2020 International Conference on Computing, Networking and Communications (ICNC): Communications and Information Security Symposium

Anonymous and Coercion-Resistant Distributed Electronic Voting

Ehab Zaghloul, Tongtong Li and Jian Ren

***Abstract*—Electronic voting (e-voting) presents a convenient and cost-effective alternative to current paper ballot-based voting. It provides many benefits such as increased voter turnout and accuracy in the decision-making process. While presenting many improvements, e-voting still faces serious security challenges that hinder its adoption. In this paper, we propose a novel remote e-voting scheme that can enhance the integrity, efficiency, and voter turnout of the election process. Our proposed work is secure and preserves the privacy of voters through secure multi-party computations performed by parties of differing allegiances. It also leverages a blockchain running smart contracts as a publicly accessible and tamper-resistant bulletin board to permanently store votes and prevent double-voting. In our security and privacy analysis, we show that our proposed scheme is secure against potential threats and provides voter anonymity. Our performance discussion shows that the proposed scheme is practical for large-scale elections.**

***Index Terms*—Remote e-voting, blockchain, smart contract.**

I. Introduction

Internet voting (i-voting) is a special type of remote electronic-voting (e-voting) where voters submit their desired votes remotely via the Internet, regardless of their location. Such systems aim to provide enhancements to current paper voting systems in that they can reduce operating costs, increase voter turnout, and improve election integrity. For example, an i-voting system enables absent personnel serving in the military to efficiently cast their votes toward an election while being deployed overseas with votes counted timely. However, because these systems are electronic, they may be subject to manipulation and fraud through malicious acts that range from powerful state-level attackers to independent hackers or even dishonest election staff running the system. In order for i-voting to be adopted in large-scale elections, significant research is required to develop systems that can achieve paper ballot equivalent or greater levels of security and anonymity assurance.

Estonia is one of the leading countries to develop and utilize i-voting systems on a wide scale since 2005. In 2019, the Estonian government reported that approximately 44% of their citizens used their i-voting system to cast their votes in the parliamentary elections [1]. Their system utilizes the *double-envelope* concept where eligible voters initially encrypt their votes using a publicly available election key (inner envelope) then digitally sign the result (outer envelope) using each voter's private key. Next, voters cast their votes to the election servers for the votes to be stored. Voters are able to individually verify whether their ballots have been included and counted within the election. In addition, voters are allowed to cast their votes multiple times with only the last vote being counted, with intention to prevent coercion. Once the voting phase is complete, the system validates the eligibility of voters and removes the outer envelope signature if valid. Finally, the system decrypts the inner envelope of the verified votes using the secret election key for the votes to be counted.

To audit the system and the legitimacy of the election, the Estonian government allows specialists to observe the election process. However, even with allowed auditability, it is insufficient to claim that the system is secure and anonymous based on its design. Primarily, the system validates the eligibility of voters before including their cast votes in the total count which may easily result in sacrificing the anonymity of the voters. In addition, with one entity controlling the election process and storing the votes in its servers, it becomes difficult to pinpoint potential cheating.

In this paper, we propose a novel i-voting scheme that leverages the existence of parties of an election with different allegiances. In our scheme, for voter data to become public, the opposing parties must collude to gain access to voter information. With this information, parties may coerce voters to vote in their favor. Given that parties will not wish to compromise votes and expose voter privacy through the act of collusion, this method decreases the likelihood of collusion in comparison to a single election authority. Our proposed scheme also stores votes in a distributed manner by relying on a blockchain, a publicly accessible bulletin board. This helps counter major issues such as double-voting, election results manipulation, and DDoS attacks.

The rest of this paper is organized as follows. In Section II, the related work is reviewed. In Section III, the problem formulation is described, outlining the system components and design goals. Next, in Section IV, the proposed scheme is presented. In Section V, we discuss the security and privacy of our work. Following that, in Section VI, we present a performance evaluation of our proposed scheme. Finally, in Section VII, we conclude our work.

II. Related Work

Previous work that achieves coercion-resistance and remote e-voting dates back to 2005 when the work by Juels *et al.* [2] was introduced and later refined resulting in Civitas [3]. Although proven to be secure, the security came at the price of tabulation which is quadratic in respect to the total number of ballots being submitted in an election. Shortly, Helios [4] was proposed as a web-based open-audit voting system for elections where coercion is not a serious problem. The system achieves privacy using mixnets [5] and was later improved by Demirel *et al.* [6] by replacing

978-1-7281-4905-9/20/$31.00 ©2020 IEEE

the mixnets with homomorphic encryption and multi-party tallying. However, these systems primarily focus on universal verifiability while intentionally not taking into account coercion-resistance.

