Trust and Privacy in Electronic Voting

Navid Abapour

Surrey Centre for Cyber Security
Computer Science Research Centre
University of Surrey

n.abapour@surrey.ac.uk

Supervisors: Cătălin Drăgan and Ioana Boureanu

December 9, 2024

Introduction and Aim

Why electronic voting instead of classical one?

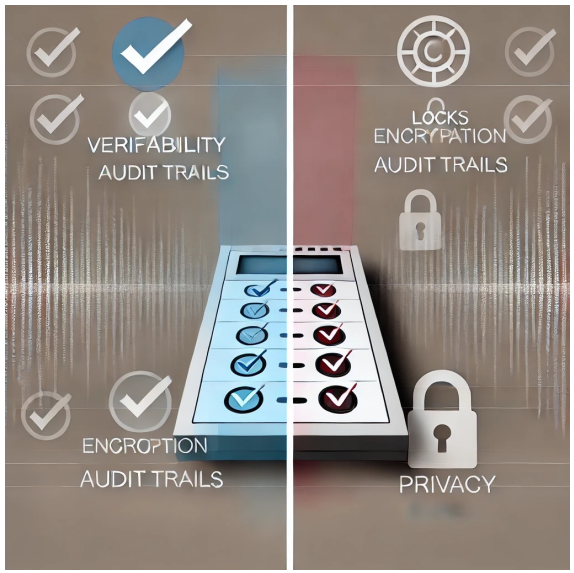
- Speed, Accuracy, Accessibility, Cost-Effective, Convenience, Transparency
- Security & Privacy 🤭

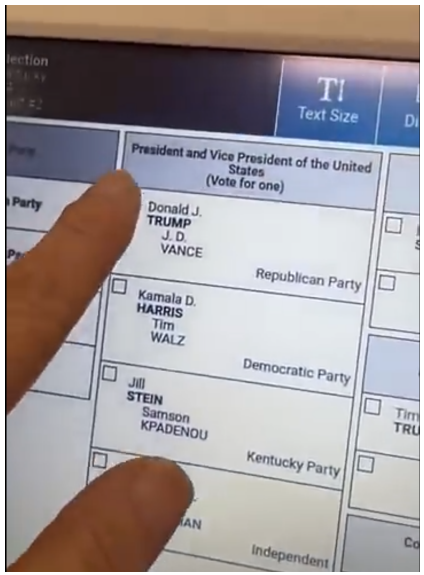
Many academic methods have been proposed.

- Helios, Belenios, ElectionGuard, Prêt-à-Voter, Selene, sElect, ...

Then, why can we not vote online after 40 years?

- Because of a set of conflicting properties: e.g., Verifiability and Privacy.





https://www.youtube.com/watch?v=dvxdb4QPSZc

Research Problem and Challenges

Why is this so hard?

- Inexpressive information in ElectionGuard documentation
- Insufficient evidence for equivalency of ElectionGuard and Helios[1]
- Unconventional way of using cryptographic primitives in ElectionGuard

[1] An online voting system

ElectionGuard

What is an SDK and ElectionGuard?

- An open-source software development kit
- Has been implemented in public elections of
  - US such as Wisconsin, California, Maryland, etc.
  - France and Switzerland
- End-to-end verifiable cryptographic toolkit

ElectionGuard

- Who are the entities?
  - Election Authority, Guardians, Voters
  - [Election] Verifier
- What are the components of this SDK?
  - Parameter Requirements
  - Key Generation
  - Ballot Encryption
  - Verifiably Decryption
  - Independent Verification
    - Verification of Key Generation, Ballot Correctness, and Election Record
- How does it work?
  - Setup Election Authority
  - Setup Guardians
  - Vote
  - Verification of Votes
  - Tally and Verification of Result

ElectionGuard - Primitives

Which cryptographic primitives this SDK is using?

