, despite the shortcomings outlined in Section 5 and the trust assumptions that were made by the working groups. The Federal Ministry of Labor and Social Affairs is currently working on the introduction of online elections for works council elections. The Federal Election Commissioner for social election has proposed that the legislator create the legal conditions for all social security institutions to be given the opportunity to offer online voting as a rule in future social security elections. Initial steps have been taken to analyze and revise existing guidelines including guideline TR-03162 [Bu23], which was custom-tailored to the social elections 2023. We summarize the experience with the social elections 2023 in a few recommendations.

1. Review the technical guideline [Bu23] with respect to the findings of [Be21].
2. Document and publish the design of the online voting system.
3. Review the design of the voting protocol. This way the issue with the range proofs could have been detected earlier.
4. Review the implementation of the online voting system and compare it against its design. This way the determinism problem could have been identified earlier.
5. Review the cryptographic foundations of the recorded-as-intended mechanism and ensure that it does not break the CPA-security of ElGamal.
6. Review the software for concurrency issues and identify remedial actions that could be taken. Formal verification methods can help to ensure that duplicate votes are never recorded in the ballot box.
7. Raise public awareness for online participation in the social elections, as well as for participation in the counted-as-recorded verifiability event.

Acknowledgments: I would like to thank Wolfgang Becker, Federal Ministry for Labour and Social Affairs, and Sven Heiberg, Smartmatic-Cybernetica, for their insights and discussions. I would also thank the anonymous reviewers for their invaluable comments and suggestions.

[6] See https://www.valimised.ee/en/archive/statistics-about-internet-voting-estonia

Bibliography

[Be21] Beckert, Bernhard; Budurushi, Jurlind; Grunwald, Armin; Krimmer, Robert; Kulyk, Oksana; Küsters, Ralf; Mayer, Andreas; Müller-Quade, Jörn; Neumann, Stephan; Volkamer, Melanie: Aktuelle Entwicklungen im Kontext von Online-Wahlen und digitalen Abstimmungen. Technical report, Karlsruhe Institute of Technology, 2021. 46.23.01; LK 01.

[Bu17a] Bundesamt für Sicherheit in der Informationstechnik: BSI-Standard 200-1 Managementsysteme für Informationssicherheit (ISMS). online, 2017.

[Bu17b] Bundesamt für Sicherheit in der Informationstechnik: BSI-Standard 200-3, Risikomanagement. online, 2017.

[Bu20] Bundesministerium für Gesundheits: Online-Wahl Verordnung. online, 01. Oktober 2020.

[Bu21] Bundesamt für Sicherheit in der Informationstechnik: DER.2: Security Incident Management. online, February 2021.

[Bu23] Bundesamt für Sicherheit in der Informationstechnik: IT-sicherheits-technische Anforderungen zur Durchführung einer Online-Wahl im Rahmen des Modellprojektes nach § 194a Fünftes Buch Sozialgesetzbuch (Online-Wahl). Technische Richtlinie TR-03162, Version 1.3, 3. Februar 2023.

[BV09] BVerfG: Urteil des Zweiten Senats - 2 BvC 3/07 -, Rn. 1-163,, 03. March 2009.

[CDS94] Cramer, Ronald; Damgård, Ivan; Schoenmakers, Berry: Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In: Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology. CRYPTO '94, Springer-Verlag, Berlin, Heidelberg, p. 174–187, 1994.

[Ch24] Christ, Miranda; Baldimtsi, Foteini; Chalkias, Konstantinos Kryptos; Maram, Deepak; Roy, Arnab; Wang, Joy: SoK: Zero-Knowledge Range Proofs. Cryptology ePrint Archive, Paper 2024/430, 2024.

[ESBHL23] Ehrenberg-Silies, Simone; Busch-Heizmann, Anne; Lüddecke, Jost: E-Voting - alternative Wahlformen und ihre Absicherung. Technical report, Büro für Technikfolgen-Abschätzung beim Deutschen Bundestag (TAB), 2023. 46.24.02; LK 01.

[So88] Sozialgesetzbuch Fünftes Buch: Gesetzliche Krankenversicherung - (Artikel 1 des Gesetzes v. 20. Dezember 1988, BGBl. I S. 2477), 1988.

SoK: Mechanisms Used in Practice for Verifiable Internet Voting

Florian Moser [1], Michael Kirsten [2], and Felix Dörre [2]

Abstract: Increasing demands for internet voting instigated the deployment of a multitude of systems used in practice. Within this work, we are interested in which security mechanisms are currently used by vendors to implement verifiable and secret elections.

We perform a systematic market study and review academic literature, where out of 82 candidate systems, we find 29 internet voting systems that are both in active use and claim to employ some form of verifiability. Thereof, we characterize and systematize the 18 systems that provide sufficient information to extract their security mechanisms relevant for state-of-the-art verifiability and secrecy. Overall, we find that only eight systems are well-documented, of which only a few employ state-of-the-art mechanisms in all categories that we consider.

Keywords: Internet Voting, Systematic Study, Verifiable Elections, Security Mechanisms

1 Introduction

The internet plays an increasingly active role in many people's lives, and more and more scenarios, e.g., working from home, depend crucially on performing tasks over the internet. Hence, expectations and demand are growing that more aspects of our lives are accessible in a remote setting. Remote voting over the internet, also called *internet voting*, forms a natural part in this demand, which is also reflected in a growing interest in electronic voting (*e-voting*) by research [Ai24].

Internet voting comprises inherently uncontrolled environments with many stakeholders and potential adversaries, and comes with many requirements, e.g., robustness, integrity, verifiability, secrecy, usability, etc. [HGB23]. Depending on the specific trust assumptions, many of the requirements are inherently or partially in conflict [Kr23]. Moreover, simply adding new technologies may create new potential for attacks and can generally make voting systems more vulnerable [Pa21]. While various mechanisms for verifiable internet elections are state-of-the-art in academia [CT16; KHC22], bringing verifiable electronic voting into practice is still challenging [Te21]. Our primary interest in this work is hence to find out which mechanisms current systems actually use, to shed some light on gaps between academia and practice.

[1] INRIA Nancy, France, florian.moser@inria.fr, https://orcid.org/0000-0003-2268-2367
[2] KASTEL Security Research Labs, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, kirsten@kit.edu, https://orcid.org/0000-0001-9816-1504;
felix.doerre@kit.edu, https://orcid.org/0009-0009-7244-7753

doi:10.18420/e-vote-id2024_10

Contributions. Our contribution consists in a systematization of knowledge of the current implementation of verifiable internet voting systems used in practice, by classifying their mechanisms based on state-of-the-art criteria from academia, specifically the following:

1. a market study and an academic literature study identifying 82 different internet voting systems used in practice,
2. a systematization of security mechanisms suitable to characterize state-of-the-art internet voting systems,
3. a mapping of our systematization to the 18 voting systems sufficiently documented for our assessment,
4. a brief discussion about our findings and particular observations.

2 Related Work

Comparing practical voting systems based on requirements from academia is challenging. Most related works aim for satisfied security properties, but the challenges therein are multidimensional and not mutually aligned. First, there is no generally agreed-upon list of required properties with fixed definitions. Second, the trust assumptions within each voting system are different, each striving for different criteria and a different election context. In the following, we reflect on the most recent surveys targeting internet voting systems.

Li et al. [LKZ14] systematize 2014's state of the art for 14 electronic voting systems, 9 of them suitable for internet voting, both for cryptographic mechanisms and for satisfied properties. However, they do not report on how they choose the systems, and many of them are not actively used anymore. Moreover, as Li et al. do not specifically target internet voting systems, their classification does not consider the different phases therein. Kho et al. [KHC22] provide an extensive and fine-grained systematization of various cryptographic mechanisms from academia. They identify and compare different electronic voting approaches from literature, provide general structures for mixnet-based, homomorphic, blind signature-based, blockchain-based, and post-quantum electronic voting, and discuss the approaches based on various academic publications. However, they do not report on the selection of the discussed approaches and systems, as well as their relation to systems used in practice.

Most recently, Finogina et al. [FCC24] select twelve internet voting systems that they deem well-known, extract the stakeholders, and compare their trust assumptions for the properties cast-as-intended verifiability, recorded-as-cast verifiability, universal verifiability, privacy, and receipt-freeness. They also compare the systems on their communication channels, scalability, voting channel flexibility, support for preferential voting, as well as support for digital governmental identity. While proposing a unified convention and systematization, their selection of systems is limited, and it is unclear how their trust assumptions and property interpretations can be generalized for a wider comparison of systems. In the attempt to handle the actual situation on the market, our comparison is more general.

3 Method

In this section, we describe our method in detail. For this work, we aim to study verifiable internet voting systems in active use. In a first step, we discover candidate systems, trading recall for precision, and in a second step, we reduce the list to systems that meet our selection criteria, as described in Sect. 3.1. Moreover, we construct initial reviews of the systems, which we use as a basis to perform a systematization of mechanisms, as described in Sect. 3.2. Finally, using only publicly available material, we review each system in detail and assess the implemented mechanisms, as described in Sect. 3.3.

3.1 Discovery of Systems

To discover practical systems in active use, we emulate an organization who wishes to provide verifiable internet voting to their electorate. We assume that the organization has no specific election competencies, however, we assume some intrinsic motivation to do well by spending dedicated effort to learn about internet voting for choosing a system that meets appropriate security criteria. For this matter, we assume a high-level understanding of internet voting, i.e., knowledge about some relevant security requirements, e.g., “end-to-end verifiability” and related mechanisms, e.g., “mixnets”. However, we do not assume any corresponding technical expertise or prior knowledge about existing systems. Assuming this coarse-grained understanding of secure internet voting, we perform a market study, with the (hypothetical) aim of choosing a secure state-of-the-art system.

To replicate this setting, we let a business consultancy perform the market study, with experience in market studies, but without prior knowledge in internet voting, thereby ensuring an unbiased perspective. The consultancy received a brief (less than one hour) introduction by an internet voting expert on the general process of holding elections, guarantees generally desired in such systems, and an overview of some of the mechanisms usually employed to reach these guarantees. We consider this sufficient to give a coarse-grained understanding, but not more than what we would expect from motivated but non-technical election organizers.

To check the output of the market study, and minimize chances of ignoring relevant systems, we complement the market study by gathering practical systems referenced in academic literature (e.g., previous comparisons or system proposals). Further, we cross-check the resulting list of systems with two other market analyzes and one list of systems maintained by a research team. Throughout this process, we prioritize recall over prevision, which however leads to discovering many products that are not relevant to our work. Hence, in a final step, we filter the *longlist* of discovered systems for a *shortlist* of systems that meet our criteria of being actively used for verifiable internet voting. See Tab. 1 for an overview on how many systems were added in which step.

Source	Longlist	Shortlist
Our Market Study	60	17
Our Literature Study	+12	+5
Analysis on French Market	+6	+5
Analysis on German Market	+2	+1
Researchers' List in France	+2	+1
Total	**82**	**29**

Tab. 1: We found a *longlist* of 82 systems broadly related to verifiable internet voting in practice. Filtering for our criteria, i.e., internet voting, end-to-end verifiability and privacy, and in recent use, leaves a *shortlist* of 29 systems for further analysis.

Market study. To discover products on the market, the business consultancy searched for products that are broadly related to online voting, thereby aiming for breadth over depth. They conducted the search primarily in January 2024 by using Google, Bing, and Yandex, as well as relevant Wikipedia articles and LinkedIn, with native language proficiency in English and German. In their search, the consultancy used keywords related to the general topic (e.g., "election", "voting"), mechanisms (e.g., "zero-knowledge proofs", "mixnet"), and security properties (e.g., "end-to-end verifiability", "privacy"). For cross-checking intermediate results, they consulted ChatGPT 4.0 with a dataset from around January 2022. In total, the consultancy identified between 250 and 350 products that fit one or more of the keywords. Within the discovery phase, the consultancy employed superficial judgement of the providers' marketing and immediately discarded products not directly related to *verifiable* internet voting. We included the remaining 60 products on our *longlist*.

Literature study. To discover products referenced in academic literature, we ran a manual search, primarily in October 2023, of online-accessible databases and webpages for all proceedings (including relevant satellite workshops) since 2016 of the most relevant academic venues.[3] Moreover, we included the preprint server *IACR Cryptology ePrint Archive* on which many academics publish technical reports or preprint versions of their submissions in cryptography- or security-related domains. In our search, we used the general keywords "voting" and "election", and then scanned the titles for any of the keywords "voting system", "voting protocol", "voting scheme", "internet voting", "remote voting", "electronic voting", "e-voting", "electronic election", "verifiable", "verifiability", "review", "study", or "survey". We then did a semantic scan by manually reading all titles and abstracts for whether the publication mentions any verifiable internet voting system, which comprises papers presenting specific systems of that nature, as well as surveys, evaluations, analyzes or employments. Moreover, we extended our search by scanning the papers' references from their introduction, related work section, and evaluation sections, for the same criteria.

[3] These venues are the *International Joint Conference on Electronic Voting* (E-Vote-ID), the *ACM Conference on Computer and Communications Security* (CCS), the *IEEE Computer Security Foundations Symposium* (CSF), the *European Symposium on Research in Computer Security* (ESORICS), the *International Conference on Financial Cryptography and Data Security* (FC), and the *IEEE Symposium on Security and Privacy* (SP).

Our semantic scan aimed to identify systems with recent (i.e., within the last two years) usage in real-world internet elections, i.e., we excluded any kind of in-place systems. Our criterion of real-world comprises any election with decision power on the level of national governments, states or cantons, districts, municipalities, companies, associations or organizations, institutions (e.g., universities), or political parties. We disregarded systems used purely for demonstration purposes or scientific papers, as well as clearly abandoned systems. Collecting all systems proposed in academia would include many more systems, but is out of scope for our purpose. In total, we found 22 systems and, after deduplication and consultation of the systems' webpages and manuals, we added 12 systems to our longlist.

Cross-check with other system collections. To ensure that our list of systems does not contain significant gaps, we did a cross-check with the following two market studies and one list of systems:[4]

- market analysis focused on French market leaders in internet voting (9 systems, 2023)
- market analysis focused on the German market of internet voting (63 systems, 2022)
- list of systems maintained by a French research team focused on end-to-end verifiable systems (12 systems, continuously updated)

The added systems from the first analysis only provide information in French, and were hence hard to discover in our market study performed in English and German. From the second analysis, we directly discarded systems which provide only simple polls or conference voting, as this is in the scope of our work. In total, after deduplication, we added 6 systems from the analysis on the French market, 2 systems from the analysis on the German market, and 1 system from the list by French researchers (see Tab. 1).

Result. The discovery phase resulted in a longlist with 82 systems (see Tab. 1), for which we collected general facts, i.e., vendor, location and webpage. We further constructed a *shortlist*, for which we reduced the longlist's systems in order to fulfill the following criteria:

- Real internet voting systems. The system allows casting votes over the internet. This excludes, e.g., election management software, or systems which require a paper trail where the voter sends a receipt of their vote to the election authorities after voting.
- Aim for end-to-end verifiability and some form of privacy. We welcome different jargon, as long as the overall goal promises some form of end-to-end verifiability and privacy. Further, we admit claimed compliance to some electronic-voting-related certification (e.g., to the French CNIL [Co19a; Co19b] or the Common Criteria Protection Profiles published by the German BSI [IT23; VV08]), while we deem more generic certifications such as ISO 27001 as insufficient.

[4] Made internally available as part of a study on E2E-verifiability for the Federal Office for Information Security.

- Active or recent use. We check the proxy criterion that some official communication (e.g., on LinkedIn), code or documentation was produced within the last two years.

Based on these criteria, we found that 9 systems do not provide internet voting, 15 systems already vanished (e.g., dead startups), and 29 systems have no claims related to verifiability. This resulted in a shortlist that leaves 29 systems which fulfill all our criteria, and which we aim to analyze in detail. We give an overview of which source contributed how many systems to the longlist and the shortlist in Tab. 1. The full longlist with all systems and their respective criteria evaluation is publicly available.[5]

Discussion. We observe that the market study alone already covers a large share of the systems on both lists (see Tab. 1), and we therefore expect our findings to represent the market well, by having found the most relevant systems used in practice. Further, more than 80 percent of the systems in the market study's longlist differ from the systems in the literature study, indicating that many systems in practice do not publish scientifically and are not considered when comparing systems in scientific publications.

For transparency, we report on indicators of gaps on our lists in the following. In our hypothetical scenario, we do not consider hard-to-find systems as explicit gaps, as the organization emulated in our market study would also fail to learn of the system. Yet, we expect gaps for systems that are neither advertised in English nor in German, as we observed in our cross-check for the French market. Further, our market study may exclude companies not explicitly advertising verifiability, but who only use it under the hood.

3.2 Systematization of Mechanisms

The chosen systematization aims to consider only mechanisms which are fundamentally different. Hence, we ignore implementation details, and collapse slight variations of a mechanism (small tweaks to achieve, e.g., improved verifiability or secrecy) into a single mechanism. Thereby, we aim to avoid noise in our evaluation, but still distinguish clearly separate approaches. To design this systematization, we first perform two independent coarse-grained reviews of some of the chosen systems. Each review results in a systematization proposal which captures the respective subset of systems from the shortlist. Second, we unify the two systematization proposals into our final systematization, in order to capture the full diversity of the systems.

[5] The list is available at `https://inria.hal.science/hal-04686386v1/file/longlist.csv`.

3.3 Extraction of Mechanisms from Systems

All the systems on our shortlist (see Sect. 3.1) claim some level of verifiability, and as verifiability aims to make a system observable to outsiders, this directly requires that the system's inner workings are communicated clearly and openly. We would therefore expect all systems to publish at least a detailed system specification, possibly also a reference implementation, security proofs, and expert reviews of their system.

Our goal consists in assessing which mechanisms are used in practice, not in assessing the systems themselves. We can hence relax the thoroughness of our reviews and require less documentation than would be necessary for full system reviews. Our classification targets the most positive and secure or secret interpretation of the described mechanisms, where we favorably fill gaps with educated guesses, and generally trust the claims by the vendors. Practically, often a high-level overview of the system (e.g., a whitepaper, or explicit claims on the vendor's webpage) is sufficient, if that allows to clearly identify which mechanisms are in use. However, even in this relaxed setting, many systems do not publicly provide sufficient information for extracting the mechanisms in use. The following list contains all systems on our shortlist for which we could not find sufficient information and which we therefore excluded from further analysis:

- *Electis* (France) [El24b]. The code is open source [El24a] and seems to integrate with ElectionGuard [Be24b] and the Tezos blockchain, but it is hard to extract the specific security mechanisms of the currently implemented solution. The product claims to follow the CNIL recommendations for electronic voting [Co19b].
- *Eligo* (Italy) [El24d]. The published documentation does not disclose security details apart from support for two-factor authentication.
- *Genolive* (Germany) [Co24]. While the product claims to be developed in compliance with the BSI Common Criteria Protection Profile for internet voting [VV08], the employed security mechanisms are not described.
- *Kercia Solutions* (France) [Ke24], *LegaVote* (France) [Le24], *Neovote* (France) [Ne24a], *WebVote* by Gedivote (France) [Ge24], and *WeeChooz* (France) [We24]. We could not find documentation about the security mechanisms by these systems. All products claim to follow the CNIL recommendations for electronic voting [Co19b].
- *Neuvote* (Canada) [Ne24b]. The webpage does not disclose any security details besides support for end-to-end encryption and two-factor authentication.
- *Nvotes* (Spain) [nV24]. The webpage does not contain any concrete details about their system. However, the technical leadership seems to have moved to Sequent, which we analyze as part of this work (see below).
- *Onlz* (Belgium) [On24]. The webpage does not contain details about the system and only mentions to employ client-side encryption, blockchain and ring signatures.

For the following systems, we are able to extract the mechanisms from public specifications:

- *Belenios* (France) [Gl24]. Besides the specification, there is a high-level academic publication which discusses some of the design decisions [CGG19]. Further, the code is open source [In24], and there are further supporting documents, including academically reviewed formal proofs [Be24a].
- *Electa* by AssemblyVoting (Denmark) [As23]. Besides the specification, some of the code is public [As24].
- *Helios* [Ad08]. Besides the initial academic publication and the specification, which was available until April 2024 [Ad12], the code is open source [He24]. Recent development in the repository seems to focus on dependency updates, minor usability features, and authentication mechanisms.
- *IVXV* by Cybernetica (Estonia) [Va22]. Besides the specification, the code is open source [Va24].
- *Polyas Core-3* (Germany) [Tr23]. Besides the system specification, there is an academic paper and a specification for their cast-and-audit approach [MT23; Tr24]. Further, there are some non-technical descriptions on the webpage [PO24].
- *Sequent* (USA) [Se24a]. There is a webpage with a system overview, documentation [Se24c] including usage demos, explanations, the high-level cryptographic protocol [Ru21], and system architecture [Ro22], as well as open-source code [Se24b].
- *Swiss Post Voting System* (Switzerland) [Sw24d; Sw24e]. Besides the specification, extensive documentation of the system, formal proofs, as well as the code is public [Sw24c]. The system is provided for political elections in the Swiss cantons [Sw24b], and is hence compliant to the corresponding Swiss law [BV13]. Further, numerous system reviews by independent experts are publicly accessible [Sw24a].
- *uniWAHL* by ElectricPaper (Germany) [El24c]. The webpage focuses on explanations and demos and is mostly aimed at election organizers. Under the hood, they use the security mechanisms provided by Sequent.[6]

For these systems, we use informal partial specifications, whitepapers, code, or webpages:

- *BigPulse Voting* (UK) [Bi24b]. The verification is sketched on the webpage [Bi24a]. We did not find details about authentication or encryption of the votes.
- *DecentraVote* (Germany) [De24a]. Besides the whitepaper [Fa20], the code is open source [De24b].
- *EVoters* by EVoting (Chile) [EV24a]. The used encryption, as well as the tally mechanism, are described on their webpage [EV24b]. We could not find details about

[6] https://sequentech.io/case-study/bringing-sequents-end-to-end-verifiable-voting-solution-to-the-german-market/

how votes are verified by the voter or how the votes are stored, and found only very high-level information on the employed authentication.

- *Followmyvote* (USA) [Fo24a]. The webpage provides informal descriptions and some system code is public [Fo24b].[7]
- *Invote* by Scytl (Spain) [Sc24]. The specification is not public, but the mechanisms are sketched in an academic survey paper [FCC24].
- *Simply Voting* (Canada) [Si24b]. Some mechanisms are documented [Si24a].
- *V8te* (France) [V824]. The privacy policy describes some mechanisms [Cé23].
- *Voatz* (USA) [Vo24a]. The webpage features a video which describes some of the security mechanisms and links to a high-level whitepaper [Mo20]. It remains unclear how ballots are encrypted and authenticated.
- *Voxaly* (France) [Vo24b]. Other than compliance to the recommendations of CNIL, the webpage does not provide a closer description of the solution. However, a partial specification [Ch23], yet excluding authentication and record details, of the system used for the legislative elections in France is public and described as an adaptation of the Belenios system, which we also analyze as part of this work (see above).
- *zkVoting* (Korea) [Pa24]. An academic whitepaper describes cryptographic details.