In contrast, Selections [7] is a system that was proposed in which voter authentication relies on possession of certain voter passwords, allowing them to generate panic passwords in cases of coercion. This system relies on zero-knowledge proofs during the vote casting phase. Other systems such as [8], use a linear-time scheme to remove duplicated votes which may be submitted by voters to avoid coercion. This system also relies on voters indicating which electoral roll their votes belong to so that tallying authorities can identify which votes should be included in the total count. This results in faster tallying during the tabulation process. However, it requires additional trust on the election trustees to add dummy ballots to make the system coercion-resistant.

Based on concepts from [7] and [8], a protocol was introduced in [9] that requires voters to specify an anonymity set where each voter claims to be one of the voters within the set. This resulted in additional voter overhead costs during the authentication phase. Later, Zeus [10] was proposed following the initial framework of Helios [4] where mixing is performed using external agents. Although it provides universal verifiability, the system is not coercion-resistant as it provides voters with receipts at the end of voting.

More recently, a smart contract for a boardroom voting system [11] was implemented over the Ethereum blockchain using the Open Vote Network [12]. The main advantage of this system is that it is completely decentralized, provides self-tallying and achieves E2E characteristics. However, the system is limited to small elections due to its expensive computations including zero-knowledge proofs that are required.

III. Problem Formulation

In this section, we introduce the main components of our proposed work followed by our design goals.

A. System Components

The system consists of six main entities.

Voters: A set of eligible voters $\{v_i \in \mathcal{V} \mid 1 \leq i \leq n\}$ that are granted the right to cast a vote in an election.

Registrar $\mathcal{R}$: An entity that generates unique and random digital ballots to be shared with voters anonymously.

Moderator $\mathcal{M}$: An independent party responsible for concealing the identities of voters and delivering the ballots to them anonymously.

Election Candidates: A set of eligible candidates $\{\mathsf{cand}_k \in \mathcal{C} \mid 1 \leq k \leq m\}$ that can run in an election.

Blockchain Network: A non-trusted peer-to-peer network that maintains a publicly accessible blockchain and runs the election smart contracts.

Tallying Authority: A party that performs vote tabulation at the end of the casting votes phase.

B. Design Goals

Our proposed scheme provides the following features:

Eligibility: Voting is limited to eligible voters that satisfy the voting prerequisites.

Double-voting resistant: Each eligible voter is entitled to only a single vote counted toward the election.

Anonymity: The identity of each voter is untraceable and cannot be linked to the cast votes.

Coercion-resistant: The voting process protects voters against coercers.

Voter verifiability: Voters can verify that their cast votes have been included correctly.

Election result manipulation-resistant: Votes are concealed throughout the entire casting votes phase preventing last-minute eligible voters from manipulating the election results in a close race.

IV. The Proposed Scheme

Our proposed scheme consists of five main sequential phases, each occurring in a specific time frame predefined by the election organizer. We assume that the registrar $\mathcal{R}$ and the moderator $\mathcal{M}$ are two opposing parties in an election competing to win the election. Therefore, they are unlikely to collude. Let $\mathbb{G}$ be a publicly chosen multiplicative cyclic group of prime order p and g is a generator of $\mathbb{G}$.

A. Voter Registration

Voter registration is the initial phase where voters prove their voting eligibility to the registrar by providing the required evidence specified by the election. Once eligibility is validated, the registrar adds the voters to the electoral roll. Fig. 1 presents a summary of this process.

1) Registrar Signing Key Generation: The registrar requires a signing key pair to sign the identities of the eligible voters when adding them to the electoral roll. The registrar selects randomly a secret key $x_r \in \mathbb{Z}_p^*$ and computes its corresponding public key as $y_r = g^{x_r} \pmod p$.

2) Voter Key Generation and Registration: Voters are required to have voting key pairs corresponding to their identities for the election. A voter v_i selects a secret key $x_i \in \mathbb{Z}_p^*$ randomly and then computes the corresponding public key as $y_i = g^{x_i} \pmod p$. For registration, voters share their public key with the registrar.