- Hash Function
- Hash-based Message Authentication Code
- Modified ElGamal
- Hashed ElGamal
- Verifiable Secret-sharing Scheme
- Proof Systems
  - Schnorr Zero-knowledge Proof
  - Chaum-Pedersen Zero-knowledge Proof

Agreement on the Public Key from All Guardians

	$\mathbf{G}_i$		$\mathbf{G}_\ell$
1	$\mathsf{ParVer}([\mathsf{aux}], \mathsf{pp}, \mathsf{H_P}, \mathsf{H_B}, \mathsf{H_E})$		$\mathsf{ParVer}([\mathsf{aux}], \mathsf{pp}, \mathsf{H_P}, \mathsf{H_B}, \mathsf{H_E})$
2	$(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{sk}_i)$		$(\mathsf{sk}_\ell, \mathsf{pk}_\ell) \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{sk}_\ell)$
3	$\pi_i \leftarrow \mathsf{ZKP}_1.\mathsf{P}_1(\mathsf{sk}_i, \mathsf{pp}, (\mathsf{pk}_i, [\mathsf{aux}]))$		$\pi_\ell \leftarrow \mathsf{ZKP}_1.\mathsf{P}_1(\mathsf{sk}_\ell, \mathsf{pp}, (\mathsf{pk}_\ell), [\mathsf{aux}]))$
4		$\mathsf{pk}_i, \overbrace{(c_i, z_i)}^{\pi_i} \longrightarrow$	
5			**if** $(\mathsf{pk}_i \notin \mathbb{Z}_p^r) \vee (c_i \notin \mathbb{Z}_q)$ **then** $\perp$
6			**if** $(\neg\mathsf{ZKP}_1.\mathsf{V}_1(\mathsf{pp}, (\mathsf{pk}_i, [\mathsf{aux}]), \pi_i))$ **then**
7			$\mathsf{Complaint}_1(G_\ell, G_i, \mathsf{pk}_i, \pi_i)$
8		$\mathsf{pk}_\ell, \overbrace{(c_\ell, z_\ell)}^{\pi_\ell} \longleftarrow$	
9	**if** $(\mathsf{pk}_\ell \notin \mathbb{Z}_p^r) \vee (c_\ell \notin \mathbb{Z}_q)$ **then** $\perp$		
10	**if** $(\neg\mathsf{ZKP}_1.\mathsf{V}_1(\mathsf{pp}, (\mathsf{pk}_\ell, [\mathsf{aux}]), \pi_\ell))$ **then**		
11	$\mathsf{Complaint}_1(G_i, G_\ell, \mathsf{pk}_\ell, \pi_\ell)$		

Exchange Shares of Secret Keys

	$\mathbf{G}_i$		$\mathbf{G}_\ell$
1	$\mathsf{ParVer}([\mathsf{aux}], \mathsf{pp}, \mathsf{H_P}, \mathsf{H_B}, \mathsf{H_E})$		$\mathsf{ParVer}([\mathsf{aux}], \mathsf{pp}, \mathsf{H_P}, \mathsf{H_B}, \mathsf{H_E})$
2	$(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{sk}_i)$		$(\mathsf{sk}_\ell, \mathsf{pk}_\ell) \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{sk}_\ell)$
3	$\pi_i \leftarrow \mathsf{ZKP}_1.\mathsf{P}_1(\mathsf{sk}_i, \mathsf{pp}, (\mathsf{pk}_i, [\mathsf{aux}]))$		$\pi_\ell \leftarrow \mathsf{ZKP}_1.\mathsf{P}_1(\mathsf{sk}_\ell, \mathsf{pp}, (\mathsf{pk}_\ell), [\mathsf{aux}]))$
4		$\mathsf{pk}_i, \overbrace{(c_i, z_i)}^{\pi_i} \rightarrow$	
5			**if** $(\mathsf{pk}_i \notin \mathbb{Z}_p^r) \vee (c_i \notin \mathbb{Z}_q)$ **then** $\perp$
6			**if** $(\neg\mathsf{ZKP}_1.\mathsf{V}_1(\mathsf{pp}, (\mathsf{pk}_i, [\mathsf{aux}]), \pi_i))$ **then**
7			$\mathsf{Complaint}_1(G_\ell, G_i, \mathsf{pk}_i, \pi_i)$
8		$\mathsf{pk}_\ell, \overbrace{(c_\ell, z_\ell)}^{\pi_\ell} \leftarrow$	
9	**if** $(\mathsf{pk}_\ell \notin \mathbb{Z}_p^r) \vee (c_\ell \notin \mathbb{Z}_q)$ **then** $\perp$		
10	**if** $(\neg\mathsf{ZKP}_1.\mathsf{V}_1(\mathsf{pp}, (\mathsf{pk}_\ell, [\mathsf{aux}]), \pi_\ell))$ **then**		
11	$\mathsf{Complaint}_1(G_i, G_\ell, \mathsf{pk}_\ell, \pi_\ell)$		
12	$\mathsf{shr}_i \leftarrow \mathsf{Share}(\mathsf{pp}, t, n, \mathsf{sk}_i)$		$\mathsf{shr}_\ell \leftarrow \mathsf{Share}(\mathsf{pp}, t, n, \mathsf{sk}_\ell)$
13	$\forall j \in U: \ E_\ell \leftarrow \mathsf{Enc}_5(\mathsf{pp}, \mathsf{shr}_i, \mathsf{pk}_\ell)$		$\forall j \in U: \ E_i \leftarrow \mathsf{Enc}_5(\mathsf{pp}, \mathsf{shr}_\ell, \mathsf{pk}_i)$
14		$E_\ell \rightarrow$	
15			$\forall j \in U. \ell \neq j: \ \mathsf{shr}_i \leftarrow \mathsf{Dec}_5(\mathsf{pp}, \mathsf{sk}_\ell, E_\ell)$
16			**if** $(\neg\mathsf{Vf}(\mathsf{pp}, t, n, \mathsf{pk}_i, (\mathsf{shr}_i, (\ell, \mathsf{sk}_\ell))))$ **then**
17			$\mathsf{Complaint}_2(G_\ell, G_i, \mathsf{pk}_i, E_\ell, (\ell, \mathsf{sk}_\ell))$
18		$E_i \leftarrow$	
19	$\forall j \in U. i \neq j: \ \mathsf{shr}_\ell \leftarrow \mathsf{Dec}_5(\mathsf{pp}, \mathsf{sk}_i, E_i)$		
20	**if** $(\neg\mathsf{Vf}(\mathsf{pp}, t, n, \mathsf{pk}_\ell, (\mathsf{shr}_\ell, (i, \mathsf{sk}_i))))$ **then**		
21	$\mathsf{Complaint}_2(G_i, G_\ell, \mathsf{pk}_\ell, E_i, (i, \mathsf{sk}_i))$		