4 Analysis

We organize the mechanisms into *Cast*, *Authentication*, *Cast Verification*, *User Record*, *System Record* and *Tally*. Cast considers all mechanisms which contribute towards forming the ballot. Authentication authenticates this ballot (or the corresponding voter). Cast verification helps the voter be assured that the ballot was formed correctly. Record tracks mechanisms which store data on the voter's device or system-side. Tally finally encompasses the mechanisms to cleanse the list of ballots (e.g., of revotes), and anonymize and decrypt.

[7] The provided code includes smart contracts and some UI code, but notably no code employed by the registrar.

4.1 Cast

When the voter casts their vote, they form their ballot. We track the employed encryption (if any) and how the corresponding (decryption) keys are handled.

Encryption. We distinguish whether the ballot is encrypted using *symmetric encryption* or *asymmetric encryption*. In the internet voting setting, asymmetric encryption is desirable, as it is a precondition to employ any of the privacy-preserving tally methods (see Sect. 4.6). Voting systems which assume anonymous channels may omit encrypting the ballots.

Keys. We place special attention to how encryption keys are handled. Storing the decryption key in a *distributed storage* requires multiple trustees to reconstruct the decryption key. Further, to prevent that a single trustee learns the decryption key when it is generated, we track support for *distributed generation*.

	Belenios	Electa	uniWAHL	Helios	IVXV	Polyas	Sequent	Swiss Post	BigPulse	DecentraVote	followmyvote	Invote	SimplyVoting	V8te	Voatz	EVoters	Voxaly	zkVoting
Encryption																		
symmetric										●								●
asymmetric	●	●	●	●	●	●	●	●				●		●		●	●	
Keys																		
distributed storage	●	●	●	●	●	●	●	●				●		●		●	●	
distributed generation	●	●	●	●		●	●	●										

Tab. 2: Comparison of voting systems concerning their mechanism used for encryption and keys, well-documented systems to the left of the dashed line. DecentraVote uses a hash to hide the plain vote, taken from the plain vote concatenated with secrets, with the secrets published in the tally phase to count the votes. zkVoting uses a hybrid encryption scheme, with the ballot itself encrypted under a symmetric key, and that symmetric key under the public election key. For BigPulse, followvoting, Simply Voting and Voatz, we did not find claims about the used encryption.

4.2 Authentication

Most systems document that the voter needs to pass one or multiple logins, similar to how the voter would interact with other online systems. We are however interested in whether and how the system employs credentials, which we understand as cryptographic keys with a private and a public part, necessary for state-of-the-art cryptographic ballot authentication.

Credential generation. Credentials may be generated *centralized*, where a single authority learns all credentials, or *distributed* by multiple authorities. Some systems may generate the credentials *on the voter's device*, and others rely on *election-independent* credentials established outside the specific election context, e.g, for other uses such as electronic IDs.

Credential interface. For credentials that were not already generated on the voter's device, we distinguish the interface for the credentials' usage. Credentials may be explicitly delivered to the voter on a dedicated channel, where the credential is then *directly entered* by the voter into the voter's device, possibly looking similar to entering a long password. Another possibility is that the credential is stored on *trusted hardware*, such as an electronic ID card, to ensure that the credential can only be used by the holder of the trusted hardware.

Credential properties. Most systems use credentials that are *cryptographically bound* to the ballot, to harden against a dishonest voter attempting to (re)use the authentication for a different ballot. Systems may further use *distributed authentication*, where they rely on multiple authorities who authenticate the public part of the credential.

	Belenios	Electa	uniWAHL	Helios	IVXV	Polyas	Sequent	Swiss Post	BigPulse	DecentraVote	followmyvote	Invote	SimplyVoting	V8te	Voatz	EVoters	Voxaly	zkVoting
Credential Generation																		
centralized	●					●		●				●						
distributed		●																
on the voter's device		●								●	●							●
election-independent					●													
Credential Interface																		
directly entered	●	●				●		●				●						
trusted hardware					●													
Credential Properties																		
cryptographically bound	●	●			●	●		●		●		●						●
distributed authentication		●																

Tab. 3: Comparison of voting systems on how credentials are generated and used, well-documented systems to the left of the dashed line. Electa supports credential-based authentication, where credentials are distributed by credential authorities and entered by the voter, and identity-based authentication, where the voter's device generates the credential, which the voter authenticates using identity providers.

4.3 Cast Verification

When casting the vote, systems often allow verifying whether the ballot has been formed correctly. These verification mechanisms may deliver either probabilistic or definite guarantees. In the former, the voter gets probabilistic (game-based) guarantees over whether their ballot is correct, which get stronger when the voter repeats the verification multiple times. In the latter, if the verification succeeds, the ballot represents only with negligible probability something different (assuming that the system's trust assumptions are fulfilled).

Verification. We observe the probabilistic *audit-or-cast* approach in use, where after the ballot is formed, the voter is given the choice to either audit or cast the ballot. If the voter chooses the former, they may verify whether the ballot was formed correctly, but must restart the cast procedure until they finally choose to cast the ballot. If the voter chooses to cast the ballot, they need to check that the ballot reached the voting system exactly as formed.

Some systems allow auditing the cast vote, which we name *cast-then-audit* approach. Such systems may take precautions attempting to avoid that the voter gets a trivial receipt, e.g., limiting the time during which the voter can audit, keeping the bulletin board private, and employing zero-knowledge proofs. We name the mechanism *clear-text receipt* if the voter gets some confirmation which includes the plaintext vote. If the voter receives opaque codes that correspond to their vote in the voting phase, we name it *return codes*. We name it *audit codes* when the voter may look up the content of their anonymized vote using a code.

	Belenios	Electa	uniWAHL	Helios	IVXV	Polyas	Sequent	Swiss Post	BigPulse	DecentraVote	followmyvote	Invote	SimplyVoting	V8te	Voatz	EVoters	Voxaly	zkVoting
audit-or-cast		●	●	●			●											
cast-then-audit					●	●						●						●
return codes								●										
clear-text receipts													●		●			
audit codes						●			●				●		●			

Tab. 4: Comparison of voting systems on their cast verification mechanism, well-documented systems to the left of the dashed line. Some systems do not employ cast verification, notably Belenios, and the Voxaly system which adapted Belenios. Audit codes are sometimes combined with other mechanisms.

4.4 User Record

Once the authenticated vote is cast, the system may provide an artifact to the user that the ballot has really been stored. Further, this artifact may be authenticated to avoid disputes over the artifacts' validity, and chained, to harden against later changes to the system state.

User artifact. The voter may receive a *storage reference*, which upon querying the system reveals the stored ballot (e.g., a transaction ID in a blockchain). Otherwise, the voter may receive the *ballot hash*, or even the *plain vote* itself as artifact. Some systems also provide a *return code*, which asserts that some, possibly previously verified, ballot has been stored.

Artifact properties. The artifact that the voter receives may be authenticated, i.e., *signed* using a digital signature, which allows proving that the artifact really originated from the system. This makes the system accountable for the artifacts that it issues. Further, the artifact may cryptographically reference the ballots already cast by other voters, which we then name *chained*. This forces the system to commit to the ballots it already stores at the time of issuing the artifact, which hardens against later tampering with that state.

	Belenios	Electa	uniWAHL	Helios	IVXV	Polyas	Sequent	Swiss Post	BigPulse	DecentraVote	followmyvote	Invote	SimplyVoting	V8te	Voatz	EVoters	Voxaly	zkVoting
User Artifact																		
storage ref		●							●	●	●	●		●	●			●
ballot hash	●		●	●	●	●	●										●	
plain vote													●					
return code								●										
Artifact Properties																		
signed		●			●	●			●								●	
chained										●	●				●			●

Tab. 5: Comparison of voting systems concerning their user record mechanism, well-documented systems to the left of the dashed line.

4.5 System Record

Once the authenticated vote is cast, the system needs to record the vote until the tally. The system may provide access to the vote's current state. This state may be distributed over multiple authorities, with some consistency procedure in place.

System access. Some systems may provide voters with access to the *individual entry* of the individual voter, or even to *all entries* of all voters. More permissive systems may improve (public) verifiability, possibly at the cost of (everlasting) privacy.

System consistency. When a new ballot is submitted, *centralized* systems trivially achieve consistency, i.e., they register the ballot as submitted (or not). When the state is instead distributed over multiple authorities, some systems still achieve *immediate* consistency; i.e., the ballot is added immediately to the list of to-be-tallied ballots' state of the system. Other distributed systems may feature *delayed* consistency, with an explicit consistency agreement procedure at the latest before the ballots are tallied.

	Belenios	Electa	uniWAHL	Helios	IVXV	Polyas	Sequent	Swiss Post	BigPulse	DecentraVote	followmyvote	Invote	SimplyVoting	V8te	Voatz	EVoters	Voxaly	zkVoting
Access																		
indiv. entries	●	●	●		●		●		●	●	●	◐	●					●
all entries	●	●	●				●		●	●	●	◐	●	●				●
Consistency																		
centralized	●	●	●	●		●	●											
immediate										●	●				●			●
delayed					●			●										

Tab. 6: Comparison of voting systems on their system record mechanism, well-documented systems to the left of the dashed line. Invote makes the receipts publicly accessible, but not the ballots themselves.

4.6 Tally

After the voting phase closes, the system will calculate the result. First, the system may perform a cleansing to, e.g., filter out revotes. Then, ballots are anonymized, i.e., the link to the submitting user is removed, and decrypted.

Cleansing. Some systems support *revotes*, where voters may vote multiple times and the latest ballot overrides the previous ones. Further, *revoked ballots* (e.g., in case a voter later-on decides to cast in person) and *fake ballots* (i.e., ballots cast with invalid credentials) need to be prevented from entering the tally. These mechanisms can be used to increase voter privacy in case the system hides which ballots are removed in the cleansing process.

Anonymization. To anonymize, *homomorphic aggregation* allows aggregating all ciphertexts into a single ciphertext which sums up the chosen candidates from its contributing ciphertexts. Then, only this resulting ciphertext is decrypted. However, the implementation of homomorphic aggregation is difficult and computation-intensive for arbitrary counting functions, which is one of the reasons a system may use a *verifiable shuffle*. The shuffle preserves each ballot in their own ciphertext, while removing the link to the submitted ciphertext using verifiable re-encryption and secret shuffling.

Decryption. Decryption may be *verifiable*, hence the decrypting trustee needs to prove that the provided plaintext corresponds to the ciphertext. Additionally, decryption may be *distributed*, where the trustees never share their (partial) decryption keys among the other trustees, thereby allowing the trustee to remain in control over which ciphertext is decrypted.

	Belenios	Electa	uniWAHL	Helios	IVXV	Polyas	Sequent	Swiss Post	BigPulse	DecentraVote	followmyvote	Invote	SimplyVoting	V8te	Voatz	EVoters	Voxaly	zkVoting
Cleansing																		
revotes	●	●	●	●	●		●		●						●			
revoked ballots					●	●												
fake ballots																		●
Anonymization																		
homomorphic aggr.	●			●								●		●		●	●	◐
verifiable shuffle	●	●	●		●	●	●	●				●						
Decryption																		
verifiable	●	●	●	●	●	●	●	●		●		●					●	●
distributed	●	●	●	●		●	●	●										

Tab. 7: Comparison of voting systems on their tally mechanism, well-documented systems to the left of the dashed line. In zkVoting, a single authority decrypts the (hybridly-encrypted) ballots, sums up the plaintexts, and then publishes the sum with a proof that it corresponds to the authenticated ballots.

5 Conclusion

We found as many as 82 internet voting systems that are broadly related to *verifiable* internet voting. The initial barrier of providing an internet voting system seems to be low, and small and local products are prevalent. Almost all systems use security as a sales argument, largely independently of the actually taken (and publicly documented) security measures. Most systems do not even claim to implement verifiability in their products, and out of those that do, most systems document their solution poorly. As an example, from the systems claiming to implement the CNIL e-voting recommendations [Co19b], six out of nine systems did not publish concrete documentation, and two published a very partial description. Already for other secure systems, such a lack of documentation would be surprising, considering the security principle of *open design* [SJT08]. In the context of internet voting, however, transparency is especially crucial, and voters do consider it important [Ag23].

When evaluating the mechanisms, most of the eight sufficiently documented systems do not implement state-of-the-art security mechanisms in all relevant categories. While asymmetric encryption is typically used to encrypt the vote, only seven systems combine it with distributed generation and distributed storage of the corresponding private key. Similarly, while homomorphic aggregation or verifiable shuffling are used in practice, only the same seven systems then perform the decryption in a verifiable and distributed fashion. All other systems omit at least the distributed generation and the distributed decryption, making the component which learns the full decryption key a potential single point of failure. While the state-of-the-art for casting and tallying seems mostly established, and most systems use the same mechanisms, many different approaches are in use for authentication, cast verification and recording, generally with no clearly emerging favorite. Notably for authentication, some systems support multiple mechanisms and let the election organizer pick the approach that suits their election context best.

Besides the large gaps between practice and research reported above, we want to briefly discuss the systems which employ some of the relevant state-of-the-art security mechanisms. The systems Belenios and Helios, both largely developed directly within academia, provide very transparent and extensive documentation: While Helios shows no recent development anymore and is a considerably smaller system with not as much documentation, both systems provide source code and a range of academic publications about the systems and proposed variants, and Belenios even provides formal proofs [Ad08; Be24a]. Yet, also some systems developed in industry publish extensive documentations and source code. Most notably, the Swiss Post System provides detailed specification, source code, formal proofs, as well as extensive third-party expert reviews [Sw24a; Sw24c]. These extensive transparency measures are required by law for the Swiss national political elections [BV13], for which the Swiss Post System is built. Both, the Sequent (and hence also uniWAHL) and the IVXV system also publicly provide many details on their mechanisms and their source code [Se24a; Va22; Va24]. Moreover, the Electa system also provides an easily-accessible and high-quality specification describing their used cryptographic mechanisms [As23]. Also, for the Polyas system, system specifications and an academic publication are available [Tr23].

Finally, we have seen many systems report using the open-source libraries and development kits Verificatum [Wi24] and ElectionGuard [Be24b] under the hood, profiting from their well-vetted implementation of core security mechanisms, including extensive specifications.

Limitations. Despite the large number of inspected systems and our general systematization thereof, our study has some limitations to be considered. First, as the quality of the systems' documentations varied widely, our systematization only considers the general mechanisms claimed to be used by the system, and not necessarily the specific implementations or variations of the mechanisms. Second, while employing critical judgement of the advertised systems and mechanisms, our study is bound to believe the officially available statements for a fair comparison between those systems which provide implementation details and other systems which only provide high-level and informal descriptions. Our characterization may hence slightly favor an ambitious high-level description over disadvantageous implementation details.

Future work. Our study and systematization provide a basis for further, more specialized endeavors for a systematization and assessment of practical systems. It would be interesting to, e.g., use the recommendations for source code examinations by Haines and Rønne [HR21] within each of the categories for mechanisms from systems in practice, where the systems provide public access to their code. Moreover, our systematization opens up the potential for using more detailed analysis frameworks for specific trust models within each of the categories, instead of challenging comparisons of complete systems. Such category-specific assessments could allow more fine-grained comparisons of adversarial capabilities [NNV17], trust levels [Ne21] and trust assumptions [Kr23] in a meaningful way. Finally, practical systems do not only aim for security and privacy, and it might be sensible to also include further dimensions on which internet voting systems are assessed, instead of focusing solely on the security mechanisms [Wi18].

Acknowledgements

Parts of this work were based on a study on end-to-end-verifiability by the Federal Office for Information Security (BSI). We thank Jeremy L. Meyer from Performance Partners GmbH for support when executing the market analysis. This work was supported by funding from the topic Engineering Secure Systems of the Helmholtz Association (HGF) and by KASTEL Security Research Labs. Further, this work benefited from funding managed by the French National Research Agency under the France 2030 program with the reference ANR-22-PECY-0006. It was also partly supported by the ANR Chair IA ASAP (ANR-20-CHIA-0024) with support from the region Grand Est, France.

References

[Ad08] Adida, B.: Helios: Web-based Open-Audit Voting. In: 17th USENIX Security Symposium, San Jose, CA, USA July 28–Aug. 1, 2008. USENIX Association, pp. 335–348, 2008, URL: https://www.usenix.org/legacy/events/sec08/tech/full_papers/adida/adida.pdf.

[Ad12] Adida, B.: Helios v4, tech. rep., Helios, 2012, URL: https://web.archive.org/web/20240417102302/https://documentation.heliosvoting.org/verification-specs/helios-v4.

[Ag23] Agbesi, S.; Budurushi, J.; Dalela, A.; Kulyk, O.: Investigating Transparency Dimensions for Internet Voting. In: 8th International Joint Conference on Electronic Voting (E-Vote-ID 2023), Luxembourg City, Luxembourg Oct. 3–6, 2023. Springer, pp. 1–17, 2023, DOI: 10.1007/978-3-031-43756-4_1.

[Ai24] Aidynov, T.; Goranin, N.; Satybaldina, D.; Nurusheva, A.: A Systematic Literature Review of Current Trends in Electronic Voting System Protection Using Modern Cryptography. Applied Sciences 14 (7), 2742, 2024, DOI: 10.3390/app14072742.

[As23] Assembly Voting: Electa: Documentation of the cryptographic protocol (Version 2.0), tech. rep., Assembly Voting, 2023, URL: https://web.archive.org/web/20240714102041/https://downloads.assembly-voting.com/download/marketing/electa_-_documentation_of_the_cryptographic_protocol.pdf.

[As24] Assembly Voting: Assembly Voting SDKs, 2024, URL: https://github.com/aion-dk.

[Be24a] Belenios: Belenios documentation, 2024, URL: https://web.archive.org/web/20240708065438/https://www.belenios.org/documentation.html.

[Be24b] Benaloh, J.; Naehrig, M.; Pereira, O.; Wallach, D.: ElectionGuard: a Cryptographic Toolkit to Enable Verifiable Elections. In: 33rd USENIX Security Symposium (USENIX Security 2024), Philadelphia, PA, USA Aug. 14–16, 2024. USENIX Association, 2024, URL: https://www.usenix.org/conference/usenixsecurity24/presentation/benaloh.

[Bi24a] BigPulse: Election vote count verification protocol, 2024, URL: https://www.bigpulsevoting.com/about/vote-count-verification-protocol/.

[Bi24b] BigPulse: Secure Online Voting System, 2024, URL: https://www.bigpulsevoting.com/.

[BV13] Bozzini, D.; Varone, A.: Federal Chancellery Ordinance on Electronic Voting, ordinance, Swiss Federal Chancellery, 2013, URL: https://cva.unifr.ch/content/federal-chancellery-ordinance-electronic-voting.

[Cé23] Cécile: Privacy Policy - V8TE, tech. rep., V8TE, 2023, URL: https://web.archive.org/web/20240822085749/https://drive.usercontent.google.com/download?id=1SOxRXNdHsTazUjcJt1EOcYw3hA6fy5JG&export=download&authuser=0.

[CGG19] Cortier, V.; Gaudry, P.; Glondu, S.: Belenios: A Simple Private and Verifiable Electronic Voting System. In: Foundations of Security, Protocols, and Equational Reasoning - Essays Dedicated to Catherine A. Meadows. Vol. 11565. LNCS, Springer, pp. 214–238, 2019, DOI: 10.1007/978-3-030-19052-1_14.

[Ch23] Chenon, B.: MEAE - Transparence et vérifiabilité V2 (Version 2.04), tech. rep., Voxaly, 2023, URL: https://web.archive.org/web/20240308073039/https://www.voxaly.com/wp-content/uploads/VOXALY-LEG2023-Transparence-et-Verifiabilite-Specifications-publiques-v2-04.pdf.

[Co19a] Commission nationale de l'informatique et des libertés: Délibération n° 2019-053 du 25 avril 2019 portant adoption d'une recommandation relative à la sécurité des systèmes de vote par correspondance électronique, notamment via Internet, tech. rep., Journal Officiel de la République Française, 2019, URL: https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000038661239.

[Co19b] Commission nationale de l'informatique et des libertés: Sécurité des systèmes de vote par internet: la CNIL actualise sa recommandation de 2010, 2019, URL: https://www.cnil.fr/fr/securite-des-systemes-de-vote-par-internet-la-cnil-actualise-sa-recommandation-de-2010.

[Co24] Conventex: Digitale Wahlen, 2024, URL: https://conventex.com/digitale-wahlen/.

[CT16] Culnane, C.; Teague, V.: Strategies for Voter-Initiated Election Audits. In: 7th International Conference on Decision and Game Theory for Security (GameSec 2016), New York, NY, USA Nov. 2–4, 2016. Vol. 9996. LNCS, Springer, pp. 235–247, 2016, DOI: 10.1007/978-3-319-47413-7_14.

[De24a] DecentraVote: Decentralized e-voting protocol powered by blockchain, 2024, URL: https://decentra.vote/.

[De24b] DecentraVote: DecentraVote-Core, 2024, URL: https://github.com/DecentraVote-eVoting/DecentraVote-Core.

[El24a] Electis: Electis Core, 2024, URL: https://gitlab.com/electisNGO/electis-core/.

[El24b] Electis: Vote électronique CSE, 2024, URL: https://www.tessi.eu/.

[El24c] Electric paper Wahlsysteme: Wahlen sicher und effizient durchführen, 2024, URL: https://wahlen-organisieren.de/.

[El24d] Eligo: Electronic and online voting platform, 2024, URL: https://www.eligo.social/.

[EV24a] EVoting: Electronic Voting Service, 2024, URL: https://www.evoting.com/.

[EV24b] EVoting: Encryption and voting secrecy, 2024, URL: https://www.evoting.com/en/seguridad-integral/secreto-voto/.

[Fa20] Fazekas, Z.: DecentraVote: Electronic Voting secured by Blockchain (Version 1.0), tech. rep., DecentraVote, 2020, URL: https://github.com/DecentraVote-eVoting/DecentraVote-Core/blob/72c0a20ed1764fb5e80d4475934965a2f57a743e/Whitepaper_DecentraVote.pdf.

[FCC24] Finogina, T.; Cucurull Juan, J.; Costa, N.: Selective comparison of verifiable online voting systems. Security and Privacy, 2024, DOI: 10.1002/spy2.394.

[Fo24a] Followmyvote: Blockchain Voting: Cryptographically Secure Voting, 2024, URL: https://web.archive.org/web/20240820075328/https://followmyvote.com/cryptographically-secure-voting-2/.

[Fo24b] Followmyvote: Followmyvote SmartContract code, 2024, URL: https://github.com/FollowMyVote/Pollaris-Contract.

[Ge24] Gediote: Expert en solutions de vote, 2024, URL: https://www.gedivote.fr/.