3) Signing Voter's Public Key: To grant voting rights, the registrar first validates the eligibility of the voters then signs their public keys before adding them to the electoral roll. The registrar selects randomly u_i where $1 < u_i < p-1$ and $\gcd(u_i, p-1) = 1$ then computes the following:

$$w_i = g^{u_i} \pmod p, \tag{1}$$

$$s_i = (\mathsf{Hash}(y_i) - x_r w_i) u_i^{-1} \pmod p, \tag{2}$$

where Hash is a hash function and (w_i, s_i) is the signature. At the end of the registration phase, the registrar publicly discloses the electoral roll.

B. Acquiring a Ballot

Eligible voters acquire ballots needed to cast their votes toward the election. Fig. 2 summarizes this process.

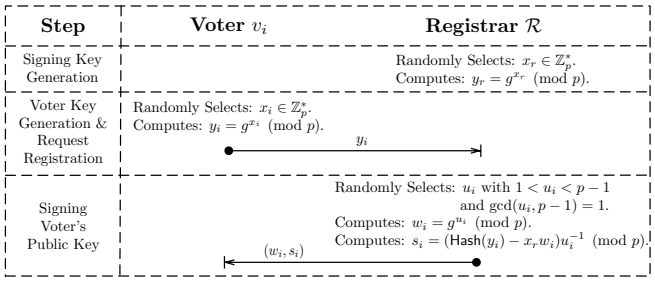

Fig. 1: A summary of the voter registration phase.

1) Ballots Generation: The registrar generates n unique and random digital ballots $\mathcal{T} = \{t_i \,|\, i \in \mathbb{Z}_n\}$ and digitally signs the ballots using ElGamal signature scheme. Denote the set as $\mathcal{B} = \{\mathsf{bal}_i = \mathsf{EG\text{-}Sign}(t_i) \,|\, i \in \mathbb{Z}_n\}$. Then it performs a permutation π on $\mathcal{B}$ such that:

$$\pi \colon \mathcal{B} \to \mathcal{B}.$$

2) Voter Permutation: The moderator acts as an intermediary that conceals the identities of voters from the registrar as they request ballots. The moderator generates a one time permuted set σ of n unique numbers used with every voter ballot request such that:

$$\sigma \colon \mathbb{Z}_n \to \mathbb{Z}_n.$$

3) Requesting a Ballot: Voters share their public keys with the moderator to prove their voting eligibility and get their ballots from the registrar who generates them.

4) Voter Identity Obscuring: Upon receiving the public key of a voter, the moderator checks whether the signature exists in the electoral roll and whether the signature (w_i, s_i) corresponding to the shared public key y_i is valid by verifying the following:

$$g^{\mathsf{Hash}(y_i)} \equiv y_i^{w_i}, w_i^{s_i} \pmod p, \tag{3}$$

holds true. In this event, the moderator obscures y_i by selecting randomly a blinding factor $b_i \in \mathbb{Z}_p^*$ and computing the following:

$$\begin{aligned} y_i' &= y_i^{b_i} \pmod p, \\ &= g^{x_i b_i} \pmod p. \end{aligned} \tag{4}$$

The moderator then sends y_i' and $\sigma(i)$ to the registrar.

5) Ballot Assignment and Encryption: The registrar assign ballots randomly to the anonymous voters upon receiving a ballot request from the moderator as follows:

$$\mathsf{bal}_i = \pi(\sigma(i)). \tag{5}$$

Prior to sharing bal_i with the moderator to pass along to the anonymous voter, the registrar encrypts it under an encryption key k_i derived as:

$$\begin{aligned} k_i &= (y_i')^{q_i} \pmod p \\ &= g^{x_i b_i q_i} \pmod p, \end{aligned} \tag{6}$$

where $q_i \in \mathbb{Z}_p^*$ is selected randomly. Using k_i, the registrar encrypts the ballot as the following:

$$\mathsf{ebal}_i = \mathsf{AES\text{-}Enc}_{k_i}(\mathsf{bal}_i), \tag{7}$$

where AES-Enc is the AES encryption function. The purpose of this encryption is to conceal the ballot from the moderator. It enables the registrar to share this ballot to the voter anonymously. For this purpose, it generates an ephemeral key Q_i that would allow the voter to regenerate k_i such that:

$$Q_i = g^{q_i} \pmod p. \tag{8}$$

The registrar then sends ebal_i and Q_i to the moderator.

6) Encrypted Ballot Transmission: Upon receiving ebal_i and Q_i, the moderator sends them to the voter along with the ciphertext eb_i

$$\begin{aligned} eb_i &= (g^{r_m}, b_i \cdot y_i^{r_m}) \\ &= (c_1, c_2) \end{aligned} \tag{9}$$

of the blinding factor b_i, where $r_m \in \mathbb{Z}_p^*$ is selected randomly.