ElectionGuard - Vote and Verification

How do the electorates vote?

Vote$(id, \text{vote}, \text{pk})$

1 : $(\mathbb{c}, \xi) \leftarrow \text{ElGamal}'.\text{Enc}(\text{pk}, \text{vote})$

2 : $\pi_1 \leftarrow \text{ZKP}_2.\text{P}_2(\xi, \text{pp}, \mathbb{c})$

3 : // The below range proof is for score election

4 : // $\pi_2 \leftarrow \text{ZKP}_5.\text{P}_5((\xi, \ell), \text{pp}, \mathbb{c}, [\text{aux}])$

5 : $B \leftarrow (id, \mathbb{c}, \pi_1)$

6 : **return** B

VerifyVote(B, pbb): **return** $\text{H}(B) \in pbb$

Counting Votes and Verification of Tally

	$\mathbf{G}_i$		$\mathbf{G}_\ell$
1	**if** $\neg$Valid(BB) **then return** $\perp$		**if** $\neg$Valid(BB) **then return** $\perp$
2	FBB $\leftarrow$ Policy(BB)		FBB $\leftarrow$ Policy(BB)
3	(A, B) $\leftarrow$ ElGamal′.Add(FBB)		(A, B) $\leftarrow$ ElGamal′.Add(FBB)
4	$\text{res}_\text{i} \leftarrow$ ElGamal′.Dec(sk_i, (A, B))		$\text{res}_\ell \leftarrow$ ElGamal′.Dec(sk_ℓ, (A, B))
5	**broadcast** res_i to all $\ell \neq i$	$\xrightarrow{\text{res}_\text{i}}$	
6		$\xleftarrow{\text{res}_\ell}$	**broadcast** res_i to all $i \neq \ell$
7	$\forall j \in U.\ w_j \leftarrow \prod_{\ell \in U \setminus \{j\}} \frac{\ell}{\ell - j} \mod q$		$\forall j \in U.\ w_j \leftarrow \prod_{\ell \in U \setminus \{j\}} \frac{\ell}{\ell - j} \mod q$
8	result $\leftarrow \prod_{j \in U} \text{res}_\text{j}^{w_j} \mod p$		result $\leftarrow \prod_{j \in U} \text{res}_\text{j}^{w_j} \mod p$