[Gl24] Glondu, S.: Belenios specification (Version 2.5.1), tech. rep., Belenios, 2024, URL: https://web.archive.org/web/20240708064854/https://www.belenios.org/specification.pdf.

[He24] Helios: Helios-Server, 2024, URL: https://github.com/benadida/helios-server/tree/c7ed0608e2cdfc75bf323a834f0ca579e6273e34.

[HGB23] Heinl, M.; Gölz, S.; Bösch, C.: Remote Electronic Voting in Uncontrolled Environments: A Classifying Survey. ACM Computing Surveys 55 (8), 167, pp. 1–44, 2023, DOI: 10.1145/3551386.

[HR21] Haines, T.; Rønne, P.: New Standards for E-Voting Systems: Reflections on Source Code Examinations. In: International Workshops on Financial Cryptography and Data Security (FC 2021), Revised Selected Papers, Virtual Event Mar. 5, 2021. Vol. 12676. LNCS, Springer, pp. 279–289, 2021, DOI: 10.1007/978-3-662-63958-0_24.

[In24] Inria@CNRS: belenios, 2024, URL: https://gitlab.inria.fr/belenios/belenios/.

[IT23] IT Security Evaluation Facility of Deutsche Telekom Security GmbH: Protection Profile for E-Voting Systems for non-political Elections: BSI-CC-PP-0121 (Version 0.9), tech. rep., Federal Office for Information Security Germany - BSI, 2023, URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Schutzprofile/BSI-CC-PP-0121-Protection-Profile-for-E-Voting-Systems.pdf.

[Ke24] Kercia: Solutions de vote électronique sécurisées, 2024, URL: https://www.kercia.com/.

[KHC22] Kho, Y.; Heng, S.; Chin, J.: A Review of Cryptographic Electronic Voting. Symmetry 14 (5), 858, 2022, DOI: 10.3390/sym14050858.

[Kr23] Krips, K.; Snetkov, N.; Vakarjuk, J.; Willemson, J.: Trust Assumptions in Voting Systems. In: ESORICS 2023 International Workshops on Computer Security, Revised Selected Papers, The Hague, The Netherlands Sept. 25–29, 2023. Vol. 14399. LNCS, Springer, pp. 309–329, 2023, DOI: 10.1007/978-3-031-54129-2_18.

[Le24] LegaVote: Vous avez un projet de scrutin, nous avons une solution de vote électronique, 2024, URL: https://www.legavote.fr/.

[LKZ14] Li, H.; Kankanala, A.; Zou, X.: A taxonomy and comparison of remote voting schemes. In: 23rd International Conference on Computer Communication and Networks (ICCCN 2014), Shanghai, China Aug. 4–7, 2014. IEEE Computer Society, pp. 1–8, 2014, DOI: 10.1109/icccn.2014.6911807.

[Mo20] Moore, L.: Voatz Mobile Voting Platform – An Overview: Security, Identity, Auditability (Version 1.1), tech. rep., Voatz, Inc., 2020, URL: https://web.archive.org/web/20240822073616/https://voatz.com/wp-content/uploads/2020/07/voatz-security-whitepaper.pdf.

[MT23] Müller, J.; Truderung, T.: CAISED: A Protocol for Cast-as-Intended Verifiability with a Second Device. In: 8th International Joint Conference on Electronic Voting (E-Vote-ID 2023), Luxembourg City, Luxembourg Oct. 3–6, 2023. Vol. 14230. LNCS, Springer, pp. 123–139, 2023, DOI: 10.1007/978-3-031-43756-4_8.

[Ne21] Nemes, M.; Schwerdt, R.; Achenbach, D.; Löwe, B.; Müller-Quade, J.: And Paper-Based is Better? Towards Comparability of Classic and Cryptographic Voting Schemes. IACR Cryptology ePrint Archive, 2021, Report 2021/1122.

[Ne24a] Neovote: Neovote, 2024, URL: https://www.neovote.com/.

[Ne24b] Neuvote: Online Voting, 2024, URL: https://www.neuvote.com/.

[NNV17] Neumann, S.; Noll, M.; Volkamer, M.: Election-Dependent Security Evaluation of Internet Voting Schemes. In: 32nd IFIP TC 11 International Conference on ICT Systems Security and Privacy Protection (SEC 2017), Rome, Italy May 29–31, 2017. Vol. 502. IFIP Advances in Information and Communication Technology, Springer, pp. 371–382, 2017, DOI: 10.1007/978-3-319-58469-0_25.

[nV24] nVotes: Secure Online Voting Software, 2024, URL: https://nvotes.com/.

[On24] Onlz: Electronic voting you can trust. 2024, URL: https://www.onlz.com/.

[Pa21] Park, S.; Specter, M.; Narula, N.; Rivest, R.: Going from bad to worse: from Internet voting to blockchain voting. Journal of Cybersecurity 7 (1), 2021, DOI: 10.1093/cybsec/tyaa025.

[Pa24] Park, S.; Choi, J.; Kim, J.; Oh, H.: zkVoting: Zero-knowledge proof based coercion-resistant and E2E verifiable e-voting system. IACR Cryptology ePrint Archive, 2024, Report 2024/1003.

[PO24] POLYAS: POLYAS, 2024, URL: https://www.polyas.de/.

[Ro22] Robles, E.: Sequent Voting SystemArchitecture Overview (Work in progress), tech. rep., Sequentech, 2022, URL: https://web.archive.org/web/20240830084927/https://sequentech.github.io/documentation/assets/files/2022-04-10-arch-1-5bc31f49172dc628367719c7785abe36.pdf.

[Ru21] Ruescas, D.: Sequent Tech Cryptographic Protocol, tech. rep., Sequentech, 2021, URL: https://web.archive.org/web/20240830085125/https://sequentech.github.io/documentation/assets/files/2021-03-19-proto-1-18fd36c9669813aadcf59e672f0f7a84.pdf.

[Sc24] Scytl: Secure Online Voting, 2024, URL: https://scytl.com/.

[Se24a] Sequent: Open source end-to-end verifiable online voting, 2024, URL: https://sequentech.io/.

[Se24b] Sequent: Sequent, 2024, URL: https://github.com/sequentech/.

[Se24c] Sequent: Sequentech docs, 2024, URL: https://sequentech.github.io/documentation.

[Si24a] Simply Voting: Checking, Interpreting, and Publishing Results, 2024, URL: https://web.archive.org/web/20240711091037/https://help.simplyvoting.com/docs/checking-interpreting-and-publishing-results.

[Si24b] Simply Voting: Simply Voting, 2024, URL: https://www.simplyvoting.com/.

[SJT08] Scarfone, K.; Jansen, W.; Tracy, M.: Guide to General Server Security, tech. rep. Special Publication 800-123, National Institute of Standards and Technology (NIST), 2008, DOI: 10.6028/nist.sp.800-123.

[Sw24a] Swiss Federal Chancellery: Examination of systems, 2024, URL: https://www.bk.admin.ch/bk/en/home/politische-rechte/e-voting/ueberpruefung_systeme.html.

[Sw24b] Swiss Post: E-Voting: Online voting and elections, 2024, URL: https://digital-solutions.post.ch/en/e-government/digitization-solutions/e-voting.

[Sw24c] Swiss Post: swisspost-evoting, 2024, URL: https://gitlab.com/swisspost-evoting.

[Sw24d] Swiss Post Ltd.: Swiss Post Voting System – System Specification (Version 1.4.1), tech. rep., Swiss Post Ltd., 2024, URL: https://gitlab.com/swisspost-evoting/e-voting/e-voting-documentation/-/blob/a3e1063615e2efcdf2692e8a47697daf7ccdb2d1/System/System_Specification.pdf.

[Sw24e] Swiss Post Ltd.: Swiss Post Voting System Verifier Specification (Version 1.5.2), tech. rep., Swiss Post Ltd., 2024, URL: https://gitlab.com/swisspost-evoting/e-voting/e-voting-documentation/-/blob/a3e1063615e2efcdf2692e8a47697daf7ccdb2d1/System/Verifier_Specification.pdf.

[Te21] Teague, V.: Which E-Voting Problems Do We Need to Solve? In: 41st Annual International Cryptology Conference on Advances in Cryptology (CRYPTO 2021), Virtual Event Aug. 16–20, 2021. Vol. 12825. LNCS, Springer, pp. 3–7, 2021, DOI: 10.1007/978-3-030-84242-0_1.

[Tr23] Truderung, T.: POLYAS 3.0 Verifiable E-Voting System (Version 1.3.2), tech. rep., Berlin, Germany: POLYAS GmbH, 2023, URL: https://github.com/polyas-voting/core3-verifiable-doc/blob/d23fdb8d2627e17b181e06f97fb44e38865bf157/pdf/polyas3.0-verifiable.pdf.

[Tr24] Truderung, T.: POLYAS-Core3 Second Device Protocol (Version 1.1), tech. rep., Berlin, Germany: POLYAS GmbH, 2024, URL: https://github.com/polyas-voting/core3-verifiable-doc/blob/432434bb85ee24e5aa9caaacf3e5e24bc6d50708/pdf/second-device-spec.pdf.

[V824] V8TE: Online voting platform 100% self-service, 2024, URL: https://www.v8te.com/.

[Va22] Valimised.ee: IVXV protocols: Specification (Version 1.8.0), tech. rep., Valimised.ee, 2022, URL: https://web.archive.org/web/20240623062715/https://www.valimised.ee/sites/default/files/2023-02/IVXV-protocols.pdf.

[Va24] Valimised Eestis: IVXV, 2024, URL: https://github.com/valimised/ivxv/.

[Vo24a] Voatz: Voatz secure and convenient voting anywhere, 2024, URL: https://voatz.com/.

[Vo24b] Voxaly: Vote électronique - Voxaly vous accompagne, 2024, URL: https://www.voxaly.com/vote/electronique/.

[VV08] Volkamer, M.; Vogt, R.: Common Criteria Protection Profile for Basic set of security requirements for Online Voting Products: BSI-PP-0037 (Version 1.0), tech. rep., Federal Office for Information Security Germany - BSI, 2008, URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Reporte/ReportePP/pp0037b_engl_pdf.pdf.

[We24] WeChooz: Vote électronique - Solutions de vote par internet sécurisées - WeChooz, 2024, URL: https://www.wechooz.fr/solution-de-vote-electronique/.

[Wi18] Willemson, J.: Bits or paper: Which should get to carry your vote? Journal of Information Security and Applications 38, pp. 124–131, 2018, DOI: 10.1016/j.jisa.2017.11.007.

[Wi24] Wikström, D.: Open Verificatum, 2024, URL: https://www.verificatum.org/.

Electronic Voting in Albania: Pilot Implementation, Insights, and Evaluations;

Elira Hoxha [1], Jona Josifi [2], and Daniela Alimemeti[3]

Abstract: This paper examines the implementation and outcomes of electronic voting (e-voting) in Albania. As e-voting systems are increasingly adopted globally to enhance electoral efficiency, transparency, and accessibility, this study focuses on Albania's specific context as a transitional democracy. The research investigates the theoretical foundations of e-voting, its global practices, and the associated benefits and challenges. However, it also addresses significant challenges, including security concerns, the digital divide, and public skepticism. The paper presents a detailed analysis of Albania's e-voting pilot project, covering three consecutive elections. The findings provide valuable insights into the practicalities of integrating technology into the electoral process from the point of view of two out of the three authors that were part of the planning and implementation team in the Albanian EMB, contributing to the broader understanding of e-voting.

Keywords: e-voting, configuration, deployment, metrics, services, support, retrieving.

1 Introduction

Electronic voting (here and after referred to as e-voting) systems have been increasingly adopted worldwide to enhance efficiency and accessibility of the electoral process. The implementation of e-voting varies significantly across different countries, influenced by their unique political, technological, and cultural contexts. This literature review examines the theoretical foundations, global practices, benefits, and challenges associated with e-voting, with a specific focus on its practical application in Albania.

The theoretical underpinnings of e-voting are rooted in the principles of electronic governance and digital democracy. E-governance aims to leverage information and communication technologies (ICT) to improve the delivery of government services, enhance public participation, and ensure transparent decision-making processes [KB06]. Digital democracy extends this concept to the electoral domain, advocating for the use of ICT to facilitate voter registration, ballot casting, vote counting, and result dissemination [No01]. Several countries have successfully implemented e-voting systems, each with distinct approaches and outcomes. Estonia is a pioneer in this field, having introduced Internet voting in 2005. The Estonian model is lauded for its security measures, including the use of digital signatures and blockchain technology [ZT19]. Brazil, on the other hand, employs direct recording electronic (DRE) machines across its vast territory, emphasizing the system's

[1] University of Tirana, Department of Statistics and Applied Mathematics, Tirana, Albania, elirahoxha@yahoo.com, https://orcid.org/0000-0003-3459-2633

[2] University of Medicine , Tirana, Albania, Jona0josifi@gmail.com, https://orcid.org/0009-0008-7869-8425

doi:10.18420/e-vote-id2024_11

robustness and scalability (Hall et al., 2009). The advantages of e-voting include improved accessibility, especially for disabled and remote voters, faster and more accurate vote counting, reduced human error, and enhanced voter turnout [AH08]. Furthermore, e-voting can increase public trust in the electoral process by providing transparent and verifiable results [CC97]. Despite its potential benefits, e-voting faces significant challenges. Security concerns, such as hacking, malware, and cyber-attacks, are paramount [Ru02]. Additionally, the digital divide, characterized by unequal access to technology, can disenfranchise certain voter groups [No01]. Skepticism about the integrity and reliability of e-voting systems can also undermine public confidence [Me02]. The authors, two of which are part of the implementation team on behalf of the Albanian EMB, have carefully reviewed the previous research on Albania's e-voting system to the E-Vote-ID 2021 Conference with conclusions drawn from the first piloting of this project in 32 polling stations. Some of the conclusions drawn in the research have been addressed in the following pilots of 2022 and 2023, as explained below:

- Although electronic voting was still implemented with less than 10 percent of the voters list, this percentage has significally increased to approximately 8.5 percent of the total voters list for the local elections of May 14th 2023 (voters list for ELbasan Kamëz Vora 310 846 out of 3 650 202);

- There was a thorough manual ballot verification conducted, for 2023 elections it happened in 10 percent of the ballot boxes where electronic voting was piloted (Vora, Kamëz, Elbasan), 100 percent for the partial local elections of Vora on March 6th 2022 and 100 percent of the votes for the first piloting in 2021. This is a mandatory legal binding process (article 167/2 of the Electoral Code) that the EMB has to perform and it's live streamed. The same applies to the manual audit for tallying.

- The invalid votes printed as QR-Codes on PAT, making it difficult for voters to verify their choice has been addressed with the option/selection 'I don't vote for anyone', so the voter either chooses or chooses to not choose any option.

- With regards to the conclusion that no security required during electronic transfer of results, this was not applicable for the 2021 and 2022 because there was no transmission of results.

However there is still the need to address the other conclusions drawn in that research and this one. The implementation of e-voting in Albania represents a critical case study for examining the practicalities and challenges of introducing such technology in a transitional democracy. Previous attempts at electoral reforms in Albania have highlighted issues such as voter fraud, administrative inefficiencies, and political distrust [Sc99]. This context underscores the importance of a carefully managed e-voting pilot project [Bu].

2 Methodology

This study employs a mixed-methods research design to explore the practical usage of e-voting identification in Albania's elections [VM22] . A thorough review of existing literature on e-voting systems, global practices, and Albania's electoral context forms the foundation of this research. A detailed case study of the e-voting pilot project in Albania will be conducted, focusing on its planning, execution, and outcomes. This will include an analysis of technical specifications, security measures and logistical challenges.

This paper will be organized as follows: To assist the reader to comprehend the notion of E-Voting in Albania, Section 3 covers background of our work on implementing E-voting currently in Albania. Following in Section 4 (Infrastructure and Solutions) used will be presented to the reader. Lastly this paper will follow with Section 5 (Data Handling) and Section 6 (Logistical Implications).

It is not in the scope of this paper to address or analyze the project/solution cost implications.

3 Background

The electronic voting project piloted in elections in Albania for the third time in 2023, as per this paper, aimed to introduce and further extend the application of the Electronic Voting and Results Management System from 2021 up to May 2023. This solution was implemented for the first time for the Parliamentary Elections of 2021, for which the Central Elections Commission (from here on referred to as CEC) of Albania purchased the Electronic Voting and Results Management System (from here on referred to as EVRMS) central system, as well as an initial group of Electronic Voting and Counting Devices (from here on referred to as EVCD).

CEC extended 740 additional EVCDs to support electronic voting in the target polling stations - which included two (2) EVCDs per each polling station and 5% of EVCDs as contingency, as well as the necessary software configuration/enhancements to the existing EVRMS as required, related implementation and support services. Also, as part of this project, the CEC did deploy the existing EVCDs already purchased in 2021. The EVCDs enabled electronic voting and calculation of the final election result in the EVRMS at the level of the polling station after the voting process was closed. Moreover, as a new functionality, the EVCDs were able to transmit electronically the election results to the central EVRMS.

The technical solution included:

- 740 EVCDs
- 1,480 removable storage devices for the storage of the votes and results for each EVCD (including redundancy storage).

- 1,480 smartcards for the access of the EVCDs administrative functions by the poll workers
- 2,424 paper rolls to print the voter receipts
- 808 communication devices to enable the electronic communication of EVCD for the new EVCD and the ones already purchased in 2021, as well related communication services to enable electronic transmission capability
- 740 External Batteries cables to ensure the operation of the EVCDs in case of an energy interruption at the Polling Station.
- Enhancements/configuration to the existing EVRMS

This proposed solution enabled the electronic voting by accomplishing the following:

- Establish the necessary technological infrastructure to enable electronic voting in the target polling stations in Albania
- Dispose of any invalid ballots by preventing a voter from voting for more than one selection or making incorrect entries, according to the preliminary rules adopted by the CEC
- Securely and anonymously store the votes and simultaneously reproduce the hard copy confirmation/in the paper of the vote, which will be shown to the voter for his verification and confirmation, after which the vote paper receipt will be cut and dropped to the ballot box
- Calculate the voting result for each polling station in a quick and accurate way
- Establishment of the process to safely send data, as approved by the voting commission, from each EVCD, ONLY upon the closing of the voting process at each Polling Station as part of the project
- Once the election process is closed at the Polling Station level, it enables the electronic transmission from each EVCD to the EVRMS central system, to consolidate the election results from the EVCDs from all polling stations

4 Infrastructure and operative model of electronic voting for Albania

4.1 Solution implemented

The solution implementation team implemented a technological system that enabled the process of electronic voting in Albania for the Local Elections of 2023, in the targeted polling stations. The system enabled electronic voting through EVCDs purposely designed for electronic voting, which were compatible in form and technical performance with the EVCDs that the CEC already purchased in 2021, although the EVCD was an improved version/model. The team also customized the Central EVRMS System to integrate it with the

EVCDs, to receive the electoral configuration files generated by the EVRMS, and to transmit the election results files. It is to be noted that transmission of results was implemented for the first time in the 2023 elections, though it wasn't in 2021 and 2022.

The solution implementation team guaranteed the appropriate audit levels of the technological processes part of this project and the security provisions according to the highest technological standards. Each EVCD was physically sealed with security seals and subsequently sealed in the premises of the CEC, denying any kind of access (computer communication gates) to any premises other than the premises of the CEC, in compliance with a protocol duly approved by the CEC.

The solution implementation team set the parameters of the entire system and any device which was part of this project, in accordance with the legal acts of the CEC for the 2023 Elections. The team ensured that the EVCD automatically, anonymously, and independently generated a hard copy document (voter receipt) with the respective ballot of each voting session and automatically stored it via an inaccessible way throughout the voting process. When more than one election was held simultaneously, the EVCD generated a voter receipt for each ballot. This hard copy receipt was then (6 months post elections) used for audit/verification purposes solely upon the closing of the voting process at the polling station level.

The solution implementation team guaranteed to store the ballot securely and anonymously in the EVCD, but at the same time generated a hard copy document/receipt of the vote of each voter, which was shown to the voter. Such vote receipt was then cut and dropped into the ballot box after being printed. The EVCD's on-screen interface of the voting process was user-friendly, understandable, and clear as much as possible for each voter, although there were some 'non-familiarity' issues raised mainly in one municipality.

The EVCD's on-screen interface was able to present the ballot, including all electoral contests and data according to the ballot paper format approved by CEC, and the candidates of the multi-name lists registered with the CEC.

The solution implementation team guaranteed that the electronic voting process was the shortest possible, provided that the voter was also to be requested to confirm his/her vote. Only after the voter has selected the candidate/party of his/her choice and only after he/she has re-confirmed the action by reviewing the selections, then the device proceed with the next consecutive process and stored the vote in the EVCD. When more than one election is held simultaneously (as was the case for the 2023 elections: one ballot for mayor and another for municipality council), the EVCD requested confirmation of the selections for each ballot before moving to the next one. The voting interface was, of course, in the Albanian language.

The solution implementation team provided logistics of means, equipment, and technical assistance to support any issues concerned throughout the voting process, in each of the regions where this system was to be implemented. As per the Albanian EMB human resource capacities and/or local ones, covering of different processes with staff differed from one pilot/election year to another. In 2021, almost all technical, logistical and field support capabilities were covered by the vendor, and the EMB relied in local capacities only for the polling station operator of the EVCD (one person/polling station). In 2023, for the expansion of the pilot project to 401 polling stations less responsibilities were attributed to the vendor and more to local capacities. in terms of services, the EMB with its own staff (permanent and temporary) covered logistical arrangements (polling station operators, trainer of the PS operator, distribution of EVCDs in each of the 401 PSs). The vendor covered: training of trainers, field support staff during e-day, configuration of EVCDs, EVRMS central system configuration and enhancements. so, technically and practically speaking, the EMB capabilities and experience in managing this process has improved, however there is still a moderated dependence from the vendor mostly in regards to system configuration and enhancements.

Fig. 1: EMB training of the PS operators for the local elections of 2023 [KQb]

4.2 System architecture

Two main components of this project are the EVCDs and EVMRS.

4.2.1 EVCD

EVCD is used to display the ballot paper on the screen in a clearly and easily identifiable by the voter. The EVCD uses a secure UEFI boot implementation to protect the system from malicious codes that can be loaded and executed at the beginning of the device boot process, even before the operating system is loaded. This implementation validates/verifies the boot path of the device and includes not only the "boot loader"but also the kernel and operating system. If the system encounters a manipulated boot path it will block the device’s ignition sequence. This however did not happen. The EVCD operating system has only the necessary software modules/packages for the voting process and does not have any other unnecessary software modules/packages. The EVCD Operating System has all the necessary security updates that the specific operating system has. The EVCDs provide the following levels of security against unauthorized login:

- During configuration, the removable memory that is used to transfer the configuration is synchronized with the appropriate data.
- In case the removable memory is removed from the EVCD, the system detects it and immediately blocks the operation of the EVCD.
- Removable memories are installed and physically positioned in such a way that they are protected with CEC seals so that unauthorized persons cannot remove them.
- Each EVCD port compartment allows the possibility to be sealed with security seals by the CEC.
- The EVCD provides the functionality to identify the polling station worker who will operate it through a two-factor authentication mechanism.
- The EVMRS central system has the capacity to (for each EVCD) generate a smart card and a respective PIN code. Both are made available by CEC only to the relevant PS worker of that polling station so that the PS worker can be identified for using an EVCD by a smart card or by entering a PIN code or a combination of both.