7) Deriving Ballot: Once the voter receives eb_i from the moderator, it decrypts eb_i and recovers b_i as:

$$\begin{aligned} b_i &= c_2 \cdot c_1^{-x_i} \\ &= b_i \cdot y_i^{r_m} \cdot (g^{r_m})^{-x_i}. \end{aligned} \tag{10}$$

Next, the voter regenerates key k_i as:

$$\begin{aligned} k_i &= (Q_i)^{x_i b_i} \pmod p \\ &= g^{q_i x_i b_i} \pmod p. \end{aligned} \tag{11}$$

Finally, the voter decrypts ebal_i as:

$$\mathsf{bal}_i = \mathsf{AES\text{-}Dec}_{k_i}(\mathsf{ebal}_i), \tag{12}$$

where AES-Dec is the AES decryption function.

C. Casting Votes

A registered voter in possession of a ballot can submit a vote toward the election.

1) Ballot Double-Encryption: The voter first encrypts the ballot associated with a vote, denoted as B_i, under the public keys y_r and y_m of both the registrar and moderator as:

$$\begin{aligned} B_i &= (g^v, T \cdot (y_r \cdot y_m)^v) \\ &= (c_3, c_4), \end{aligned} \tag{13}$$

where $v \in \mathbb{Z}_p^*$ is selected randomly, $T = (\mathsf{bal}_i \,\|\, \mathrm{Vote})$, and $\mathrm{Vote} = (\mathsf{cand}_1, \mathsf{cand}_2, \ldots, \mathsf{cand}_m)$ is a sequence of bits representing each candidate such that:

$$\mathsf{cand}_k = \begin{cases} 1, & \text{if voting for } \mathsf{cand}_k, \\ 0, & \text{if voting against } \mathsf{cand}_k. \end{cases} \tag{14}$$

2) Submit Vote: The voter triggers the election vote smart contract $\mathsf{SC}_{\mathsf{vote}}$ and integrates B_i as input. Once the result of the smart contract is appended to the blockchain, the vote will be stored permanently.

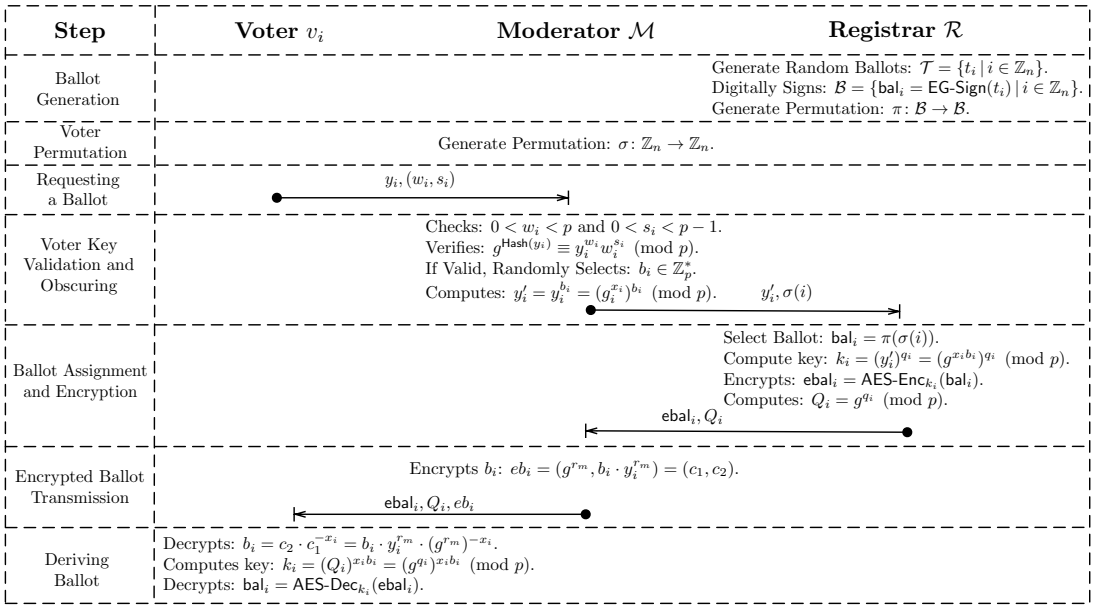

Fig. 2: A summary of the voter acquiring a ballot phase.