Counting Votes and Verification of Tally

	$\mathbf{G}_i$		$\mathbf{G}_\ell$
1	**if** ¬Valid(BB) **then return** $\perp$		**if** ¬Valid(BB) **then return** $\perp$
2	FBB ← Policy(BB)		FBB ← Policy(BB)
3	(A, B) ← ElGamal′.Add(FBB)		(A, B) ← ElGamal′.Add(FBB)
4	$\mathsf{res_i} \leftarrow \mathsf{ElGamal'.Dec}(\mathsf{sk}_i, (\mathsf{A}, \mathsf{B}))$		$\mathsf{res}_\ell \leftarrow \mathsf{ElGamal'.Dec}(\mathsf{sk}_\ell, (\mathsf{A}, \mathsf{B}))$
5	**broadcast** $\mathsf{res_i}$ to all $\ell \neq i$	$\xrightarrow{\mathsf{res_i}}$	
6		$\xleftarrow{\mathsf{res}_\ell}$	**broadcast** $\mathsf{res_i}$ to all $i \neq \ell$
7	$\forall j \in U.\ w_j \leftarrow \prod_{\ell \in U \setminus \{j\}} \frac{\ell}{\ell - j} \mod q$		$\forall j \in U.\ w_j \leftarrow \prod_{\ell \in U \setminus \{j\}} \frac{\ell}{\ell - j} \mod q$
8	$\mathsf{result} \leftarrow \prod_{j \in U} \mathsf{res_j}^{w_j} \mod p$		$\mathsf{result} \leftarrow \prod_{j \in U} \mathsf{res_j}^{w_j} \mod p$
9	$r_i \leftarrow\!\$\ \mathbb{Z}_q$		$r_\ell \leftarrow\!\$\ \mathbb{Z}_q$
10	$u_i \leftarrow g^{r_i} \mod p,\ v_i \leftarrow \mathsf{A}^{r_i} \mod p$		$u_\ell \leftarrow g^{r_\ell} \mod p,\ v_\ell \leftarrow \mathsf{A}^{r_\ell} \mod p$
11	**broadcast** (u_i, v_i) to all $\ell \neq i$	$\xrightarrow{(u_i, v_i)}$	
12		$\xleftarrow{(u_\ell, v_\ell)}$	**broadcast** (u_ℓ, v_ℓ) to all $i \neq \ell$
13	$u \leftarrow \prod_{j \in U} u_j \bmod p,\ v \leftarrow \prod_{j \in U} v_j \bmod p$		$u \leftarrow \prod_{j \in U} u_j \bmod p,\ v \leftarrow \prod_{j \in U} v_j \bmod p$
14	$c \leftarrow \mathsf{H}(\mathsf{H_E}; \mathsf{0x30}, \mathsf{pk}, \mathsf{A}, \mathsf{B}, u, v, \mathsf{result}), u, v)$		$c \leftarrow \mathsf{H}(\mathsf{H_E}; \mathsf{0x30}, \mathsf{pk}, \mathsf{A}, \mathsf{B}, u, v, \mathsf{result}), u, v)$
15	$\forall j \in U : c_j \leftarrow (c \cdot w_j) \mod q$		$\forall j \in U : c_j \leftarrow (c \cdot w_j) \mod q$
16	$\mathsf{share}_i = \sum_{j \in U} \mathsf{shr_{j,i}} \mod q$		$\mathsf{share}_i = \sum_{j \in U} \mathsf{shr_{j,i}} \mod q$
17	$z_i \leftarrow (r_i - c_i \cdot \mathsf{share}_i) \mod q$		$z_\ell \leftarrow (r_\ell - c_\ell \cdot \mathsf{share}_\ell) \mod q$
18		$\xrightarrow{z_i}$	
19		$\xleftarrow{z_\ell}$	
20	$\forall j.\ j \neq i :$ **recompute** (u_j, v_j) as (u'_j, v'_j) $u'_j \leftarrow g^{z_j} \left(\prod_{j \in U} \prod_{k=0}^{t-1} A_{j,k}^{j^k}\right)^{c_j}$ and $v'_j \leftarrow A^{z_j} \mathsf{res_j}^{c_j}$		$\forall j.\ j \neq \ell :$ **recompute** (u_j, v_j) as (u'_j, v'_j) $u'_j \leftarrow g^{z_j} \left(\prod_{j \in U} \prod_{k=0}^{t-1} A_{j,k}^{j^k}\right)^{c_j}$ and $v'_j \leftarrow A^{z_j} \mathsf{res_j}^{c_j}$
21	**if** $\exists\ j \in U \setminus \{i\} :\ ((u'_j, v'_j) \neq (u_j, v_j))$ **then return** $\perp$		**if** $\exists\ j \in U \setminus \{\ell\} :\ ((u'_j, v'_j) \neq (u_j, v_j))$ **then return** $\perp$
22	$z \leftarrow \sum_{j \in U} z_j \bmod q$		$z \leftarrow \sum_{j \in U} z_j \bmod q$
23	$\pi \leftarrow (c, z)$		$\pi \leftarrow (c, z)$
24	**return** (result, π)		**return** (result, π)