EVCD guarantees the following levels of voting secrecy:

- The anonymity and secrecy of the ballot by never retaining any voter identification data. Nothing about voter identity is ever programmed or registered in the EVCD. In this way, the EVCD cannot identify which voter is voting at a given moment.
- The EVCD deletes any voting data at the end of the voting session, safely, so that selection from a current voting session is no longer available once the session ends.
- The EVCD stores the votes at a random position in the system files. This ensures that the voting order in the EVCD cannot be saved and consequently a specific voter's vote cannot be identified by following the voting order in the PS.

EVCD ensures that:

- Every file that the EVCD generates uses the Authenticated Encryption method with Encrypt-then-MAC.
- Authenticated Encryption involves the use of AES-256 to encrypt all files.
- EVCD should not have any active wireless communication (Wi-Fi, Bluetooth, mobile GSM antenna) **during** the voting process.

The transfer of data by each EVCD is conducted through its own physical memory or through direct electronic transmission to the EVRMS, only after the voting process has been closed, and the results of that Polling Station have been printed. EVCD has the following features:

- Portrait touch screen, minimum 17-Inch capacitive.
- Possibility to move the screen angle to adapt to the voting environment in the PS.

- Manual remote activation button with a cable length of at least five (5) meters to guarantee the privacy of the voter. The button is located on the PS worker's desk. The EVCD voting session is activated manually by this button which must be physically pressed by one of the PS workers. The button has an integrated identification light to indicate the status of the EVCD: the device is switched off (the voting process has not yet started in the polling station or the voting process has ended), activated if someone is voting and the button can not be pressed, passive when the EVCD is free to wait for the next voter.
- The EVCD is an assembled, integrated device, dedicated to electronic voting.
- The EVCD is an easily transportable device.
- Includes a thermal printer for reports and verification receipts with a minimum paper width of 75 mm, with an automatic cutter. The thermal printer has an integrated space for the paper roll and is secured with CEC security seals.
- Part of the EVCD is the translucent box for storing the votes' physical verification. The printer is integrated with a storage box for ballot paper verification receipts. This storage box is provided as a separate unit, in which case, the EVCD is connected to the corresponding thermal printer and the box, to automatically print, cut, and deposit the paper receipts into the ballot box.
- The ballot storage box is not completely transparent (semi-transparent) to guarantee the anonymity of the ballot.
- The visible window where the voter can see his/her ballot paper receipt (but without accessing it) is at least 75 mm wide.
- The paper compartment of the printer allows visibility of the paper roll without the need to open such a compartment.
- The printer and storage box are sealable with security seals.
- The capacity of the paper ballot storage box is up to five hundred (500) voting receipts.
- The EVCD has a smart card reading module, for the identification of the user of the device.
- EVCD allows the connection of external batteries to be independent of electricity if power is interrupted in the polling place.
- EVCD has a specific design for use in electoral processes, portable and assembled to ensure the shortest possible configuration and packaging time.
- USB removable memory and a backup memory. Each EVCD stores voting data securely in a removable USB, as well as another removable backup USB, for redundancy and verification purposes.
- Removable USBs are previously prepared/configured with data belonging to that PS, electronic ballot paper etc., and are physically protected in the EVCD body in a

designated compartment physically secured by CEC security seals, as shown in the figure below:

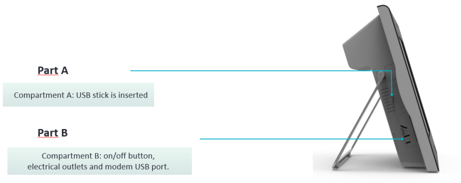

Part A
Compartment A: USB stick is inserted
Part B
Compartment B: on/off button, electrical outlets and modem USB port.

Fig. 2: EVCD compartments secured with security seals

- The EVCD allows the incorporation of the CEC security seals on the on/off button as well as on its other ports.
- The EVCD doesn't have any network ports/interfaces activated **during** the voting process. In addition, no communication module should be connected to the EVCD throughout the PS voting process. -
- The EVCD operating system blocks any communication port in the network during voting hours.
- EVCD allows the incorporation of CEC security seals (tamper evident seal) also for the thermal printer. The thermal paper used in EVCD is marked with unique secret codes of the specific EVCD to immediately identify any other thermal printing.

4.2.2 EVMRS

Before each voting process, the first step is configuring the respective election process in the EVMRS. The EVMRS then generates all the required parameters for each EVCD respectively for each Polling Station. EVMRS operates on a private network on the CEC premises. For each election process, EVMRS generates public and private keys to manage all digital certificates which are mandatory throughout the voting process. Each EVCD has its own unique digital certificate and a certificate to ensure the secure transfer of the results after the closing of the voting process. The exchange of parameters in the CEC private network from EVMRS to EVCDs for each Polling Station is conducted through verified data accompanied by their respective certificates. This ensures that EVCDs are set only from a secure source identified and verified with their certificate. This also avoids any kind of EVCD setting from any source other than the EVMRS installed at the CEC and verified as per its own certificate for the concrete elections being worked on. The final voting results are transferred from the EVCDs to the central EVRMS, only after the completion of the voting process in the PS, through physical memories or by a direct electronic transmission. In either case, this is asymmetrically coded and signed with a digital stamp. Upon extracting the results from memories or receiving the direct electronic transmission, the central EVRMS verifies the data by decoding them. In addition, the system performs a check to verify that the data is coming from an accurate EVCD by checking its digital stamp. The central system EVRMS constantly generates logs for each event, so that any issues with the setting process is resolved as quickly as possible. Each file generated by EVMRS is digitally signed.

5 Data Handling

The life-cycle of the data in an Electronic Voting System can be summarized in Figure 3:

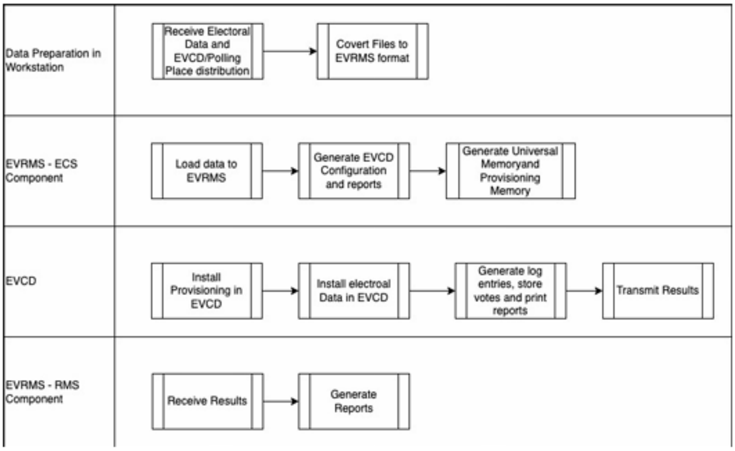

Data Preparation in Workstation
Receive Electoral Data and EVCD/Polling Place distribution
Covert Files to EVRMS format
EVRMS - ECS Component
Load data to EVRMS
Generate EVCD Configuration and reports
Generate Universal Memoryand Provisioning Memory
EVCD
Install Provisioning in EVCD
Install electroal Data in EVCD
Generate log entries, store votes and print reports
Transmit Results
EVRMS - RMS Component
Receive Results
Generate Reports

Fig. 3: Data life-cycle throughout the process

Electronic Voting System implements a data scheme that is the result of the evaluation and implementation of different and worldwide electoral frameworks, geopolitical distributions, electoral events characteristics with regard to contests, and tally rules. Because of this, there is a minimum set of information that needs to be loaded with a specific format and relationship so it can be accepted by EVMRS. Once the raw data is received from the CEC, it is formatted, and any missing information required by the system that is not part of (or does not affect) the CEC data, is completed standard tools such as a text editor and/or EXCEL. This task is not performed by any component of the solution, and it is performed in constant communication between the CEC and the vendor. At this stage, the data is not encrypted, nevertheless is considered trusted information considering its source and reviewer is the CEC. It is important to mention that part of the deployment process includes a User Acceptance Test (UAT) in which the CEC validates the data loaded in the system. For the 14 of May 2023, CEC performed the UAT on May 4th, 2023 pictures below:

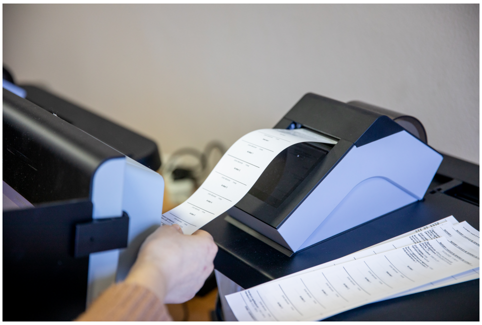


Fig. 4: EVCD report printing [kqa]

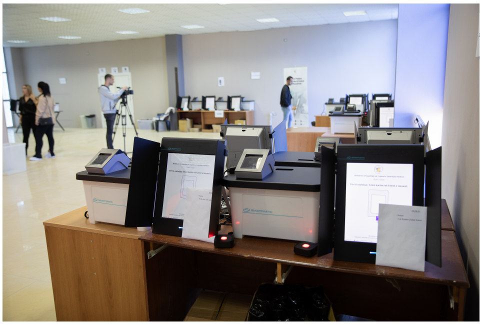


Fig. 5: EVCD screen, voting flow [kqa]

5.1 Data handling in the EVRMS

The interface offered by the EVRMS, either via Bulk Data Load files (plain text files) or the task offered for each type of data. The EVMRS ensures all the necessary data is loaded before it is possible to generate the necessary configuration files for the EVCDs. All electoral data is persisted in a relational database with limited access only to the EVRMS application. Only the EVRMs application accesses the database, and it should only do so in the election data center; the data access occurs throughout the electoral events. There are additional controls implemented only for the sensitive data stored in the EVRMS Database; in addition to the security controls described, the EVRMS software encrypts and completely restricts access to the sensitive data it stores in the database. When the EVRMS Software starts running, it asks for the passphrase it uses to generate an encryption key; this generated Encryption Key is never stored or shared, it is only kept available for the EVRMS software while it remains active. The fact that the encryption key is only available for the EVRMS software ensures that only the EVRMS software can decrypt the sensitive data encrypted with that key. None of EVRMS tasks allow access to the sensitive data in the EVRMS Database; although the EVRMS Software has access to sensitive data, but it does not allow its users to directly read or edit it. The configuration files package for the EVCDs are stored using AES256/CBC/PKCS7 encryption with a symmetric protecting key derived using PBKDF12/HMAC-SHA512 algorithm based on a random passphrase generated by the EVRMS application. This package can only be accessed by the EVCDs with proper input from the operator. Configuration file packages are generated directly to a USB memory with a valid file system and this memory is used by the EVCDs in the configuration warehouse during the readiness process, where the electoral configuration is installed in them. Users cannot install malware in any machine/EVCD because the EVMRS only recognizes the valid files for the system.

5.2 Data handling in the EVCD

The configuration package, which at this moment includes the data for all the polling stations, is received by the EVCD encrypted as a single file following the SE1 encoding as described below. The passphrase is provided by the warehouse operator, via the EVCD touchscreen, during the readiness process, along with the code of the polling station to install. Once the polling station to install in the device is confirmed, the EVCD re-encrypts the corresponding subset of files using the same scheme (SE1), but this time the passphrase is derived from the provisioning information previously installed in the EVCD. After the election is closed in the EVCD, the results are binary encoded and signed in a QR code with the private key assigned to the device in the configuration. The signed content is stored as a CMS-signed envelope.

In the three municipalities where electronic voting and counting was piloted the results came in from the EVCD immediately at the closing of the voting process at 7:00pm. The total result for the municipality of Vora (36 PS-s) was publicly made available at around 8:00pm of that same day. While, Kamëz and Elbasan municipality total result was publicly made available the next morning at around 7:00am. Anyhow, every EVCD immediately at the closing of the voting process printed out the report of the result for that PS which was first printed and put inside the ballot box, and then the other 7 other copies printed were given to the PS workers (who manage the voting process during e-day and are appointed by CEC Albania, proposed by all political parties). The evidence of such result is given to them without need to request it, they're fully entitled to have it. The manual count to check and compare the votes in the ballot box with those counted by the EVCD is a right that any political party and/or candidate can exercise if they have claims about the result. Practically, this right was exercised by the opposition ang granted form CEC Albania for two municipalities: Kamëz and Elbasan. Tabulation of the results, is made public live by the CEC in its official website, in the level of PS as well as administrative units, as well as municipalities. You can find such an example of PS nr. 1569/00 in Kamëz municipality in the following link: https://iemis.kqz.gov.al/results2023/results2023.htm?lang=AL.

6 Logistical implications

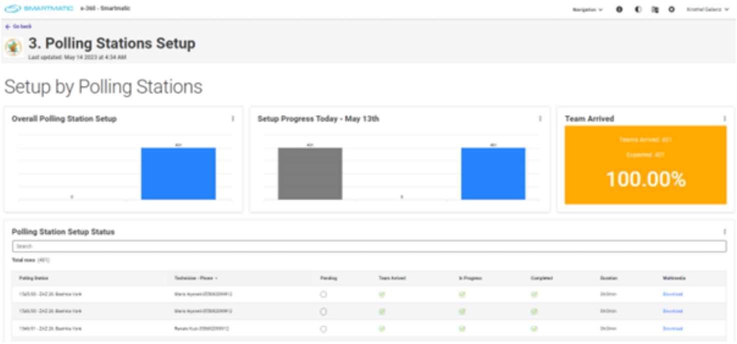
6.1 UAT KIT

Combo Name	SKU Content	Qty
6*UAT MAIN EVCD KIT	EVCD KIT - EVCD -Top Cover -Printer -Ballot box -Button	1
	USB modem	1
	USB Memories (1 main - 1 backup)	2
	Thermal paper rolls	2
	Bag with consumables for physical installation at the polling station.	1 power extension 1 power strip 1 duct tape
	white envelope	2 smartcards 1 configuration receipt 1 hat receipt 2 metal seals
6*UAT CONTINGENCY KIT	EVCD KIT - EVCD -Top Cover -Printer -Ballot box -Button	1
	USB modem	1
	USB Memories (1 main - 1 backup)	2
	Thermal paper rolls	2
	Bag with consumables for physical installation at the polling station.	1 power extension 1 power strip 1 duct tape
	white envelope	2 smartcards 1 configuration receipt 1 hat receipt 2 metal seals

Tab. 1: UAT KIT Contents

6.2 National Support Center

The National Support Center was activated on Saturday, May 13, and Sunday, May 14 2023. On this day, the process of physical installation of the EVCDs was carried out in each of the polling places. This activity was conducted by the EVCD operator assigned by the CEC. To guarantee the correct physical installation of the EVCDs and continuity of the voting process, also attend to any logistical, operational or technical incident, field support technicians were deployed through a detailed pre-planned geographical distribution to visit polling places in case support was needed. Once the support technicians had completed the visit and verified the information, they used the Smartsurvey where they placed the corresponding information and uploaded pictures of the EVCDs installed to provide status in real-time. Following are the monitoring dashboards for this activity as described in Figure 2


SMARTMATIC e-360 - Smartmatic
Navigation
Go back
Overall Polling Station Setup
401
0
Setup Progress Today - May 13th
401
0
401
Team Arrived
Teams Arrived: 401
Expected: 401
100.00%

3. Polling Stations Setup

Last updated: May 14 2023 at 4:34 AM

Setup by Polling Stations

Polling Station Setup Status

Search

Total rows (401)

Polling Station	Technician - Phone	Pending	Team Arrived	In Progress	Completed	Duration	Multimedia
[illegible]	[illegible]	○	✓	✓	✓	[illegible]	Download
[illegible]	[illegible]	○	✓	✓	✓	[illegible]	Download
[illegible]	[illegible]	○	✓	✓	✓	[illegible]	Download

Fig. 6: Monitoring Dashboard

Additionally, the operation of the National Support Center started at 8 am where 12 support agents and 6 dispatchers attended and coordinated the logistic, operational and technical problems reported. The problems were reported and attended to at the National Support Center on the day of the physical installation of the EVCDs at the polling stations were:

- 14 ticked created
- 100% of the tickets were closed and solved successfully
- 1 EVCD replacement executed, which represents 0.12% percentage of failure of the total of 802 EVCDs deployed in the field

Additionally, the operation of the National Support Center started at 6 am where 12 support agents and 6 dispatchers attended and coordinated the logistic, operational and technical problems reported. The problems were reported and attended to at the National Support Center on the day of the physical installation of the EVCDs at the polling stations were:

- 57 ticked created

- 100% of the tickets were closed and solved successfully
- 1 EVCD replacement executed, which represents 0.12% percentage of failure of the total of 802 EVCDs deployed in the field

The aforementioned information was graphically represented in real time on our dashboards, Figure 3 and Figure 4:

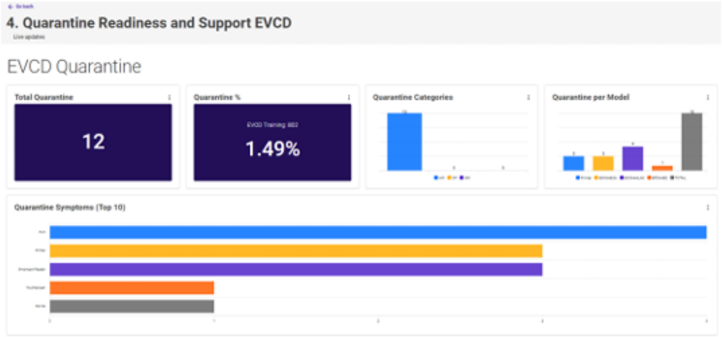
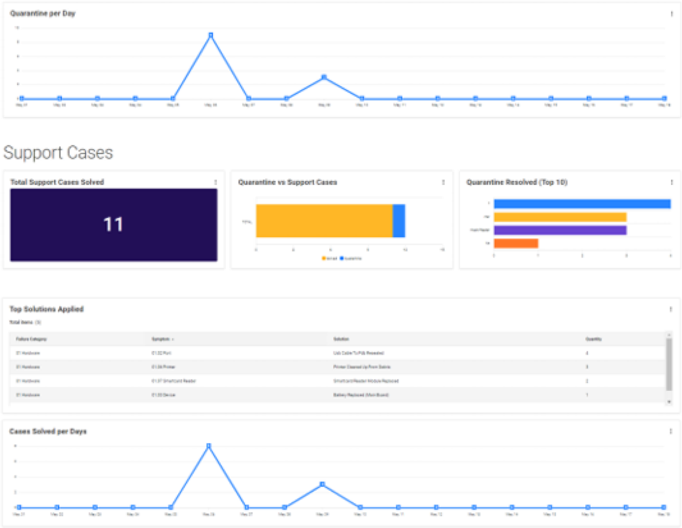


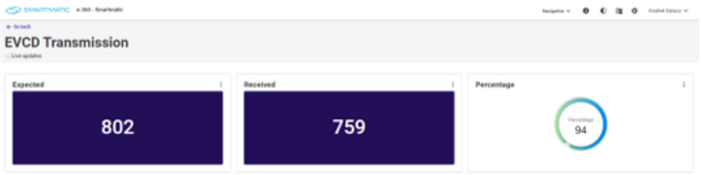
Fig. 7: Ticked System Dashboard



Failure Category	Symptom	Solution	Quantity
01 Hardware	01.02 Port	Usb Cable To PIB Reseated	4
01 Hardware	01.04 Printer	Printer Cleaned Up From Debris	3
01 Hardware	01.07 Smartcard Reader	Smartcard Reader Module Replaced	2
01 Hardware	01.03 Device	Battery Replaced (Main Board)	1

Fig. 8: Real Time Monitoring Dashboard

At 7 pm on May 14, the transmission of results from the polling stations began. The National Support Center staff was supporting and instructing the EVCD operators on how to connect the modem and carry out the transmission. Finally, it was obtained as a result that 759 of 802 EVCDs transmitted successfully, which represents 94% transmission (Figure 5).



EVCD Transmission

Live updates

Expected: 802

Received: 759

Percentage: 94

Fig. 9: Transmission Dashboard

7 Conclusions

This study provides valuable insights into the practical usage of e-voting in Albania's elections. A project journey that started in only 32 polling stations to be extended in a number of 401 in 2023 local elections (12.5 times) despite being the same solution and methodology applied, resulted to great practical implications in terms of resource management (both logistical and human). By examining the e-voting pilot project's implementation and outcomes for three consecutive elections, this research contributes to the broader understanding of the benefits, challenges and future prospects of integrating technology into the electoral process in transitional democracies. The findings indicate that e-voting can significantly enhance the efficiency of timely counting and transparency of elections. The deployment of EVCDs demonstrated the potential for faster and more accurate vote counting, reducing the likelihood of human error and ensuring that results are available more quickly than with traditional paper ballots. The successful transmission rate of 94% in the 2023 local elections highlights the reliability of the system in real-world conditions.

However, the study also shows several challenges that need to be addressed to ensure the successful and sustainable implementation of e-voting systems. Security concerns remain of great importance, as the risk of cyber-attacks and hacking could undermine the integrity of the electoral process. Ensuring strong cybersecurity measures and continuous monitoring is essential to protect against potential threats.

The practical application of e-voting in Albania also revealed the importance of comprehensive voter education also as per OSCE/ODIHR Final Report on the Local Elections of May 14th 2023, recommendation no.12 [os], and training programs for poll workers. Both education for the voter and the poll worker is needed because of the number of assisted voting that was found mainly in two municipalities: Kamëz and Elbasan, in remote rural areas where the voter asked for assistance and the poll worker did assist. Political parties claimed that this was a risk that could lead to manipulation of the voters will or a threat to the vote secrecy. Such need to better tailor education programs was revealed during the appeal process. Ensuring that all participants are familiar with the technology and procedures is vital for the smooth operation of the system and for building public trust. The involvement of a National Support Center to assist with technical and logistical issues proved to be a valuable resource in managing the complexities of the e-voting process.

Several areas require further research and development. These include applying independent scrutiny and audit of the system which has not yet happened due to financial and time constraints (budget for elections in the country is always granted in the last 3-4 months prior to election day), as well as developing more user-friendly interfaces to improve accessibility for all voters. Additionally, conducting more extensive pilot projects in different regions of the country and under various electoral conditions will provide deeper insights into the scalability and adaptability of e-voting systems.

Bibliography

[AH08] ALVAREZ, R. MICHAEL; HALL, THAD E.: Electronic Elections: The Perils and Promises of Digital Democracy. Princeton University Press, 2008.