D. Tabulation

Tabulation occurs by the tallying authority once the casting votes phase is complete. Votes that appear on the blockchain within this phase are collected to be validated and counted. For validation, the moderator and registrar publicly disclose their secret keys x_m and x_r, respectively. The registrar also discloses a new permutation $\gamma \colon \mathcal{B} \to \mathcal{B}$ containing all ballots distributed among voters during the acquiring ballots phase. The tallying authority decrypts the attached ballots appearing on the blockchain as:

$$\begin{aligned} T &= c_4 \cdot c_3^{-x_m} \cdot c_3^{-x_r} \\ &= T \cdot (y_r \cdot y_m)^v \cdot (g^v)^{-x_m} \cdot (g^v)^{-x_r}. \end{aligned} \tag{15}$$

If $\mathsf{bal}_i \in \gamma$, the tallying authority examines the attached vote sequence and increments the counter of each candidate accordingly. Each election may specify its own rules that disqualify votes which are cast incorrectly. For example, voting *yes* for two opposing candidates.

E. Voter Verifiability

At the end of the tabulation phase, the tallying authority publishes the results of the election on the blockchain. Each vote is published with its corresponding ballot as proof of the legitimacy of the results. Voters can simply recognize their ballots and verify that their votes have been counted properly toward the election.

V. Security and Privacy Analysis

In this section, the security and privacy are analyzed.

A. Double-Voting Resistance

Elections running over i-voting schemes may vary in terms of voting policies. While some may allow voters to cast multiple votes using the same ballot and only count their last vote, others may allow voters to submit their votes only one time, disqualifying those with multiple attempts. In both cases, our proposed scheme is resistant to double-voting since it uses the blockchain as a bulletin board of stored votes. For the former, the tallying authority scans the blockchain for all cast votes that incorporate the same ballot, counting only the final vote. For the latter, the tallying authority disqualifies any votes that appear to incorporate the same ballot.

B. Coercion-Resistance

In order for voters to cast their votes, they must receive a valid ballot bal_i. However, in our proposed scheme, voters do not receive the ballots *directly*. In fact, they receive ebal_i, an encryption of the ballot under the key k_i. This means, a coercer would require the voter to decrypt ebal_i before being able to claim the voting credentials of the voter. A voter may simply cheat the coercer by decrypting ebal_i with k_i' then sharing the fake ballot bal_i'. It is infeasible for a probabilistic polynomial time coercer to distinguish between any given $(g, g^a, g^b, g^c, k_i = g^{abc})$ and $(g, g^a, g^b, g^c, k_i' = R)$, where $R \in \mathbb{Z}_p^*$ is a random element. An algorithm $\mathcal{A}$ that outputs a guess $z \in \{0, 1\}$ has advantage ε in solving the Decisional Diffie-Hellman (DDH) problem in $\mathbb{Z}_p^*$ if:

$$\begin{aligned} &\big| Pr[\mathcal{A}(g, g^a, g^b, g^c, k_i = g^{abc})] \\ &- Pr[\mathcal{A}(g, g^a, g^b, g^c, k_i' = R)] \big| \geq \varepsilon, \end{aligned} \tag{16}$$

for any randomly selected $a, b, c \in \mathbb{Z}_p^*$. The DDH assumption holds if no polynomial algorithm has a non-negligible advantage in solving the DDH problem. Therefore, our proposed scheme is coercion-resistant.

C. Election Result Manipulation

Cast votes are stored publicly and permanently over the blockchain during the casting vote phase. However, our

TABLE I: Number of multiplication and exponentiation operations at each phase.

	Voter Registration		**Acquiring Ballots**		**Casting Votes**		**Tabulation**	
Online \Offline	Offline		Online		Offline		Offline	
Operation	Mult	Exp	Mult	Exp	Mult	Exp	Mult	Exp
Voter	0	1	2	2	2	2	0	0
Moderator	0	0	$2n$	$6n$	0	0	0	0
Registrar	$2n$	$n+1$	0	$2n$	0	0	0	0
Tallying Authority	0	0	0	0	0	0	$2(n+\delta)$	$2(n+\delta)$

TABLE II: Average encryption and decryption costs per vote during vote casting and tabulation.