ElectionGuardSimple

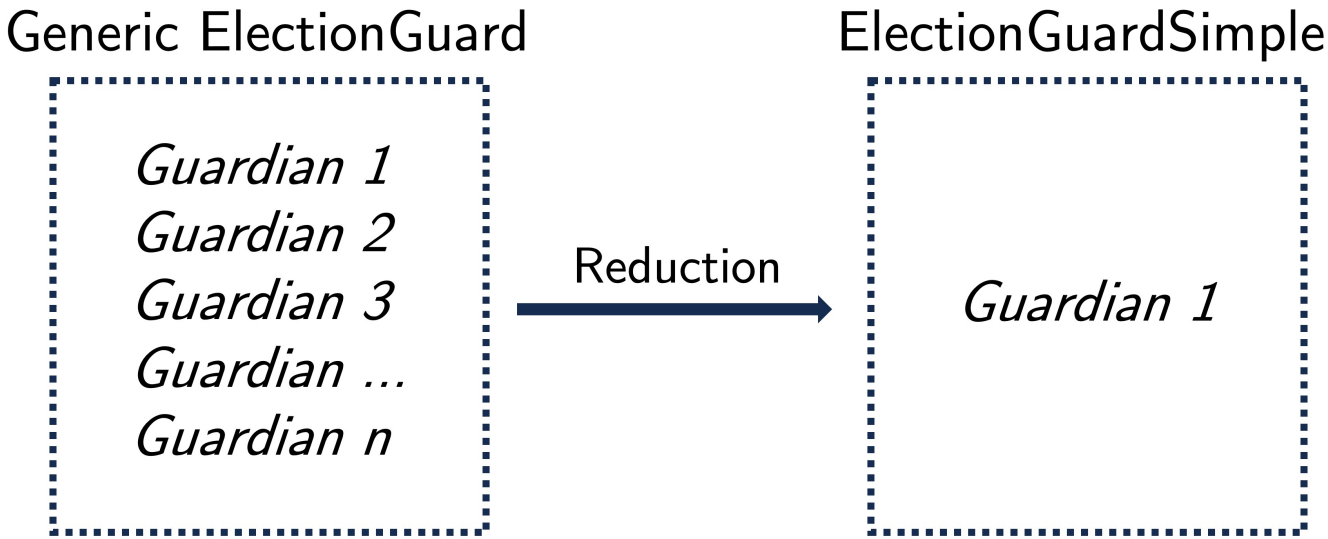
ElectionGuard with only one guardian...

1. **Setup:**
   - Verify system parameters ParVer.
   - Generate key pair $(sk, pk) \leftarrow$ KeyGen.
2. **Voting:**
   - Encrypt voter choice $(\mathbb{c}, \xi) \leftarrow$ ElGamal′.Enc(pk, vote).
   - Create proof of encryption correctness $\pi \leftarrow \text{ZKP}_2.\text{P}_2$.
   - Package ballot $B \leftarrow (id, \mathbb{c}, \pi)$.
3. **Vote Verification:**
   - Check if ballot hash $\text{H}(B)$ exists in the public bulletin board PBB.
4. **Tallying:**
   - Validate ballots and apply election policy $fbb \leftarrow$ Policy(BB).
   - Aggregate and decrypt result $r \leftarrow \text{Dec}(sk, \text{Add}(fbb))$.
   - Generate proof of correct tally $\Pi \leftarrow \text{P}$.
5. **Tally Verification:**
   - Verify tally correctness using proof $\text{V}((pk, PBB, r), \pi)$.

Hold on! 😮 Isn't this similar to Helios?!