[Bu] Budurushi, Jurlind: Sixth International Joint Conference on Electronic Voting E-Vote-ID 2021. 5-8 October 2021 — dspace.ut.ee. https://dspace.ut.ee/items/89865075-d222-46bc-8747-26a7e6d4fdfc. [Accessed 30-08-2024].

[CC97] Cranor, L.F.; Cytron, R.K.: Sensus: a security-conscious electronic polling system for the Internet. In: Proceedings of the Thirtieth Hawaii International Conference on System Sciences. volume 3, pp. 561–570 vol.3, 1997.

[KB06] Kumar, Rajendra; Best, Michael L.: Impact and Sustainability of E-Government Services in Developing Countries: Lessons Learned from Tamil Nadu, India. The Information Society, 22(1):1–12, 2006.

[kqa] kqz: Testimi operacional i PEVN, Celibashi: Pajisjet garantojnë fshehtësinë e plotë të votës — kqz.gov.al. https://kqz.gov.al/2023/05/04/testimi-operacional-i-pevn-celibashi-pajisjet-garantojne-fshehtesine-e-plote-te-votes/. [Accessed 30-08-2024].

[KQb] KQZ: Zgjedhjet vendore 2023 - Trajnohen operatorët e pajisjeve të votimit dhe numërimit elektronik — kqz.gov.al. https://kqz.gov.al/2023/05/06/zgjedhjet-vendore-2023-trajnohen-operatoret-e-pajisjeve-te-votimit-dhe-numerimit-elektronik. [Accessed 30-08-2024].

[Me02] Mercuri, Rebecca: Government: a better ballot box? IEEE Spectr., 39(10):46–50, oct 2002.

[No01] Norris, Pippa: Digital Divide: Civic Engagement, Information Poverty, and the Internet Worldwide. Communication, Society and Politics. Cambridge University Press, 2001.

[os] osce: osce.org. https://www.osce.org/files/f/documents/d/d/553972.pdf. [Accessed 30-08-2024].

[Ru02] Rubin, Aviel D.: Security considerations for remote electronic voting. Commun. ACM, 45(12):39–44, dec 2002.

[Sc99] Schnytzer, Adi: Albania in Transition: The Rocky Road to Democracy. By Elez Biberaj. Nations of the Modern World: Europe. Boulder, Colo.: Westview Press, 1998. xiv, 377 pp. Notes. Bibliography. Index. Tables. 72.00, $hardbound.SlavicReview, 58(3) : 679\breve{}680$, 1999.

Vukatana, Kreshnik; Mata, Gerta: E-voting, a proposed framework for Albania scenario". GJIT, 12:51–58, 2022.

Zein, Reyan M.; Twinomurinzi, Hossana: Towards Blockchain Technology to Support Digital Government. In (Kő, Andrea; Francesconi, Enrico; Anderst-Kotsis, Gabriele; Tjoa, A Min; Khalil, Ismail, eds): Electronic Government and the Information Systems Perspective. Springer International Publishing, Cham, pp. 207–220, 2019.

Dream of i-voting versus the reality of digitization of electoral processes

Case study of Polish "Party Proxy" app

Sylwester Oracz[1] and Anna Frydrych-Depka[2]

Abstract: Most likely in the second half of 2025 the Polish Electoral Code is going to be significantly amended. One of the most reverberating ideas is the introduction of i-voting. To assess the chances of success, case study of the recently created system "Party Proxy"[3] was made. Both the system itself and the way in which it was implemented have raised many controversies. Combined with previous experiences in the digitization of electoral processes in Poland, this can lead to justified concerns about probable future works on i-voting.

Keywords: observation, digitization, Polish e-administration

1 Introduction

It is expected that Polish electoral law and the system is going to be significantly modified in the near future, most likely in the second half of 2025 (after the presidential elections). One of the most reverberating ideas is the introduction of i-voting. Unfortunately, the ongoing discussions on the topic are, on the one hand, full of optimistic assurances from politicians about the rapid implementation of the idea, and on the other hand, concerns regarding security repeated like a mantra by practitioners and experts. In our opinion, a moderate knowledge-based approach is not sufficiently strong. To assess the chances of success for the i-voting programme in Poland, we decided to study a case of the recently created (2023) IT system "Party Proxy", which consists of a webform, a database and 2 mobile applications (iOS and Android). While the system is still in use we focused on the first usage, while noting that it hasn't changed since then. "Party Proxy" is used to send photos and recordings of election committees work made by party proxies or observers. Unfortunately, both the system itself and the way it was implemented have caused controversy. In our opinion, combined with previous experiences [Mi16] in the informatization of electoral processes in Poland, this can lead to justified concerns that in such an important issue as i-voting, the governing bodies and implementers may lack common sense and "throw the baby out with the bathwater".

[1] Political Accountability Foundation, Poland, sylwester@odpowiedzialnapolityka.pl

[2] Nicolaus Copernicus University in Toruń, Centre of Electoral Studies, Toruń, Poland, aniafryd@umk.pl, https://orcid.org/0000-0002-3765-4046

[3] PL: "Mąż zaufania". The mobile application is available on both Google Play: https://play.google.com/store/apps/details?id=pl.mc.mzow and Apple's AppStore, and the registration form on https://bazamezowzaufania.gov.pl/

doi:10.18420/e-vote-id2024_12

While this paper does not describe electoral software in sense of e- or i-voting, some of the challenges are shared e.g. processing the political preferences, trust in the software, accountability, ownership and procurement. Seems that decision-makers do not foresee them as core issues before i-voting development starts. Polish politicians often present i-voting as remedy for voter mobilisation or as a requirement of the modern times.

2 Legal framework

Introduction of "Party Proxy" was another governmental electoral task without EMB supervision, but with questionable justification and potential risk of obtaining data on voters' political preferences. The amendment to the Electoral Code of January 26, 2023, extended the rights of party proxies and observers to record the entire electoral process on election day. In connection with this change, the legislator obliged the Minister of Digital Affairs to create a new digital system for collecting photos and recordings made them documenting the course of the process on election day. The intent was to make those materials evidences in case of an election protest or other ongoing legal proceedings related to electoral irregularities. However, they should be deleted after a court certifies the results.

The rationale for creating this system was questioned [Po23]. The idea of creating the service was never brought to the public debate until it appeared suddenly in December 2022 [WR24]. Since 2018 transferring the materials was possible in physical or digital form to the precinct commission itself as part of the election documents subject to archiving[4].

Between 2018 and 2022 the cases of transferring such materials have been negligible. With extension of party proxies and observers rights, number of restrictions on the use of the collected materials were introduced, explained by the need of protecting the personal data. It was ordered that they need to be immediately deleted from the devices of party proxies and observers. Initially, however, it was not specified to which date the deadline should refer: the recording, election day[5], announcement of results, expiration of the date for submitting election protests, or the certification by court.

Additionally, the requirement for upfront registration of party proxies was introduced, raising concerns about revealing the political preferences of tens of thousands of citizens[6]. Although the necessity only arose if the party proxy intended to record the commission's work and upload materials, a practical question emerges how were the committees to know whether such a need would arise or not. It seems that committee heads faced that dilemma and in the 2023 parliamentary elections, the so-called democratic opposition electoral committees reported party proxies, while the right-wing forces' committees, including Law and Justice, did not. In effect, the database contains over 10,000 people who planned performing the

[4] *v.* Art. 8 of Act on Electoral Code (Journal of Laws 31 January, 2011 with amendments), further referred to as *Electoral Code*

[5] *v.* Art. 42 § 6a of Electoral Code

[6] *cf.* Art. 9 of Regulation (EU) 2016/679, further referred to as *General Data Protection Regulation*

duty during parliamentary elections, even if they did not take up their privilege and decided not to transfer any materials.

Even at the conceptual stage, a number of doubts were raised. First and foremost, it was unclear why potential evidence indicating violations of election law should be sent to the minister a politician and usually a candidate. The electoral bodies seemed more appropriate, giving their independent role in the Polish electoral system. Only in February 2024 did the public learn from the then-former Minister of Digital Affairs Cieszyński that the National Electoral Office (NEO) did not want to run this system. However, it can only be speculated whether the NEO requested the government to create such a system or if it was an attempt by the Ministry of Digital Affairs, as the initiator, to meet in the middle with the electoral administration with burden of an additional task.

3 Implementation

The final result is far from perfect since endusers where not consulted. The proposal was supported by the then-ruling coalition, and the Digitization Division of the Chancellery of the Prime Minister declared its willingness to cooperate with organizations appointing observers, which was positively received. Unfortunately, government never contacted observers and none of them took part in the development process. For several months, work on the application proceeded in silence, behind closed doors until beginning of August [Os23], just before the start of the campaign, when then-minister Cieszyński revealed that this "very simple system" was nearing completion. Information about the tests of the system was published on October 5, 2023 [Mi23].

The sign-up and submission workflow is as follows:

1. For proxies only: Representative of electoral committee registers all party proxies (latest on e-day).
2. User (party proxy or observer) downloads mobile app.
3. User authenticates in the app using Polish e-ID - login.gov.pl.
4. User selects a role (proxy in elections, proxy in referendum, domestic observer in elections, international observer) - if user selected proxy, it is compared with the online database (see step 1.).
5. For observers only: Provides certificate from NGO (domestic observer) or NEC (international observer).
6. User attaches the materials.
7. User submits the materials which are being sent immediately to COI.

The application had several shortcomings like distinction in the methods of reporting materials between different user types. There were also obvious deficiencies in basic functionalities. Some problems were resolved at the last moment - the day before the polling, but some persist to this day. Mobile application:

- requires user authentication with a Polish e-ID, even for international observers,
- is exclusively in Polish,
- security assessment has not been performed by any independent organisation,
- doesn't allow logging, loading data and caching them off-line,
- imposes significant limitations regarding the size and quality of the attached materials, therefore instructs users to lower the quality and shorten videos, which can be considered an encouragement to reduce their evidentiary value,
- does not allow attaching materials from the Photo Library on the iOS.

4 Aftermath

High costs for few users made in closed and non-transparent procedure can be a short summary of "Party Prox". The application significantly deviates from expectations. The developers created a process that distorts the electoral realities. These initial observations are quite obvious to someone familiar with electoral procedures, leading to the conclusion that the minister failed to collect required information, the competence in the team creating and evaluating the project, or perhaps consciously accepted subpar execution. That could be mitigated by introducing stakeholders in the process. As a result, one of the authors posed questions about the system to the Minister of Digital Affairs and the agency executing the task on behalf of the minister - the Central IT Center (COI). Unfortunately, this story may serve as an example of a lack of transparency in the informatization of electoral processes in Poland. The ministry and the Central IT Center did not respond for weeks, until the change of the ruling party. COI was the first to respond, indicating that it had spent nearly 2 mln PLN on the project. It also provided information about the project and the use of the application. In the 2023 parliamentary elections, 10,000 party proxies were registered in the database, about 500 people downloaded the application, and ultimately 8 used it for its intended purpose. The materials were not used in any proceedings.

The ministry refused to provide any materials or even photos of documents needed to certify the role of an observer, even though, assuming it wanted to protect the facial images, it could have done so through anonymization. The ministry continued to delay answering the initial questions about the scale of the project and the system until a reminder was sent - increasing the known expenditures to 5.5 million PLN.

Notably, in February 2024, Deputy Minister of Digital Affairs Standerski presented [kr24][łg24] partial information about the costs of development of the system in the media as an example of possible mismanagement. The system, including the mobile application, is active and still used during elections, although it is burdened with numerous errors, both at the conceptual level and in its implementation. It costs 0.5 mln PLN to run it for each elections.

5 Conclusions

Based on the experiences from the 2023 parliamentary elections, the “Party Proxy” system turned out to be a very costly expenditure of public funds. Nearly 5.5 mln PLN were spent, and ultimately only 8 people submitted materials. These materials were not made available to anyone and were not useful in the elections or referendum certification in front of the Supreme Court. The system required prior registration and after the parliamentary elections the database contained data of over 10,000 people who planned to serve as party proxies. In absence of well-defined retention period, it seems that the risks associated with this outweigh the potential benefits of using the “Party Proxy ”.

The way the system was introduced, from both the legislative process and the IT perspective, leaves much to be desired and is certainly not a model to follow in future efforts to introduce i-voting for the Republic of Poland.

References

[kr24] kris and dap: Na rządową aplikację "wydano dwa miliony złotych, skorzystało całe osiem osób". Były minister zabiera głos, TVN24, 2024, URL: https://tvn24.pl/biznes/z-kraju/aplikacja-maz-zaufania-za-2-miliony-zlotych-skorzystalo-8-osob-reakcja-bylego-ministra-cyfryzacji-janusza-cieszynskiego-st7768739, visited on: 07/10/2024.

[łg24] łgo: Miliony na aplikację, z której skorzystało osiem osób (PL), Onet.pl, 2024, URL: https://wiadomosci.onet.pl/kraj/miliony-na-aplikacje-z-ktorej-skorzystalo-osiem-osob-byly-minister-odpiera-zarzuty/kz9xf8v, visited on: 07/10/2024.

[Mi16] Michalak, B.: Conclusions and Recommendations (Not Only) De Lege Ferenda After the Crisis in the Polish Local Elections of 2014. Białostockie Studia Prawnicze 20/A (09), pp. 211–229, 2016.

[Mi23] Ministry of Digital Affairs: Sprawdź, jak działa aplikacja dla mężów zaufania, Portal Gov.pl (PL), 2023, URL: https://www.gov.pl/web/cyfryzacja/sprawdz-jak-dziala-aplikacja-dla-mezow-zaufania, visited on: 07/10/2024.

[Os23] Osiecki, G. and Żółciak, T.: Centralny rejestr gotowy na wybory [WYWIAD] (PL), Dziennik Gazeta Prawna, 2023, URL: https://www.gazetaprawna.pl/wiadomosci/kraj/artykuly/9083748,centralny-rejestr-gotowy-na-wybory-wywiad.html, visited on: 07/10/2024.

[Po23] Political Accountability Foundation and others: Open Letter of Observers and 50 Other Civic Society Organisations Regarding Amendments of Electoral Code in 2023. 2023, URL: http://www.gi.de, visited on: 07/10/2024.

[WR24] Wittenberg, A.; Rutkowska, E.: Standerski: Nie chodzi o uchwalenie prawa w jedną noc [WYWIAD] (PL), Dziennik Gazeta Prawna, 2024, URL: https://www.gazetaprawna.pl/wiadomosci/kraj/artykuly/9438320,standerski-nie-chodzi-o-uchwalenie-prawa-w-jedna-noc-wywiad.html, visited on: 07/10/2024.

When Audits and Recounts Distract from Election Integrity

The 2020 U.S. Presidential Election in Georgia

Philip B. Stark[1]

Abstract: The U.S. state of Georgia was central to efforts to overturn the results of the 2020 Presidential election, including a phone call from then-president Donald Trump to Georgia Secretary of State Brad Raffensperger asking Raffensperger to 'find' 11,780 votes. Raffensperger has maintained that a '100% full-count risk-limiting audit' and a machine recount agreed with the initial machine-count results, which proved that the reported election results were accurate and that 'no votes were flipped.' While there is no evidence that the reported outcome is wrong, neither is there evidence that it is correct: the two machine counts and the manual 'audit' tallies disagree substantially, even about the number of ballots cast. Some ballots in Fulton County, Georgia, were included in the original count at least twice; some were included in the machine recount at least thrice. Audit handcount results for some tally batches were omitted from the reported audit totals: reported audit results do not include all the votes the auditors counted. In short, the two machine counts and the audit were not probative of who won because of poor processes and controls: a lack of secure physical chain of custody, ballot accounting, pollbook reconciliation, and accounting for other election materials such as memory cards. Moreover, most voters used demonstrably untrustworthy ballot-marking devices; as a result, even a perfect handcount or audit would not necessarily reveal who really won. True risk-limiting audits (RLAs) and rigorous recounts can limit the risk that an incorrect electoral outcome will be certified rather than being corrected. But no procedure can limit that risk without a trustworthy record of the vote. And even a properly conducted RLA of some contests in an election does not show that any other contests in that election were decided correctly. The 2020 U.S. Presidential election in Georgia illustrates unrecoverable errors that can render recounts and audits 'security theater' that distract from the more serious problems rather than justifying trust.

Keywords: risk-limiting audit, election recount, evidence-based elections

1 Introduction: The 2020 U.S. Presidential Election in Georgia

Georgia was one of the 'swing states' that determined the outcome of the 2020 U.S. presidential election: its 16 electoral college votes went to Joe Biden. In a well publicized recording of then-president Donald Trump to Georgia Secretary of State Brad Raffensperger, Trump asked Raffensperger to 'find' 11,700 votes.[2]

[1] Department of Statistics, University of California, Berkeley, CA 94720-3860, USA, pbstark@berkeley.edu, https://orcid.org/0000-0002-3771-9604

[2] See, e.g., https://int.nyt.com/data/documenttools/highlights-of-trump-s-call-with-the-georgia-secretary-of-state-1/b67c0d9dbde1a697/full.pdf visited 11 July 2024. Subsequently, in early 2021, Trump-affiliated parties gained improper access to all components of the voting system in Coffee County, Georgia and copied and distributed the codebase and data. See, e.g., https://www.cnn.com/2023/08/13/politics/coffee-county-georgia-voting-system-breach-trump/index.html, https://apnews.com/article/2022-midterm-elections-

doi:10.18420/e-vote-id2024_13

Georgia performed a second machine count and hired VotingWorks to orchestrate a 'risk-limiting audit' of the 2020 presidential contest, including providing software. This paper shows that the audit does not support the election results; that the election, recount, and audit disagree; and that all three were unreliable. Among other issues, some memory cards containing votes were not uploaded in the first machine count. Some ballots were included in the first machine tally at least twice. Some ballots were included in the second machine tally at least three times. And some votes manually tabulated in the audit were not included in the reported audit totals. Moreover, the state of Georgia requires all in-person voters to use Dominion Voting Systems (DVS) ballot marking devices (BMDs) to mark their ballots. These devices are vulnerable to hacking and misconfiguration [Ha23]; voters rarely check BMD printout [Be20, KBW20, HI21]; when voters do check, they are unlikely to notice and report printing errors [Be20, KBW20]; and no feasible amount of pre-election testing, logic and accuracy testing, or election-day monitoring can suffice to show that BMDs misbehavior did not alter the outcome [SX22]. BMD printout is thus not a trustworthy basis for evidence-based elections [SW12, ADS20, AS20], even when voted ballots are curated adequately and proper procedures are followed. While there is no evidence of widespread fraud, the mismanagement of the election, reliance on untrustworthy vote records, lack of physical controls on ballots and other voting materials, lack of sanity checks, and poorly executed procedures make it impossible to know who "really" won.

This story is about Georgia, but the moral is broader: some of the things that can and do go wrong in administering elections result in an untrustworthy vote record. Auditing a poorly run election with an untrustworthy vote record is a distraction from the fact that the vote record is not trustworthy, not a way to justify trust. Auditing cannot restore trustworthiness to a poorly run election; rather, it is a way to "tie a bow around" a *well-run* election to show that whatever might have gone wrong did not alter the electoral outcome.

2 The 2020 audit

Secretary of State Brad Raffensperger claimed, "Georgia's historic first statewide audit reaffirmed that the state's new secure paper ballot voting system accurately counted and reported results. . . . [W]e did a 100 percent risk-limiting audit with a hand recount which proved the accuracy of the count and also proved that the machines were accurately counting it, and that no votes were flipped."[3] VotingWorks Executive Director Ben Adida claimed "Georgia's first statewide audit successfully confirmed the winner of the chosen contest and should give voters increased confidence in the results."[4] Per the official report of the audit, "[t]he audit confirmed the original result of the election, namely that Joe Biden won the

technology-lawsuits-donald-trump-voting-6a1324cc6cf45c95ca086a5c81617b15, https://www.washingtonpost.com/investigations/2022/10/28/coffee-county-election-voting-machines/, all accessed 11 July 2024.

[3] https://sos.ga.gov/news/historic-first-statewide-audit-paper-ballots-upholds-result-presidential-race, accessed 11 July 2024.

[4] Ibid.

Presidential Contest in the State of Georgia. The audit [] provides sufficient evidence that the correct winner was reported."[5]

Secretary Raffensperger has also used the recount and audit in his defense against a lawsuit that seeks to provide all Georgia voters the option to hand-mark paper ballots in person, rather than being compelled to use BMDs (Curling et al. v. Raffensperger et al., Civil Action No. 1:17-CV-2989-AT, U.S. District Court for the Northern District of Georgia, Atlanta Division). Raffensperger has publicly painted the opposing election security experts in this matter—some of the world's top cybersecurity experts—as "stop-the-steal" conspiracy theorists, muddying the waters with false claims about the recount and audit and deliberately conflating "there is strong evidence that the election was poorly run and little evidence that the outcome is correct" with "there is strong evidence that the outcome is wrong and that fraud was committed." Some of the data analyzed below (cast vote records, in particular) were obtained in discovery in *Curling v. Raffensperger*, but most are a matter of public record and can be downloaded from the Georgia Secretary of State's website, from URLs given below.

The so-called 'risk-limiting audit' did not limit the risk of certifying an incorrect electoral outcome for many reasons, starting with its reliance on an untrustworthy record of the votes. The record is untrustworthy because of how it was created (largely BMD printout), curated (a lack of physical accounting for ballots and other materials, lack of pollbook reconciliation, and other elements of a proper canvass), and organized (no "ballot manifest"). The audit *could* have checked the tabulation of the validly cast ballots it found, but it did not check that properly, as proved by documents on the Secretary of State's website.[6]

2.1 Things the audit did not check

The audit did not check whether BMDs correctly printed voters' selections. No audit can check that [ADS20]. (As a consequence, Secretary Raffensperger had no basis to assert that no votes were flipped.) Expert declarations and testimony in *Curling v. Raffensperger* establish that the Dominion BMDs can be hacked, misprogrammed, or misconfigured to print votes that differ from voters' selections as confirmed onscreen or through audio. Logic and accuracy testing cannot establish that BMDs behave correctly in practice [SX22]. Only voters are in a position to check—but few do, and those who do check generally check poorly

[5] Ibid.

[6] https://sos.ga.gov/news/historic-first-statewide-audit-paper-ballots-upholds-result-presidential-race accessed 11 July 2024. Audit data at the urls https://sos.ga.gov/admin/uploads/Georgia%202020%20RLA%20Report.xlsx, https://sos.ga.gov/admin/uploads/county-summary-data.pdf, and https://sos.ga.gov/admin/uploads/audit-report-November-3-2020-General-Election-2020-11-19.csv, linked from that page, are periodically unavailable, producing the message "Sorry, you have been blocked. You are unable to access sos.ga.gov." RLA manual tabulation batch sheets were downloaded from https://sos.ga.gov/admin/uploads/Fulton%20RLA%20Batches.zip on 9 January 2022. Precinct-level results for the original machine tally are at https://results.enr.clarityelections.com//GA//105369/271927/reports/detailxls.zip; for the second machine tally, they are at https://results.enr.clarityelections.com//GA//107231/273078/reports/detailxls.zip, both visited 2 September 2024.