Key Size/bit	**512**		**1024**		**2048**		**4096**	
Function	Enc	Dec	Enc	Dec	Enc	Dec	Enc	Dec
Time/sec	0.0007	0.0012	0.002	0.0022	0.0108	0.0126	0.0761	0.0832

proposed scheme requires voters to encrypt their votes under the public keys x_r and x_m of the registrar and moderator. As a result, the votes are concealed throughout the entire phase. Voters that purposely delay casting their votes until the last minute would not be able to learn any information about the election results. Therefore, last-minute voters would not be able to manipulate the election results in a close race by voting in favor of a certain candidate.

D. Voter Anonymity

Our proposed scheme preserves the privacy of voters through a secure multi-party computation performed by parties of different allegiances. To ensure this property, at least two conflicting parties are required to participate during the ballot distribution process. As shown in the acquiring ballot phase, a moderator, representing one of the parties, conceals the public key y_i of the voter using a blinding factor b_i and associates it with a random value selected from the permutation $\sigma(i)$. On the other hand, the registrar, acting as the second party, selects a ballot randomly and assigns it to the anonymous voter. In order for ballots to be linked to the identities of voters, the moderator and registrar must collude. However, collusion would not be in the best interest of any of these parties, therefore, voter anonymity is preserved.

VI. Performance Evaluation

The performance of our proposed scheme can be measured by calculating the number of multiplication and exponentiation operations. Table I summarizes the number of operations performed by every actor at each phase. As shown in the table, all operations are performed offline except for the acquiring ballots phase which requires the computations to be done online. This greatly reduces the overall operational costs when running an election. We also note that the tallying authority would have to decrypt all votes that appear in the blockchain during the casting votes time frame including the fake ones, which we denote as δ.

To further investigate the performance, we calculate the average time costs for voters to encrypt their votes B_i and for the tallying authorities to decrpt a single vote. All measurements were performed using Maple v16 on a MacBook Air running OS X 10.13.6 equipped with 2 cores, 1.8 GHz Intel Core i5, and 8 GB 1600 MHz DDR3. Table II outlines these time costs of ten trials under four different encryption key sizes.

VII. Conclusion

In this paper, we proposed a novel distributed i-voting scheme that is resistant to double-voting and election result manipulation. Our proposed work leverages a blockchain as a publicly accessible bulletin board. We proved that voter anonymity is preserved and voters can combat coercers trying to claim their voting credentials. In our performance evaluation, we also showed that the overall computational expenses are reasonable to apply our proposed scheme in large-scale elections. In future work, we intend on formalizing the security and privacy analysis.

References

[1] E-Estonia, "i-voting e-estonia." https://e-estonia.com/solutions/e-governance/i-voting, 2019. Accessed: 04-17-2019.

[2] A. Juels, D. Catalano, and M. Jakobsson, "Coercion-resistant electronic elections," in *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pp. 61–70, ACM, 2005.

[3] M. R. Clarkson, S. Chong, and A. C. Myers, "Civitas: Toward a secure voting system," in *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pp. 354–368, IEEE, 2008.

[4] B. Adida, "Helios: Web-based open-audit voting," in *USENIX security symposium*, vol. 17, pp. 335–348, 2008.

[5] K. Sako and J. Kilian, "Receipt-free mix-type voting scheme," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 393–403, Springer, 1995.

[6] D. Demirel, J. Van De Graaf, and R. S. dos Santos Araújo, "Improving helios with everlasting privacy towards the public," *EVT/WOTE*, vol. 12, 2012.

[7] J. Clark and U. Hengartner, "Selections: Internet voting with over-the-shoulder coercion-resistance," in *International Conference on Financial Cryptography and Data Security*, pp. 47–61, Springer, 2011.

[8] O. Spycher, R. Koenig, R. Haenni, and M. Schläpfer, "A new approach towards coercion-resistant remote e-voting in linear time," in *International Conference on Financial Cryptography and Data Security*, pp. 182–189, Springer, 2011.

[9] M. Schläpfer, R. Haenni, R. Koenig, and O. Spycher, "Efficient vote authorization in coercion-resistant internet voting," in *International Conference on E-Voting and Identity*, pp. 71–88, Springer, 2011.

[10] G. Tsoukalas, K. Papadimitriou, and P. Louridas, "From helios to zeus," *USENIX Journal of Election Technology and Systems (JETS)*, vol. 1, no. 1, pp. 1–17, 2013.

[11] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in *International Conference on Financial Cryptography and Data Security*, pp. 357–375, Springer, 2017.

[12] F. Hao, P. Y. Ryan, and P. Zieliński, "Anonymous voting by two-round public discussion," *IET Information Security*, vol. 4, no. 2, pp. 62–67, 2010.