Reduction to ElectionGuardSimple

Moving From Generic ElectionGuard to ElectionGuardSimple:



Attacks on any Generic ($n > 1$) are equivalent to attacks on Simple (n=1).

ElectionGuardSimple Privacy

Lemma

ElectionGuardSimple is ballot-private [1, 2], participation-private [2], strongly correct [3], and strongly consistent [3], assuming Modified ElGamal is IND-CPA, decisional Diffie–Hellman assumption, collision resistance-ness of the hash function, and zero-knowledge property of proof systems.

[1] Bernhard, D., Cortier, V., Pereira, O., Smyth, B., & Warinschi, B. (2011). Adapting Helios for provable ballot privacy. In V. Atluri & C. Diaz (Eds.), Computer Security – ESORICS 2011 (Vol. 6879, pp. 335–354).

[2] Bernhard, D., Kulyk, O., & Volkamer, M. (2017). Security proofs for participation privacy, receipt-freeness, ballot privacy for the Helios voting scheme. In Proceedings of the 12th International Conference on Availability, Reliability and Security (Article No. 1, pp. 1–10).

[3] Bernhard, D., Cortier, V., Galindo, D., Pereira, O., & Warinschi, B. (2015). SoK: A comprehensive analysis of game-based ballot privacy definitions. In 2015 IEEE Symposium on Security and Privacy (pp. 499–516).

Future Works

Going to submit a paper on ElectionGuard's privacy to IEEE CSF 2025 on January

Open Problems:

- Completing privacy analysis on ElectionGuard + reduction to ElectionGuardSimple
- Are there any general approaches to taking PKE and making them into TPKE?
- It is worth looking at whether ElGamal + Disjunctive Chaum-Pedersen is NM-CPA
- Is it possible for a system to have dispute resolution for all potential errors?
- Are there alternative reasonable definitions and mechanisms for resolving disputes?
- Can a system provide comprehensive dispute resolution where every conflict can be resolved using evidence generated by the election system?
- Is achieving dispute-free without relying on physical procedures[2] feasible?
- Designing an end-to-end verifiable dispute-free post-quantum online voting system

[2] e.g. paper ballot

Conclusion

What have we talked about today in this confirmation?

- Electronic voting is widely growing, but trust and privacy are seriously missing
- What are the cryptographic primitives of conventional voting systems
- How an online voting system is working? (Helios as Case Study)
- ElectionGuard: cryptographic SDK developed by Microsoft for voting
- Privacy analysis of ElectionGuard and Helios
- What contributions have I made in this one year?
- Future works on Dispute Resolution and Post-quantum Cryptography for E-voting

Acknowledgement

I would like to appreciate

- Ghazal Milani (PGR, Computer Science) for cordially supporting me
- Samaneh Rashidi (PGR, Psychology) for her guidance and advice
- Ioana Boureanu for sharpening my critical thinking
- Cătălin Drăgan for everything :)

References

- https://www.freepik.com (+ DALL·E 3)
- Benaloh, J., Naehrig, M., Pereira, O., & Wallach, D. S. (2024). ElectionGuard: A cryptographic toolkit to enable verifiable elections. Cryptology ePrint Archive, Paper 2024/955.
- Adida, B. (2008). Helios: Web-based open-audit voting. In Proceedings of the 17th Conference on Security Symposium (SS'08) (pp. 335–348). USENIX Association.
- Bernhard, M., Benaloh, J., Halderman, J. A., Rivest, R. L., Ryan, P. Y. A., Stark, P. B., Teague, V., Vora, P. L., & Wallach, D. S. (2017). Public evidence from secret ballots. In Electronic Voting: Second International Joint Conference, E-Vote-ID 2017, Bregenz, Austria, October 24-27, 2017, Proceedings (pp. 84–109).
- Bernhard, D., Cortier, V., Galindo, D., Pereira, O., & Warinschi, B. (2015). SoK: A comprehensive analysis of game-based ballot privacy definitions. In 2015 IEEE Symposium on Security and Privacy (pp. 499–516).
- Bernhard, D., Kulyk, O., & Volkamer, M. (2017). Security proofs for participation privacy, receipt-freeness, and ballot privacy for the Helios voting scheme. In Proceedings of the 12th International Conference on Availability, Reliability and Security (ARES '17) (Article 1, pp. 1–10)