(see citations below). Georgia has no procedures to log, investigate, or report complaints from voters that BMDs altered votes, so it is unknown whether voters observed problems.

- The audit did not check whether every validly cast ballot was included in the tally exactly once. The audit *could not* check whether every validly cast ballot was scanned, because Georgia's rules for ballot accounting, pollbook and voter participation reconciliation, physical chain of custody, etc., do not account for every cast ballot.
- The audit did not check whether the number of participating voters differed from the number of cast ballots.
- The audit did not check whether every memory card used in the election was accounted for, nor whether every memory card containing votes was uploaded to a tabulator. During the audit, it was discovered that some cards had not been uploaded, but there was no comprehensive check to confirm that every card was eventually included exactly once. Below are examples of ballots that were erroneously included in machine counts more than once.
- The audit did not check whether scans were duplicated, deleted, replaced or altered.
- The audit did not check whether QR-encoded votes on BMD printout match the human-readable selections on any ballot.
- The audit did not check whether the voting system correctly interpreted any ballot or BMD printout.
- The audit did not aggregate its own manual tallies correctly, as explained below.

The analysis below focuses on Fulton County (Atlanta), but there is no reason to believe the problems are confined to Fulton; indeed, lapses such as failing to upload memory cards occurred in other counties.

2.2 The audit report omitted some batch tallies

The audit was conducted using "sort and stack": teams sorted batches of ballots (including BMD printout) by the presidential vote, then counted the sorted stacks. Batch tallies were manually entered on paper 'Audit Board Batch Sheets,' (ABBSs). Other workers transcribed the ABBSs into VotingWorks audit software "Arlo" to create a database of tallies; totals were calculated from that database. A spreadsheet of results was produced from Arlo. Every ballot validly cast in Fulton County in the 2020 Presidential Election should be reflected in exactly one ABBS, and data from every ABBS should have been (but was not) entered exactly once into the database from which the audit spreadsheet was exported. The transcription of ABBSs was not observable by the public, but the public could in principle compare posted images of the ABBSs to the posted audit spreadsheet, as described below. (Spoiler alert: they do not match.)

Many ABBSs were not completely filled in. The "Batch Type," signifying the mode of voting (absentee, election day, advance) was often blank, as were many places numbers belonged. The four posted ABBS image files for Fulton County contain a total of 1,927 ABBSs.[7] But the audit spreadsheet contains only 1,916 rows of data for Fulton County. At least eleven ABBSs are entirely missing, not counting possible duplicate entries in the spreadsheet.[8] This sort of "sanity check" is simple to perform, but apparently was not performed by the auditors, VotingWorks, Fulton County, or the Secretary of State.

Table 1 lists 11 ABBSs that do not appear in the audit spreadsheet; the final column indicates which page of which ABBS image file contains the image (for instance, "4 at 162" means page 162 of "Fulton Audit Documents 4_redacted"). The scans of the ABBSs are available at https://figshare.com/s/9819e969a8a6172c25bc (Appendix 1). The fact that the vote data in the last two rows are identical is suspicious, but the corresponding ABBS images are clearly different. Regardless, neither appears in the audit spreadsheet.

	Scanner	Batch	Mode of voting	Trump	Biden	Jorgensen	Write-In	Undervote or blank	Overvote	Image source
1	3	48	absentee	4	93	2	0	0	0	4 at 162
2	2	52	absentee	6	92	0	0	0	0	1 at 1
3	3	12–14	?	12	83	1	0	0	0	4 at 128
4	3	239	?	13	87	0	0	0	0	3 at 177
5	1	80–84	?	118	329	3	2	2	1	3 at 519
6	3	260	absentee	30	66	0	0	0	0	4 at 355
7		AP01A-1	election day	84	62	6	2	1	0	1 at 170
8	3	179–181	absentee	85	224	5	1	2	0	4 at 293
9	2	239	absentee	4	42	0	0	0	0	2 at 153
10	Chastain	12	advance	613	605	24	7	4	0	3 at 351
11	Chastain	114	advance	613	605	24	?	4	0	3 at 270

Tab. 1: Examples of audit board batch sheets (ABBSs, tallies of votes in batches of ballots) that were not entered into the audit results spreadsheet.

There are no data in the audit spreadsheet matching rows 4–11 of table 1. There are data that match rows 1, 2, and 3, but with different batch identifiers.[9] There is no reason to doubt that these are genuinely different batches: some identical counts in different batches are to be expected. Indeed, in the entire audit spreadsheet, there are 16,807 rows that duplicate other ABBS vote counts within the same county, out of a total of 41,881 rows.

Vote totals for Trump, Biden, and Jorgensen derived by summing ABBS entries in the audit spreadsheet match the vote totals in the summary audit result spreadsheet posted by the Secretary of State at the URL https://sos.ga.gov/admin/uploads/Georgia%202020%

[7] Audit subtotals come from the detailed "audit spreadsheet" available at https://sos.ga.gov/admin/uploads/audit-report-November-3-2020-General-Election-2020-11-19.csv accessed 11 July 2024. Images of the Fulton County, GA, RLA manual tabulation batch sheets (ABBSs) were downloaded from https://sos.ga.gov/admin/uploads/Fulton%20RLA%20Batches.zip on 9 January 2022. That file contains five .pdf files, "Fulton Audit Documents 1_redacted.pdf," through "Fulton Audit Documents 4_redacted.pdf," which contain images of ABBSs, and "Fulton Audit Documents 5.pdf" which contains images of "Vote Review Panel Tally Sheets."

[8] However, there is at least one ABBS marked "Dup" (presumably meaning "duplicate") for instance, page 11 of "Fulton Audit Documents 2_redacted.pdf." However, as table 1 shows, at least 11 ABBSs are not accounted for in the audit spreadsheet. Thus, there are presumably duplicated entries in the audit spreadsheet.

[9] The data that match row 1 are identified as "Scanner 3 Ballot [sic] 162" rather than batch 48. The data that match row 2 are identified as "Absentee Scanner 2 Batch 400" rather than batch 52. The data that match row 3 are identified as Absentee Scanner 3 Batch 253 rather than batches 12–14.

20RLA%20Report.xlsx, downloaded on 9 January 2022. The spreadsheet does not list write-ins, undervotes, or overvotes. Both sources show Trump receiving 137,620 votes, Biden receiving 381,179, and Jorgensen receiving 6,494. Thus, the ABBSs that are missing from the audit spreadsheet are also missing from the audit's reported vote totals.

On the assumption that the ABBSs—the original source of the manual tally data entered into the audit spreadsheet—are correct, the omission of that sample of 11 ABBSs deprived Trump of 1,582 votes, Biden of 2,288, and Jorgensen of 65, not to mention write-ins. This sample alone has a total of over 3,900 votes that the audit tabulated but were not included in Fulton County's audit vote totals, compared with a *statewide* margin of less than 12,000 votes.

The original tabulation in Fulton County showed 524,659 votes; the reported audit results showed 525,293, a difference of 634 votes, about 0.12 percent.[10] Accounting for those 11 omitted ABBSs increases the apparent tabulation error from 634 votes to over 4,569 votes or 0.87 percent, far larger than the statewide margin of victory. It is also larger than 0.73 percent, which Secretary Raffensperger claimed was the maximum miscount in any Georgia county.[11]

There is no way to know whether including those 11 ABBSs would make the audit tabulation a complete count in Fulton County: many ballots might remain untabulated, because Georgia's canvass procedures are lax. The proof some Georgia jurisdictions do not keep adequate track of ballots, memory cards, and other election materials is evidenced by the fact that thousands of ballots and scans were "discovered" during the audit.[12] There is no trustworthy inventory of ballots to check the results against.

Georgia Governor Brian P. Kemp pointed out similar flaws in the audit, saying the audit report was "sloppy, inconsistent, and presents questions about what processes were used by Fulton County to arrive at the result."[13] Governor Kemp's letter points out that the audit data include duplicated entries.

[10] Data from https://sos.ga.gov/admin/uploads/Georgia%202020%20RLA%20Report.xlsx, accessed 9 January 2022.

[11] Per Secretary Raffensperger, "[i]n Georgia's recount, the highest error rate in any county recount was 0.73%." https://sos.ga.gov/index.php/elections/2020_general_election_risk-limiting_audit, accessed 9 January 2022.

[12] https://www.cbs46.com/news/floyd-county-election-director-fired-after-audit-reveals-2-600-votes-went-uncounted/article_bbd08d90-2aa2-11eb-9e4d-bf96ac56ad54.html, accessed 10 January 2022. https://www.news4jax.com/news/georgia/2020/11/18/4th-georgia-county-finds-uncounted-votes-as-hand-count-deadline-approaches/, accessed 10 January 2022. https://www.mdjonline.com/elections/cobb-elections-finds-350-uncounted-ballots-during-audit/article_0d93e26e-22bd-11eb-8bce-17067aceee33.html, accessed 10 January 2022. https://www.11alive.com/article/news/politics/elections/fayette-county-election-results-ballots-uncovered-during-audit/85-f79dd838-a15c-4407-80b2-9dfbc2466188, accessed 10 January 2022.

[13] Letter from Brian P. Kemp, Governor, to the Georgia State Election Board, dated 17 November 2021, addressing the work of Mr. Joseph Rossi; Review of Inconsistencies in the Data Supporting the Risk Limiting Audit Report, Office of Governor Brian P. Kemp, 17 November 2021.

3 First Count, Audit, and Recount Differ Substantially

Official precinct-level results for the original tabulation were downloaded from https://results.enr.clarityelections.com//GA/Fulton/105430/271723/reports/detailxls.zip and for the recount from https://results.enr.clarityelections.com//GA/Fulton/107292/275183/reports/detailxls.zip to examine the results in precinct RW01, the precinct in which the lead plaintiff in *Curling v. Raffensperger* votes.

Table 2 shows the counts of election-day votes in precinct RW01 for the three presidential candidates, according to the original machine count, the machine recount, and the "audit," and vote-by-mail and advance votes for the original election and the recount. (The audit did not report precinct-level results for vote-by-mail or advance voting.)

Count	Election Day			Advance			Absentee by Mail			Provisional		
	Trump	Biden	Jorgensen	Trump	Biden	Jorgensen	Trump	Biden	Jorgensen	Trump	Biden	Jorgensen
Original	193	88	11	1455	1003	23	619	833	15	9	4	1
Recount	162	73	9	1487	1015	25	619	809	15	5	3	1
Audit	243	88	11									

Tab. 2: Election day, advance, absentee, and provisional vote tallies for Fulton County, GA, precinct RW01 in the 2020 U.S. Presidential election

There are large, unexplained differences among these results.[14] Secretary Raffensperger attributed all differences between the audit and the original count to human counting error, citing a 2012 study that found hand-count error rates as high as 2 percent.[15] While human error presumably accounts for *some* of the difference, there is no evidence that it accounts for most of the difference, much less the entire difference, as Secretary Raffensperger claimed.[16]

The original count and audit agree with each other (but not with the recount) regarding the number of election-day votes for Biden and Jorgensen. The audit found 50 more election-day votes for Trump than the original tally and 81 more than the machine recount found: a difference of almost 50 percent. These differences have not been investigated and are unexplained. A hypothesized error rate of 2 percent in hand counts does not suffice.

The differences might result from discrepancies between the QR-encoded votes and the human-readable votes on BMD printout and/or from misconfiguration, bugs, or malware on the scanners or tabulators. As discussed above, the audit checked none of these things. Possible machine error should have been investigated, rather than assumed not to exist.

[14] There appears to be some cancellation of error, but the hand count kept ballots cast in different ways separated (advance in-person, absentee by mail, and election day). It is not clear how misclassification of the mode of voting would affect one candidate's totals much more than the other candidates. Regardless, these discrepancies are large and should be investigated, including inspecting the physical ballots.

[15] https://sos.ga.gov/index.php/elections/historic_first_statewide_audit_of_paper_ballots_upholds_result_of_presidential_race, accessed 10 January 2022.

[16] Moreover, RLAs treat the hand count as the correct count: the hand counts should be conducted with adequate care to ensure they are accurate, which typically requires different procedures from those used in initial manual tallies.

The hand count could easily be more accurate than the machine count. Indeed, it is well known that careful hand counts of hand-marked paper ballots are often more accurate than machine counts, in part because human readers can interpret faint, improper, and ambiguous marks better than machines can, even when the machines are working properly, as studies of "residual votes" and statewide recounts show [AR04, An18, AS05, ABS13, AAH13, Ca05].[17]

The scanner settings Georgia uses for its Dominion scanners (low resolution, black-and-white) can cause voters' selections not to appear at all in the images, selections that are obvious to human readers looking at the actual ballots.[18] Manual tallies generally find more valid votes than machine tallies. Hand-count error rates are known to depend on many factors, including ballot design, the method for hand counting ("sort-and-stack" versus "read-and-mark"), and the size of counting teams. They presumably also depend on whether there are additional quality control measures in place, such as checking sorted piles of ballots to ensure that each pile has votes for just one candidate.

The study [GBG12] cited by Secretary Raffensperger to support his claim is a laboratory study with 108 subjects and 120 ballots, each containing 27 contests with two candidates. It used three kinds of "ballots": printout from two kinds of DRE (direct-recording electronic) voting system and an optical scan ballot. The highest error rates were for thermal printout from DREs, which does not resemble Georgia's BMD printout nor Georgia's hand-marked paper ballots. The method with the *highest* error rate was the "sort-and-stack" tally method that Georgia used in its audit. The study did not observe hand tabulation in a real election, nor did it involve BMD summary printout.

The differences between the original count and the machine recount are large and unexplained; for instance, the difference in the counts of Biden's Absentee votes is almost 3 percent. It is now impossible to know what went wrong, nor whether the differences are primarily attributable to malware, bugs, misconfiguration, or human error.

3.1 The two machine counts in Fulton County

This section assesses the internal consistency of the two machine counts (the original machine count and the machine recount) in Fulton County using data from the election management system (EMS) including cast vote records (CVRs), scanned images of ballots, and BMD printout, and other files made available to the plaintiffs in Curling et al. v.

[17] Whether hand counts are more accurate than machine counts depends on many variables. The scrutiny and care involved in recounts and manual audits are generally higher than they are in initial hand counts. For instance, [An18] find that *initial* machine counts were often more accurate than *initial* hand counts—by using careful handcounts from statewide recounts as the touchstone for the correct counts.

[18] See, e.g., Judge Amy Totenberg's Opinion and Order of 11 October 2020, in Curling et al. v. Raffensperger, 1:17-CV-2989-AT, at 4, 30, 95, 101, 103, 114–135.

Raffensperger et al. To confirm that the EMS data were the correct data, tallies were calculated and compared to the official results for Fulton County; they matched:[19]

Candidate	1st machine count	2nd machine count
Donald J. Trump	137,240	137,247
Joseph R. Biden	381,144	380,212
Jo Jorgensen	6,275	6,320

Tab. 3: Data used to verify that the EMS download matched the official results in Fulton County, GA.

The number of cast vote records (the voting system's record of the votes on each ballot or BMD printout card, from which the system tabulates results) in the two machine counts in Fulton County were rather different: 528,776 in the first count and 527,925 in the second count, a difference of 851. Fulton County has not explained this discrepancy.

The number of cast vote records in the two machine counts should be equal. Differences might occur if (i) some ballots or BMD printout cards were misplaced or found between the two machine counts, so a different number pieces of paper was scanned in the two machine counts; (ii) malware, bugs, misconfiguration, or a bad actor added, deleted, or altered records in the election management system in one or both machine counts; (iii) Fulton County did not scan every validly cast ballot or BMD printout card exactly once in each machine count; (iv) some scans were omitted or improperly included in one or both counts. Compelling evidence that (ii) or (iii) is true is presented below, but all four could be true simultaneously.

Fulton County did not produce an image file for every cast vote record. For the first machine count, production included images of ballots or BMD printout cards for only 168,726 of the 528,776 cast vote records: 376,863 image files are missing. For the second machine count, Fulton County's production included images of ballots or BMD printout cards for 510,073 of the 527,925 cast vote records: 17,852 image files are missing.

Entire batches of images are missing from Fulton County's production; for example, images from Scanner 801 batch 117 and Scanner 801 batch 118 are referred to in the cast vote records for the second machine count but the images were not among the electronic records. Without additional information it is impossible to determine whether the missing images are missing because of human error or malfeasance, programming errors (bugs), or malware in Fulton County's election management system (EMS)—possibilities that are not mutually exclusive.

The extant images nonetheless prove that Fulton County's election results included many votes more than once in the reported tabulations. The full extent of multiple-counting cannot be determined without additional information, but there is evidence that it added thousands of bogus votes to the reported machine-count results. That is, thousands of Fulton County

[19] First machine count results: https://results.enr.clarityelections.com/GA/Fulton/105430/web.264614/#/summary (visited 11 July 2024) Second machine count results: https://results.enr.clarityelections.com/GA/107231/web.264614/#/detail/5000?county=Fulton (visited 11 July 2024)

voters' votes were included in the reported totals more than once. It is not possible to determine conclusively whether any voter's votes were omitted from the reported totals.

Repeatedly scanning the same piece of paper generally does not produce images that are bitwise identical, because of variations in the alignment of the paper, illumination within the scanner, dirt on scanner lenses, etc. Similarly, a single scan can be altered digitally to produce multiple images that look similar but are not bitwise identical.

Small variations in voters' marks (e.g., not filling an oval completely or straying outside the oval) on hand-marked paper ballots generally make it possible to tell whether two separate scans of hand-marked paper ballots that contain the same votes are scans of the same physical ballot.

It is not generally possible to tell whether two 200dpi black-and-white scans of BMD printout cards are scans of the same piece of paper simply by looking at those two scans, because BMD printout cards containing the same votes may be indistinguishable at low resolution in black-and-white.[20] However, if both scans contain a rare write-in name or rare combination of write-in names, that is evidence of a duplicate. Similarly, if a series of votes is repeated in in the same order (or reverse order) in different scan batches of BMD printout, that is also evidence that they are repeated images of the same collection of paper. If the duplicated (or reversed) vote sequences are long and include rare write-in names, the evidence that they are scans of the same physical pieces of paper is compelling.

There are at least 12 hand-marked ballots from Fulton County precinct RW01 that were scanned twice in the first machine count (the original election). Fourteen pairs of duplicate images are listed in table 4 and are available at the url https://figshare.com/s/9819e969a8a6172c25bc (Appendix 2). The format of the numbers is <scanner number>_<batch number>_<image number>. At least three BMD cards from precinct RW01 appear to have been scanned twice in the machine recount in RW01, based on the votes and the order in which they were scanned in two batches. In particular, Scanner 801, batches 43 and 44—both comprising scans of advance in-person BMD printout cards—start with images of 214 BMD cards that have the same sets of votes in the same order in both batches. The two batches were scanned within about five minutes of each other, according to the timestamps in the images. Many of the images show write-in votes[21] or votes for third-party candidates, further evidence that the match was not coincidence. Visual inspection of all 214 pairs and confirmed that they match: those BMD cards were scanned twice in the machine recount. The other 211 (214–3=211) duplicated scans are of BMD cards from other precincts in Fulton County.

[20] Differences in the monochrome threshold or scanner maintenance might create discernable differences. A sufficiently high-resolution scan might make it possible to identify differences in the arrangement of the paper fibers [Cl09].

[21] Write-ins included votes for "Anyone," "XXX," "Willie Nelson," and "Alexander Hamilton," as well as write-in votes for "Donald Trump" for District Attorney, Clerk of the Superior Court, Tax Commissioner, Sheriff, Solicitor General, and Surveyor.

pair	Image A	Image B
1	05162_00234_000096	05162_00235_000057
2	05162_00234_000093	05162_00235_000054
3	05162_00234_000074	05162_00235_000036
4	05162_00234_000072	05162_00235_000034
5	05162_00234_000068	05162_00235_000030
6	05162_00234_000069	05162_00235_000031
7	05162_00234_000054	05162_00235_000014
8	05162_00234_000031	05162_00235_000090
9	05162_00234_000026	05162_00235_000085
10	05162_00234_000017	05162_00235_000076
11	05162_00234_000013	05162_00235_000072
12	05162_00234_000014	05162_00235_000073
13	05162_00234_000003	05162_00235_000062
14	05162_00234_000001	05162_00235_000060

Tab. 4: Images that were included in the original machine count in Fulton County at least twice. Images are posted at https://figshare.com/s/9819e969a8a6172c25bc (Appendix 2).

There is also one hand-marked paper ballot that was scanned twice in RW01 in the machine recount, and at least seven hand-marked paper ballots that were scanned thrice in RW01 in the machine recount. Twenty-nine images seem to represent only 11 distinct pieces of paper, even though they contributed 29 votes to some contests, including the presidential contest. The sets of images are available at the url https://figshare.com/s/9819e969a8a6172c25bc (Appendix 3). Table 5 lists the pairs and triples.

Multiple	Image A	Image B	Image C
1	00801_00044_000168	00801_00043_000168	
2	00801_00044_000083	00801_00043_000083	
3	00801_00044_000042	00801_00043_000042	
4	05160_00074_000023	05160_00067_000008	
5	00794_00017_000024	00791_00026_000091	00791_00019_000010
6	00794_00017_000029	00791_00026_000086	00791_00019_000015
7	00794_00018_000001	00791_00026_000009	00791_00019_000092
8	00794_00018_000011	00791_00026_000019	00791_00019_000082
9	00794_00019_000002	00791_00026_000079	00791_00019_000022
10	00794_00019_000005	00791_00026_000076	00791_00019_000025
11	00794_00019_000006	00791_00026_000075	00791_00019_000026

Tab. 5: Images that were (erroneously) included in the machine recount at least three times. Images are posted at https://figshare.com/s/9819e969a8a6172c25bc (Appendix 3).

To confirm that the duplicate and triplicate images were included in the reported vote tabulation, the cast-vote records (CVRs) produced by Fulton County for each image identifier among the duplicates and triplicates of images of RW01 ballots and BMD printout cards were searched electronically. All 24 from the original count and all 29 from the machine recount were among the CVRs. Therefore, the duplicate and triplicate votes were included in the reported machine tabulations, since the vote totals derived from the CVRs agree with the reported vote totals, as mentioned above.

For Fulton County as a whole, plaintiffs in *Curling v. Raffensperger* identified images of 2,871 ballots and BMD printout cards that they claim were counted two or three times in the second machine count. Some were identified by visual inspection of the images; others were inferred to be duplicates because a sequence of cast vote records was identical (or reversed) for long portions of two scan batches. I confirmed that 214 of the purported duplicate scans of BMD cards were indeed duplicates. This list of 2,871 is a sample from a larger list of images of ballots and BMD printout cards that plaintiffs assert were included in the tabulation twice or more. All 6,118 images in question were referenced in CVRs in the second machine count, so all contributed to the tabulation.

Nine hundred sixteen (916) of the 2,871 sets of images were images of hand-marked paper ballots. In a random sample of 100 of those 916, I verified visually that 46 contained triplicate images. I confirmed the determination for 98 of the 100 sets. I disagreed about one set, and was unable to verify one set. Treating this conservatively as 98 agreements in 100 random checks yields a 95 percent lower confidence bound that at least 891 of the 916 claimed multiples are genuine multiples.

These observations make it clear that in the original count and in the machine recount, Fulton County did not keep track of which ballots and BMD cards had been scanned and which had not. It is also possible that the electronic records were altered accidentally or intentionally, or that some memory cards were not uploaded or uploaded more than once. The electronic records of the election are not intact. This is a surprising gap: the most basic election safeguard is to check whether the number of voters who participated is equal to the number of ballots and BMD printout cards that were cast and to the number that were tabulated. Moreover, one might reasonably expect all electronic election materials to be backed up onsite and offsite, at least for the U.S. federally mandated retention period of twenty-two months, so the loss of hundreds of thousands of image files from the first machine count and of nearly 18,000 images from the second machine count is hard to fathom.

Fulton County would have noticed these errors if it had kept track of ballots and BMD printout cards and checked the total number against the number reported in the electronic tabulation. It seems that Fulton County did not know how many ballots and BMD printout cards were cast in the election, how many voters cast votes, or how many pieces of paper were scanned—nor how those numbers compare to each other. Absent basic ballot accounting, pollbook reconciliation, and counting of electronic records, it is unsurprising that the two machine tallies differ so much. The U.S. Election Assistance Commission has published best practices for chain of custody.[22]

[22] https://www.eac.gov/sites/default/files/bestpractices/Chain_of_Custody_Best_Practices.pdf accessed 11 July 2024.

Fulton County's lax curation and processing of cast ballots, BMD printout, and electronic records make a true risk-limiting audit impossible because even a perfect tabulation of the votes from the available paper might not show who really won. Voters have good reason to believe that some votes counted more than others, since some votes were included twice or thrice in the totals. There is no way to know how many votes were omitted from the tabulation, absent access to the physical ballots and BMD printout and evidence that the chain of custody is intact. It is impossible to determine whether malware, bugs, misconfiguration, or malfeasance disenfranchised voters or altered the election results.

The audit planning, process, and controls did not detect the double and triple counting. Even if Fulton County did not rely on ballot-marking devices for all in-person voters, the lack of basic accounting controls makes it impossible to determine who really won, even by a perfect hand count of the votes: the record of the vote could easily be incomplete or adulterated. There is no reason to believe that problems of the kinds described above are limited to Fulton County.

4 Summary

An accurate recount of the votes in a trustworthy record can determine the true winners of an election, and a rigorous audit can provide confidence that a well-run election found the true winner(s). But neither a recount nor an audit can compensate for using untrustworthy technology to record votes, for instance, because the election was run poorly and had inadequate physical security controls; in such circumstances, recounts and audits distract attention from the real problems rather than justifying confidence. Absent a trustworthy record of the votes, no procedure can provide affirmative evidence that the reported winner(s) really won.

Georgia lacks such a record for many reasons, including the heavy reliance on BMDs; lack of physical accounting of voted ballots, memory cards, and other election materials; lack of pollbook and voter participation reconciliation; lack of rigorous chain of custody; etc. To provide reasonable assurance that every validly cast vote is counted—accurately—requires systematic improvements:

1. Every voter should have the opportunity to mark a ballot by hand, whether voting in person in advance, in person on election day, or absentee by mail.
2. Reduce the use of ballot-marking devices to a minimum:[23]

[23] Hand-marked ballots should be offered to in-person voters by default, with access to a BMD available upon request. BMDs or other accessible means of marking a ballot should be set up in advance, so that it is available if and when a voter requests to use it. BMD printout should resemble hand-marked paper ballots to the extent possible, to preserve voter privacy: they should the same paper stock, have the same format as hand-marked paper ballots, and the marks should be printed to resemble hand-made marks, e.g., by digitizing actual hand-made marks.

- BMDs do not necessarily print voters' selections accurately. They can be hacked or misconfigured [Ha23, ADS20].
- A growing body of empirical work shows that few voters check the BMD printout, and those who do rarely catch and report errors [Be20, KBW20, HI21].
- There is no way for a voter to prove to an election official or anyone else that a BMD malfunctioned. Hence, there is no way to ensure that malfunctioning devices are removed from service if voters notice BMDs misbehaving. And if a device is caught misbehaving, there is no way to reconstruct the correct election outcome [ADS20].
- There is no way to test BMDs adequately prior to, during, or after an election to establish whether they altered votes, even if they altered enough votes to change electoral outcomes [ADS20, SX22].

3. Implement better procedures and checks on chain of custody of election materials, especially voted ballots. Georgia currently cannot determine whether every validly cast ballot was included in the reported results exactly once, whether there was electronic or physical "ballot-box stuffing," or whether votes were altered.[24]
4. Implement better protocols for using and checking physical security seals on ballots and voting equipment—and check whether those protocols were followed. Require routine scrutiny of custody logs and surveillance video, and other related security measures.
5. Perform internal consistency checks as part of the canvass, including, e.g.:
   a) Verify that the number of ballots sent to each polling location (and blank paper stock for ballot-marking devices and ballot-on-demand printers) equals the number returned voted, spoiled, or unvoted. This check should be physical, based on manual inventories, not on reports from the voting system.

   b) Check pollbooks and other voter participation records against the number of voted ballots received, including whether the appropriate number of ballots of each "style" were received.

   c) Check whether the number of electronic vote records (images and CVRs) agrees with the physical inventory of ballots of each style.

A genuine risk-limiting audit requires a demonstrably trustworthy record of voter intent.

[24] This is evidenced by the fact that the 2020 audit found thousands of untabulated ballots. Per the Secretary of State's office, "[t]he audit process also led to counties catching making mistakes they made in their original count by not uploading all memory cards." https://sos.ga.gov/news/historic-first-statewide-audit-paper-ballots-upholds-result-presidential-race accessed 11 July 2024. Because physical accounting for election materials was lacking, there is no way to know how many more votes validly cast in that election were not included in any of the reported tallies. Moreover, the lax recordkeeping evidently resulted in scanning the same batches of ballots more than once. Similarly, some ABBSs were presumably entered more than once, and as shown above, some were not entered at all.

Georgia's vote records are untrustworthy for many reasons, starting with the heavy use of ballot-marking devices, which do not produce a trustworthy record of the vote [ADS20, Ha23] no matter how much logic and accuracy testing or election-day monitoring there is [SX22]. The lack of a trustworthy record is exacerbated in Georgia by the lack of ballot accounting, pollbook reconciliation, and other elements of a good canvass. There are also problems with Georgia's verification of voter eligibility and voter participation record. But even if every voter used a hand-marked paper ballot and there were no issues determining voter eligibility, Georgia does not keep track of election materials adequately through physical inventories, custody logs, and other means.

The foundation for a risk-limiting audit is a *ballot manifest*, a physical inventory of the validly cast paper ballots detailing how they are stored: the number of containers, their identifiers, and the number of cards in each. It must be derived without reliance on the voting system or the audit is trusting the voting system to check itself. For example, if some cards were never scanned or some scans were not uploaded (as discovered during the 2020 "audit"), they will be missing from any manifest derived from the voting system. Absent a physical inventory, it is impossible to account for votes reliably and impossible to limit the risk that an incorrect electoral outcome will be certified, even with a careful manual recount or rigorous audit: recounting or applying risk-limiting audit procedures to an untrustworthy collection of ballots is "security theater."

Like many states, Georgia audits only a small number of contests in each election. Even a properly conducted RLA using a demonstrably trustworthy paper trail confirms only the contest or contests that were audited—and no other contests—although election officials sometimes claim otherwise.[25]

Acknowledgments. This work was supported in part by NSF Grant SaTC–2228884.

Bibliography

[AAH13] Alvarez, R.M.; Atkeson, L.R.; Hall, T.E.: Evaluating Elections: A Handbook of Methods and Standards. Cambridge University Press, NY, 2013.

[ABS13] Alvarez, R.M.; Beckett, D.; Stewart, C.: Voting Technology, Vote-by-Mail, and Residual Votes in California, 1990–2010. Political Research Quarterly, 66(3):658–670, 2013.

[ADS20] Appel, A.W.; DeMillo, R.; Stark, P.B.: Ballot-marking devices cannot assure the will of the voters. Election Law Journal: Rules, Politics, and Policy, 19(3):432–450, 2020. Preprint: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3375755.

[An18] Ansolabehere, S.; Burden, B.C.; Mayer, K.R.; III, C. Stewart: Learning from Recounts. Election Law Journal, 17(2):100–116, 2018.

[25] For example, the State of Colorado currently conducts an RLA of two contests in each jurisdiction in each election, but the Secretary of State's website says, "Colorado residents can be confident that official election results reflect the will of voters because we conduct a statewide bi-partisan audit after every election to ensure the integrity of the results." https://www.coloradosos.gov/pubs/elections/auditCenter.html accessed 24 July 2024.

[AR04] Ansolabehere, S.; Reeves, A.: Using Recounts to Measure the Accuracy of Vote Tabulations: Evidence from New Hampshire Elections 1946–2002. In (Alvarez, R.M.; Hall, L.R. Atkesonand T.E., eds): Confirming Elections: Creating Confidence and Integrity Through Election Auditing. Palgrave MacMillan, NY, 2004.

[AS05] Ansolabehere, S.; Stewart, C.: Residual Votes Attributable to Technology. The Journal of Politics, 67(2):365–389, 2005.

[AS20] Appel, A.W.; Stark, P.B.: Evidence-Based Elections: Create a Meaningful Paper Trail, Then Audit. Georgetown Law Technology Review, 4.2:523–541, 2020. https://georgetownlawtechreview.org/wp-content/uploads/2020/07/4.2-p523-541-Appel-Stark.pdf.

[Be20] 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: 41st IEEE Symposium on Security and Privacy. IEEE, pp. 679–694, 2020.

[Ca05] Carrier, M.A.: Vote Counting, Technology, and Unintended Consequences. St. John's Law Review, 79(3):645–687, 2005.

[Cl09] Clarkson, W.; Weyrich, T.; Finkelstein, A.; Heninger, N.; Halderman, J. A.; Felten, E. W.: Fingerprinting Blank Paper Using Commodity Scanners. 2009 30th IEEE Symposium on Security and Privacy, pp. 301–314, 2009.

[GBG12] 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, pp. 36–51, 2012.

[Ha23] Halderman, J.A.: Security Analysis of Georgia's ImageCast X Ballot Marking Devices. https://storage.courtlistener.com/recap/gov.uscourts.gand.240678/gov.uscourts.gand.240678.1681.0.pdf, 2023. Expert report submitted in Curling v. Raffensperger, Civil Action No. 1:17-CV-2989-AT U.S. District Court for the Northern District of Georgia, Atlanta Division.

[HI21] Haynes, A.A.; III, M.V. Hood: Georgia Voter Verification Study. https://s3.documentcloud.org/documents/21017815/gvvs-report-11.pdf, 2021. last visited 31 October 2021.

[KBW20] 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. Technical report, 2020.

[SW12] Stark, P.B.; Wagner, D.A.: Evidence-Based Elections. IEEE Security and Privacy, 10:33–41, 2012.

[SX22] Stark, P.B.; Xie, R.: They may look and look, yet not see: BMDs cannot be tested adequately. In (Krimmer, R.; Volkamer, M.; Duenas-Cid, D.; Kulyk, O.; Rønne, P.; Solvak, M.; Germann, M., eds): Proceedings of E-Vote-ID 2022, Lecture Notes in Computer Science, volume 13553, pp. 122–138. Springer-Nature, Cham, 2022.

Improving the Observation of ICT in Elections: Widening the Methodological Scope

Vladimir Misev[1], Dr. Beata Martin-Rozumilowicz[2], and Liisa Past[3]

Abstract: A growing number of countries use ICT-enabled solutions for their elections. Some of the most discussed are those related to voting, counting and tabulation of election results. Many other aspects of the election process, however, from voter and candidate registration to election campaigning, political and campaign finance as well as results management and publication systems have become ICT-based. Unfortunately, they are also increasingly becoming the targets of a number of malicious actors intending to disrupt electoral processes and undermine public trust.

The authors showcase the Organization for Security and Co-operation in Europe‘s Office for Democratic Institutions and Human Rights (OSCE/ODIHR) recently updated methodology, which they worked to update, including increased attention to cybersecurity in elections and auxiliary systems. They also apply this new framework to the specific case of Estonia, which has substantial experience and comparatively long history of conducting online elections supported by various ICT-based ancillary systems and as such has been at the forefront of facing cybersecurity challenges across its governance systems. This advancement should help observers to better assess whether states are conducting elections that are genuine and democratic in order to better protect integrity and potentially strengthen public confidence.

Keywords: ICT, elections, cybersecurity, elections ancillary systems, OSCE/ODIHR.

1 Introduction

A growing number of countries use ICT-enabled solutions for their elections. Some of the most discussed are those related to voting, counting and tabulation of election results. Many other aspects of the election process, however, from voter and candidate registration to election campaigning, political and campaign finance as well as results management and publication systems have become ICT-based. Unfortunately, they are also increasingly becoming the targets of malicious actors intending to disrupt electoral processes and undermine public trust.

Adversaries’ objectives are often not aimed at compromising any system, but rather

[1] International Election Expert, North Macedonia, vladimir.misev@gmail.com
[2] International Election Expert, United Kingdom, rozumil@hotmail.com
[3] International Election Expert, Estonia, liisa.past@gmail.com

doi:10.18420/e-vote-id2024_14

sowing confusion and distrust, which in turn de-legitimizes democratic processes. Thus, attackers tend to be reactive and opportunistic. Attackers also tend to look for the ‘lowest hanging (or most impactful) fruit’. Therefore, election administrators and those responsible for the election technology (including technology partners and vendors) must manage risks comprehensively. Thus, the concept of cybersecurity, which commonly refers to “how electronically processed information can be secured against disruption, disablement, destruction or malicious control, thereby protecting its confidentiality, integrity, and availability” is becoming central to protecting electoral integrity[4] and the information security of systems, machines, and data.

While a number of initiatives, guidelines, and measures have been adopted and implemented globally, different cybersecurity threats and incidents have been reported during election periods. These often include malware, phishing, ransomware, hacking, exploiting hidden technology gaps, social engineering and distributed denial-of-service (DDoS) attacks. This short paper will explore the recent methodological response from the Organization of Security and Co-operation in Europe’s Office for Democratic Institutions and Human Rights (OSCE/ODIHR), which has recently revamped and updated its *Handbook for the Observation of Information and Communication Technologies (ICT) in Elections*.[5] The electoral systems across the OSCE are increasingly under threat of various hybrid attacks, often combining tactics and different types of attacks that necessitates specific and comprehensive regional responses. Although there is insufficient space for cross-national analysis in this short paper, it is intended to be the focus of future research.

Although elections, their administration, and use of ICT tools can vary considerably across OSCE participating States, global and international standards require them to conduct genuine and democratic elections. All elements of elections need to maintain a high level of public confidence while fulfilling strict legal criteria and have strict timeframes, as mandated by applicable legislation. This means that any ICT elements have to meet high availability requirements during elections (little or no downtime is acceptable). High confidentiality requirements on technology are also necessitated by the privacy and vote secrecy requirements on elections and high integrity is needed to maintain confidence in elections. Therefore, the cybersecurity requirements on elections are considerably similar regardless of the specifics of use of ICT.

The authors present an overview of the updated methodology, including its increased attention to cybersecurity in elections and auxiliary systems. They also apply this new framework to the specific case of Estonia, which has substantial experience and comparatively long history of conducting online elections supported by various ICT-based ancillary systems and as such has been at the forefront of facing cybersecurity challenges across its governance systems. It specifically looks at the cooperation

[4] See NIST Glossary https://csrc.nist.gov/glossary/term/cybersecurity

[5] See Handbook for the Observation of Information and Communication Technologies (ICT) in Elections, OSCE/ODIHR, https://www.osce.org/files/f/documents/c/9/558318_0.pdf, accessed on 6 July 2024.

between the multitude of national actors that have a mandate and capacities to operate in this space. The paper will look at lessons learned from Estonia in terms of its various initiatives dedicated to combating cyberthreats. Lastly, while states have positive obligations to protect the integrity of elections, many other stakeholders including citizen observer organizations should be equipped with sufficient tools to buttress states in conducting elections in line with their obligations and democratic standards.

2 New Elements of the Updated OSCE/ODIHR ICT Handbook

The OSCE/ODIHR first published its Handbook for the Observation of New Voting Technologies in 2013, at a time when many OSCE participating States were exploring the idea of introducing electronic voting (mainly in controlled environments, i.e., polling stations). In the intervening decade, many countries have moved away from the idea of machine voting, due to various risks and limitations that have emerged. At the same time, states have increasingly explored widening the use of technology in other spheres of elections ('ancillary systems').

Meanwhile, attention to cybersecurity risks has increased across the board, especially after a series of high-profile cyberattacks post 2014 across many OSCE countries. This necessitated a strengthening the methodological framework for OSCE/ODIHR observers to better capture these important elements of the election process. With this in mind, expert authors were commissioned to amplify the analysis and attention given to aspects such as cybersecurity, ancillary systems, and communications technologies in their observation and assessment of electoral process across the 57 OSCE participating States.

The resulting re-vamped handbook recognizes that key principles for genuine and democratic elections remain the same, irrespective of the technology used during election processes. Given the increased potential for abuse and create massive public discord in contexts where elections substantially rely on ICT-based solutions it applies these principles more rigorously and methodically in an ICT setting and discusses wider standards and requirements for protecting elections integrity, including on cybersecurity. While the new handbook notes that there are no global or OSCE-wide election-specific cybersecurity standards it lays out the basic concepts of ICT in elections and how these new challenges relate to these key principles for democratic elections.

Key amongst these are the updates to the OSCE/ODIHR methodology for observing and assessing the use of ICT in elections, especially in the voting and counting, as relates to the overall standards and principles that apply. This includes defining the different types of technologies currently applied throughout OSCE participating States, issues of verifiability, and the risks and benefits that such technologies bring to elections.

The handbook goes in detail about the international standards, principles, and good practices that pertain to elections and how they apply to the observation of ICT in elections. This includes extensive information for election observation missions of various types and develops questionnaires and checklists for all levels of observer to better address these issues. The new handbook also discusses in detail new aspects that require increased attention in observation. It also defines the ancillary systems that apply technology solutions most often. These include areas such as electronic registration, verification of voters / candidates, and results management systems.

Specific sections are dedicated to the principles of cybersecurity. New chapters are devoted to these issues to provide election observers, both ICT experts and other team members, with a good grounding and understanding of the issues at hand and what elements they can observe and report upon during their deployment. This is important to a better understanding of how ICT in elections fits into an overall assessment of elections according to international standards.

Given the risks that have come to light regarding bad actors and potential for abuse, understanding and better observing how technology dovetails with credible elections is a key piece of the observation puzzle. A better understanding and transparency of these issues is key for all electoral stakeholders. In this space, as in traditional electoral aspects, the standards and principles that hold for elections to be genuine and democratic are even more needed to protect integrity and strengthen public confidence.

3 The Case of Estonia and Cybersecurity in Practice

From a practitioners' point of view, the inclusion of cybersecurity as a separate chapter in the updated handbook highlights the need for risk management and appropriate proportionate cybersecurity measures in elections. It also addresses monitoring or assessing elections and of their components. Most importantly, it emphasizes the comprehensive scope of such an undertaking; "the EOM [election observation mission] needs to look across the whole process of electronic information and data storage, processing, transmission, confidentiality, integrity and availability."[6] The handbook then maps possible cybersecurity and information security activities through the election lifecycle.

Secondly, the importance of responsibility for technology and a multi-stakeholder approach are laid out; "given the complexity of the issue, multiple state agencies are often responsible for technology security. It is crucial for the NAM, and later for the EOM, to properly map and assess the delineation of roles and responsibilities of different agencies involved in the electoral process. The ICT Analyst needs to explore how the technology is supplied, implemented and secured through a holistic cyber-risk mitigation framework."[7] This outlines the starting point of successful implementations of election technology as well as the observation or assessment of it.

[6] Ibid, p.89

[7] Ibid, p.90

To check an election manager's cybersecurity stance against real world threats, the handbook lists the most common types of cyberattack directed at electoral processes and maps them to stages of elections.[8] As such, it could serve as a starting point for threat-driven risk assessment. Similarly, for an observer, this can be useful to assess the consideration and readiness to prevent, detect, and mitigate such attacks and their possible impact on elections. All in all, the handbook describes a risk and asset driven security approach with possible cybersecurity questions similarly pertaining to holistic IT management and comprehensive risk management.[9]

To understand what that means, let us look further at how the handbook might be used in Estonia. The country has been internet voting since 2005 and up to every second vote is cast online.[10] The online voting system, mimicking double-envelope postal systems electronically, is one of a plethora of modes of voting. Elections, including online voting, rely heavily on a lively ecosystem of digital services. For example, the voter list is imported from the population registry and online voting is facilitated by an e-ID. This, together with election-specific ICT solutions (the election information system, the online voting application) forms the ICT of elections that is implemented as well as assessed and monitored.

A taskforce of stakeholders includes the Elections Office, the Government Office, and the Information System Authority (both as the technology partner for elections as well as the cyber oversight of the .ee domain), as well as the technology vendors for online voting and the elections information system. This taskforce is an important element for e-observers to assess how the different parties come together. CERT.EE and the help desk of the Information System Authority have stood at increased readiness during elections for the last few electoral cycles. An observer would examine these formats and arrangements to assess cooperation, management and incident response readiness as well as support for technology.

The handbook's advice given on security of Electronic Voter Registration and Verification Systems applies insofar as they are deployed during elections. As Estonian elections rely heavily on ancillary ICT systems, such as the population registry for voter lists and government backed secure digital identity for authentication and authorization of online voters (identifying the internet voter and using the digital signature to create the "digital outer envelope" for their vote), an observation mission would need to carefully determine their scope and integrity. Those systems support thousands of digital services (government and private sector) and are subject to strict legal, regulatory and cybersecurity security requirements (including information security standards). This is why scoping is particularly important in observation or assessment missions, to offer a comprehensive perspective and parity with observing non-digital solutions.

The handbook includes international standards, principles, and good practices on

[8] Ibid, p.21

[9] Ibid, p.94

[10] https://www.valimised.ee/en/archive/statistics-about-internet-voting-estonia

technology in elections and observing it. If election technology is implemented and developed based on the best information security management system (ISMS) practices or in compliance with national or international information security standards (in the Estonian case, the national standard E-ITS or ISO 27001 information security standard), it also would follow the advice of the handbook. Furthermore, that would at least partially help observers answer the question of scope and compliance outlined in the previous paragraph.

The handbook serves as a solid starting point to assess the cybersecurity of elections. It is important to note that elections and their use of technologies can vary – the most obvious example being voting machines. EOMs will need to adjust to context and circumstances as a result but keep key principles at the heart of their assessments.

4 Conclusions

The new ODIHR Handbook on Observing ICT in Elections builds on the methodology outlined in the 2013 Handbook on Observation of NVT which dealt with electronic voting, counting and tabulation of election results, by updating it with the most recent standards in the ICT field and introducing two new aspects for observation and assessment. These new aspects, are the different 'ancillary' ICT-based election systems and processes, mostly related to voter registration or identification and results management systems, and the various cybersecurity issues that may occur during election periods. The handbook utilizes the experiences and lessons learned from OSCE participating States, and other credible organizations and institutions specialized in the fields of elections, ICT, and cybersecurity.

Cybersecurity threats during election periods are relatively new challenges for OSCE countries. In line with its mandate, ODIHR aims to support OSCE participating States to assist them in implementing their human dimension commitments; thus, to organize their elections in line with recognized international obligations and standards. The guidance provided in the cybersecurity section will support observers to better detect the main threats stemming from these types of challenges, how they affect electoral processes, and what is the impact of these challenges on elections integrity. While OSCE participating States have different institutional structures for tackling these threats and some of these systems provide for robust and comprehensive responses, this short paper uses the Estonian experience as an example of a country with relatively long tradition in dealing with ICT-based elections and processes.

Overall, the contribution that this handbook makes in the development of election observation methodology related to cybersecurity is two-fold. From the one side, it recognizes that the complex and dynamic nature of ICT developments is intrinsically followed by different cybersecurity threats and notes the variety of the actors involved in today's electoral processes that need to be equipped with adequate cybersecurity training, skills and readiness to respond. On the other side, given the transnational nature of cybersecurity challenges and global or regional importance of conducting democratic and genuine elections, the handbook notes that effective and comprehensive responses can be only effective with rigorous multilateral co-operation, of which election observation is a part. Therefore, OSCE participating States and other countries should consider developing and effectively implementing global ICT standards, including on cybersecurity relevant to electoral processes. While the guidance provided in the handbook is aimed mostly for ODIHR observers, it can also assist OSCE participating States in their efforts to introduce or appropriately tackle ICT challenges, including on cybersecurity during elections.

The Digital Service Act and Elections

Leontine Loeber [1]

Abstract: The use of technology in the electoral process has not only been applied to the process of voting but also to the ways voters are informed about elections. This information can be about the procedural aspects of voting, such as how to register, when to vote and the dissemination of results, but also on the parties and candidates and their positions. While the use of internet makes it easier for electoral management bodies to inform voters about elections and the way they can participate, there is a growing fear that the use of social media has let to new ways to manipulate elections, by spreading misinformation. This misinformation can lead to the loss of trust in elections and their results, as was clearly shown in the case of the 2020 US elections. Within the European Union, this fear led to the adoption of the Digital Service Act, which, among other things, regulates how social media platforms should combat misinformation within the electoral process. This paper maps the requirements of the Act and their consequences for the use of social media in elections. It highlights the benefits, but also the possible drawbacks of the choices that were made by the EU.

Keywords: Trust, Digital Service Act, Disinformation

1 Introduction

One key aspect of democracy in liberal systems is the defense of different freedoms, which are often a precondition for democracy to exist. Freedom of speech, in particular, is a liberty whose protection is regarded inherent to democracy. Key functions of democracy, such as taking (good) decisions, evaluating leaders (in order to confirm or remove them), and achieving compromises, become harder when citizens lack freedom of expression. Without freedom of speech, some important information and ideas will go unheard or remain underground, including opinions that do not align with those of the majority and thus often those in power. Another important aspect of democracy is plurality of the media. Plurality of the media has two aspects, content pluralism (there is a significant plurality of publicly available opinions and analyses) and source pluralism (there are multiple media outlets). Although these two forms are not completely independent, they are also not completely dependent; source pluralism does not guarantee content pluralism and content pluralism does not imply source pluralism.

However, not all content is good for a democracy. With the introduction of new communication technologies through the internet, the way news reaches citizens has

[1] Vrije Universiteit Amsterdam, leontine_loeber@xs4all.nl https://orcid.org/0000-0002-2272-6887.

doi:10.18420/e-vote-id2024_15

changed. The internet is nowadays recognized as the most diverse and so far, most fragmented regulated source of access to information [GR22]. Where in theory, more information should lead to more discourse and debate, stimulating the deliberative process that drives democracy, in practice, it does not really seem to work that way. If citizens receive information that is predominantly based on their pre-existing opinions and beliefs, the changes of an open and fruitful debate on different points of view might actually diminish [To23]. This process creates so called information bubbles, which are even more dangerous when they are combined with misinformation and fake news. These false narratives can distort public perceptions, fuel societal divisions, and undermine trust in democratic institutions, such as electoral management bodies. Although misinformation itself is not a new threat to clean and integer elections, the unique challenge posed by the speed and scale of misinformation on social media is. During critical times, like elections, the spread of misinformation can have particularly dire consequences, as was apparent during the 2020 US elections and riots that followed in January 2021 [BT22]. Without any policies to protect users from harmful content, large communication platforms can be manipulated to be used to propagate fear, (anonymous) harassment, and public opinion manipulation, particularly during political campaigns or public crises [Lo23].

This misinformation does not only target political parties and candidates, even though most of the academic attention seems to be directed to that area. Electoral management bodies are more and more also the target of misinformation. This has not gone unnoticed by governments and different legislative attempts have been made, or are debated, to combat the misinformation during the electoral process. This paper looks at the use of social media by EMBs and the risks that have risen from this. It then examines a specific legislative measure that was recently adopted, the EU Digital Service act, to see how did act attempts to counteract misinformation. It then discusses the benefits but also possible negative consequences of the act and closes with a conclusion.

2 The use of social media and internet in the electoral cycle

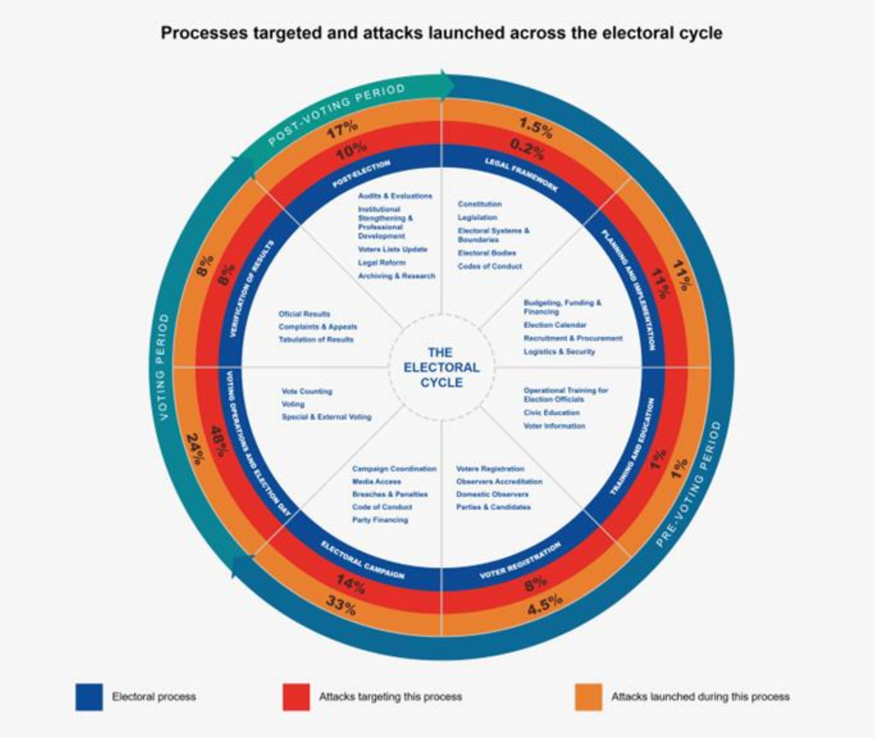
Social media and the internet, or digital communication technologies play an important role in elections. As Habermas already noted in 1991, the form and quality of democratic politics depend highly on the communication technologies that are used within a society. These technologies can have beneficial effects on the participation of citizens in the decision-making process, leading to ideas about the expansion of e-democracy and e-government [HL99]. Nowadays, social media has become fundamental in political organizing and activities and in many countries, people get much of their information on public matters from social media sites [Si15].

Electoral management bodies can use these forms of communication technologies to inform citizens in a large-scale and relatively cheap way about upcoming elections and the process of voter registration. Social media platforms and communication through the internet can also play a helpful role in providing transparency about the results of elections, by making these quickly and easily available online after the elections. This means that voters can see these results for the whole country without having to go to different polling places or municipalities in person. Data from International Idea shows that out of the 175 countries in their database, 113 (64.5%) use the internet to publish election results.[2]

3 Concerns and options for interference

As with the introduction of any new technology in the election process, the use of social media and internet to inform voters also created new ways to manipulate elections. A clear example of this was the operations of Cambridge Analytica during the 2016 US presidential elections and the alleged Russian interference in the UK elections. These cases raised questions about the impact of social media on elections and offline politics. The Capitol Hill riots in January 2021 added to these concerns.

In a comprehensive study done by IDEA in 53 countries on cases of disinformation in the online space targeting national elections, they identified cases in 92% of these countries. Most of these cases happened during the campaign phase of the electoral cycle (33%) and during the phase of voting operations and Election Day (24%).



Processes targeted and attacks launched across the electoral cycle

IDEA found frequent cases of false or deceiving information on voting methods and

[2] https://www.idea.int/data-tools/data/question?question_id=9365&data-base_theme=327.

conditions potentially leading to disenfranchisement. Examples include narratives such as "electoral IDs of people over 70 years old are being revoked" and notifications received by users of instant messaging apps regarding the invalidation of their identification cards. Incorrect information regarding certain electoral procedures and sanitary hygiene requirements for participation in the elections was also common, specifically for the elections held during the Covid-19 pandemic.

Other disinformation that was documented aimed at discrediting the election process and results, claiming that there had been unlawful manipulation of ballot papers particularly during voting operations and election day. Claims such as "pre-marked ballot papers were discovered", "ballot boxes were found filled with ballot papers", "special pens will be used with an ink that can be erased and changed", and "pencil marks are erased in the counting process" were meant to cast doubt over the fairness of the process and the accuracy of the results. The same can be said for disinformation about mail voting and voting from abroad.[3]

This misinformation about possible fraud is especially dangerous, as a study by IFES confirmed, If, for example, narratives of fraud or malpractice in polling, counting or results transmission leave citizens feeling disenfranchised, this may undermine public acceptance of the results or increase the chance of post-electoral violence [Re20].

When looking at the targets of disinformation, although it is still mostly the EMB itself, the case of the US elections of 2020 shows that EMB officials are now also under attack. This can have very serious consequences for those people, where cases of death threats are no longer uncommon.[4] This is turn has led to an increase in resignation by poll workers, leaving EMBs with less experienced people [RG24].

4 The response of the European Union: The Digital Service Act

In response to the threats outlined above, the European Union in November 2019 launched the European democracy action plan. This action plan addresses the EU institutions, national governments and parliaments − who have primary responsibility for ensuring the sound functioning of democracy −, as well as other national authorities, political parties, media and civil society, and online platforms. In full respect of national competences, it sets out a reinforced EU policy framework and specific measures to:
promote free and fair elections and strong democratic participation, support free and independent media and counter disinformation.[5]

The action plan consists of legislative proposals, dealing with greater transparency regarding the area of political advertising and clearer rules on the financing of European

[3] https://www.idea.int/theme/information-communication-and-technology-elec-toral-processes/information-environment-around-elections#targets.
[4] https://www.reuters.com/investigates/special-report/usa-election-threats-law-enforcement/.
[5] https://ec.europa.eu/commission/presscorner/detail/en/mex_24_268.

political parties, accompanied by initiatives for better cooperation between the member states. The plan also addresses countering disinformation, through the Digital Service Act.

The Digital Service Act (DSA) and the accompanying Digital Markets Act (DMA) comprise a legislative package to protect fundamental rights and create a fair marketplace for businesses in the digital space. One of the key points of the DSA is also to seriously attempt to mitigate the growing harm caused by fake news, disinformation and computational propaganda. The DSA's model for regulating platforms focuses on transparency towards the user, by prescribing obligations for reporting on content moderation and removal decisions. The DMA regulates the markets in which platforms operate by imposing technical obligations for these platforms and markets.

The DSA primarily applies to online intermediary services (art. 2), online platforms (art 3i) and online search engines (art. 3j). These are defined as online marketplaces, social networks, content-sharing platforms, app stores and online travel and accommodation platforms. In articles 4-6, the DSA establishes service provides liability conditions for any uploaded illegal content that they have actual knowledge of. An important distinction that the DSA makes is that between VLOPs, or very large online platforms and VLOSEs (very large online search engines) on the one hand and all other service providers on the other hand. VLOPs and VLOSEs are those services that have an average number of uses equal to or higher than 45 million and which are designated as such by the EU. Examples of VLOPs that could be used in the election process are Facebook, Instagram, LinkedIn, TikTok, X, Youtube and Wikipedia. For VLOSEs, it concerns Google Search and Bing.

All platforms, regardless of their size, are obliged to assess and remove flagged content through notice and action mechanisms (art. 16-17). Only the very large platforms, however, are also burdened with proactive content moderation. Art. 34 legally binds VLOPs and VLOSEs to 'diligently identify, analyze and assess any systemic risks in the Union stemming from the design or functioning of their service and its related systems, including algorithmic systems, or from the use made of their services.' Relevant for the impact of the DSA in elections is the clause that states: 'This risk assessment shall be specific to their services and proportionate to the systemic risks, taking into consideration their severity and probability, and shall include the following systemic risks (…)
(c) any actual or foreseeable negative effects on civic discourse and electoral processes, and public security'.[6] Art. 35 requires companies to take risk mitigation measures.

The European Commission can set guidelines based on art. 35 that VLOPs and VLOSEs must follow. On April 26th, 2024, the Commission published the first of these guidelines, dealing specifically with the mitigation of systemic risks for electoral processes.[7] In the

[6] DSA Regulation (EU) 2022/2065 of the European Parliament and of the Council of 19 October 2022 on a Single Market For Digital Services and amending Directive 2000/31/EC (Digital Services Act).

[7] Communication from the Commission – Commission Guidelines for providers of Very Large Online Platforms and Very Large Online Search Engines on the mitigation of systemic risks for electoral processes pursuant to Article 35(3) of Regulation (EU) 2022/2065, C/2024/2537.

guidelines, specific attention is given to misinformation on the electoral process itself. The guidelines state:
“Specifically, mitigation measures aimed at addressing systemic risks to electoral processes should include measures in the following areas:
a) Access to official information on the electoral process. To improve voter turnout and prevent the spread of misinformation, disinformation and FIMI on the electoral process itself, best practice for providers of VLOPs and VLOSEs is to facilitate access to official information concerning the electoral process, including information on how and where to vote, based on official information from the electoral authorities of the Member States concerned. Such information could be provided for example by means of information panels, banners, pop-ups, search interventions, links to websites of the electoral authorities, specific election information tabs or a dedicated part of the platform. When designing and implementing such mitigation measures, the Commission recommends that providers of VLOPs and VLOSEs take principles such as inclusiveness and accessibility into account.[8] In addition to this, it is also recommended that there should be clear, visible and non-deceptive indications of official accounts, such as the accounts of electoral authorities, so that it is more difficult to impersonate these accounts.

The guidelines note that procedures and organizational structures for elections differ between Member States. The DSA requires each Member State to appoint a Digital Service Coordinator. The guidelines recommend that the providers have regular interaction with these Coordinators and relevant national authorities to ensure that they have enough knowledge about the specific national procedures to be able to take appropriate risk mitigation measures.[9] In addition, the guidelines state that:
‘Alongside cooperation with national authorities, providers of VLOPs and VLOSEs are also recommended to establish strong cooperation with relevant non-state actors as such actors play a key role in protecting electoral processes. Prior to the elections, providers of VLOPS and VLOSEs may organize meetings as well as establish channels of regular communication with non-state actors active in electoral processes such as academics, independent experts, civil society organizations and representatives of various communities, and invite them to share their independent expertise, insights and observations that can help identify risks that may require mitigation measures and contribute to the development of such mitigation measures.’[10] To help VLOPs and VLOSEs in this regard, they can make use of the EDMO Taskforce on Elections, which is composed of independent fact-checkers, academics, and media literacy specialists.

If suspected of a breach of the DSA, the European Commission can request specific information from the company that is involved. If such a request is not answered within the deadline set by the Commission, it can impose fines up to 1% of the provider's total annual income or worldwide turnover and periodic penalties up to 5% of the provider's average daily income or worldwide annual turnover. The Commission can also impose

[8] C/2024/2537, point 27.
[9] C/2024/2537, point 43.
[10] C/2024/2537, point 47.

fines up to 1% of the provider's total annual income or worldwide turnover for incorrect, incomplete, or misleading information in response to a request for information.

5 Benefits and drawbacks

As with any new piece of legislation, there are benefits and drawbacks. The question is always if the first outweighs the latter. The DSA has just been introduced, so it remains to be seen if it will actually help in combatting disinformation. However, the European Commission seems to be ready to put it into use. On May 17th, 2024, the EC decided to increase its enforcement actions against Microsoft. On March 14th, it had already requested information on specific risks concerning the use of Bing's generative AI features, notably “Copilot in Bing” and “Image Creator by Designer”. Now, it has asked Microsoft to provide internal documents and data that was not previously disclosed. Microsoft has until May 27th to adhere to the request. The EC suspects that Bing may have breached the DSA for risks linked to generative AI, which could mislead voters, amongst others because of the viral dissemination of deepfakes. Generative AI is one of the risks identified by the Commission in its guidelines on the integrity of electoral processes, in particular for the upcoming elections to the European Parliament in June.[11]

It is therefore reasonable to assume that the moderation of content from dominant platforms affects the nature of what is debated in society. Due to the pervasiveness of certain platform services and their unique role as citizen fora for expressing opinion, protecting freedom of expression from the undue interference of algorithmic decisions is, nowadays, arguably as important as protecting the legal right of freedom of expression, which is customarily defined as a right against the interference of public authority.

There is a fine line between giving critique on the way elections are run and unfoundedly creating the image that there has been fraud. Citizens, NGOs and political parties and candidates should be able to examine the way elections are run and raise concerns if there are mistakes made or things could be improved. Those involved in content moderation should be hesitant to be too quick to remove such concerns from social media outlets. However, when these concerns are unfounded, aimed to disrupt the process and in danger of compromising trust in the outcome of the elections, action needs to be taken. In order to be able to distinguish between the two, it is of the utmost importance that EMBs are honest about possible mistakes that were made during the electoral process. The more transparent they are about the way elections are run and problems that they have encountered in the process, the easier it is to determine which concerns should be taken seriously and which are clearly misinformation. This does require more capacity in EMBs to inform the public and the press in an open, honest and quick matter, so that exaggerated stories can be debunked as soon as possible.

The main risk from the way the DSA is set up is that it is the VLOPs and VLOSEs that are responsible for the risk-mitigating measures and thus for the choices which content

[11] https://digital-strategy.ec.europa.eu/en/news/commission-compels-microsoft-provide-information-under-digital-services-act-generative-ai-risks.

will be allowed, and which content will be taken down. We should be aware that these providers are entrepreneurs without any democratic mandate, but that their decisions can dramatically shape political and cultural discourse. Platforms can influence elections by silencing certain speech or information. The question is therefore if platforms alone should be the ones to decide what content is available online and what is not [FG23].

This is especially important during elections. In an election period, which is usually very short, these decisions must be made quite quickly if they are to be effective. With the current amount of attention on misinformation, providers may be overrun with requests to judge and take-down information. This could pressure them to take the 'safe' route and remove more content than necessary, including reasonable critical questions that are raised on the way the elections were run. When that would happen, the DSA could seriously hinder the freedom of speech and have unwanted detrimental effects on the integrity of elections.

It is therefore very important that the implementation of the DSA is closely monitored in the Member States. Transparency of the providers on what information was flagged and their decisions on this information is crucial in order to see if the balance that the DSA tries to find between freedom of information and the right to be protected against disinformation is indeed achieved. EMBs play a crucial role in this process as they should inform providers about the correct procedures for the elections in their country, notify them if disinformation is being spread, but also be open and honest about justified critique of their functioning.

6 Conclusions

Although misinformation itself is not a new threat to clean and integer elections, the unique challenge posed by the speed and scale of misinformation on social media is. It will be hard to find a balance between the right of free speech and the right of citizens to be properly informed, meaning that they can trust the information that they read online. The DSA tries to strike this balance, but until it has been tried and tested in practice, there are no guarantees that it is able to do so. Although content moderation is necessary, this should not mean that critique on the way EMBs run elections should not be allowed. It is that critique that pushes EMBs to improve their processes and to periodically re-evaluate if they succeed in delivering properly run elections. This also means that EMBs, more than in the past will have to have the capacity to respond to meaningful critique in a comprehensive manner, but also to debunk misinformation. This is necessary to maintain or even restore trust that voters have in the electoral process. EMBs cannot rely on the providers of social media alone to take their role in combatting misinformation, a clear strategy how to deal with issues that can be raised through social media during the different phases of the election cycle by those aiming to undermine this trust needs to be in place. More academic research in how misinformation targets EMBs and ways to deal with this, could benefit EMBs and thus all those who believe in democracy.

7 Bibliography

[BT22] Bowden, M., & Teague, M. (2022). *The Steal: The Attempt to Overturn the 2020 Election and the People Who Stopped It*. Atlantic Monthly Press.

[FG23] Frosio, G., & Geiger, C. (2023). Taking fundamental rights seriously in the Digital Services Act's platform liability regime. *European Law Journal*, *29*(1-2), 31-77.

[GR22] De Gregorio, G., & Radu, R. (2022). Digital constitutionalism in the new era of Internet governance. *International Journal of Law and Information Technology*, *30*(1), 68-87.

[HL99] Hague, B. N., & Loader, B. (Eds.). (1999). *Digital democracy: Discourse and decision making in the information age*. Psychology Press.

[Lo23] Loi, M. (2023). Making sense of the Digital Services Act. How to define platforms' systemic risks to democracy.

[Re20] Reppell, Lisa, Beata Martin-Rozumiłowicz, and Vasu Mohan. (2020). Preserving Electoral Integrity During an Infodemic. Washington DC: IFES.

[RG24] Roberts, J. M., & Greenberger, M. (2024). Election Worker Recruitment and Retention in North Carolina.

[Si15] Silverman, C. (2015). Lies, damn lies and viral content.

[To23] Tornada, J. (2023). How (Not) To deal with the Bubble Effect in Cyberspace: The Case of the EU and Digital Services Act. *Brooklyn Journal of International Law*, *49*(1), 97.

Autorenverzeichnis