Information Theoretically Private and Secure Distributed Voting Without a Trusted Authority

Seyed Reza Hoseini Najarkolaei†, Nargess Kazempour†, Mohammad Reza Aref†, Deniz Gunduz*
†Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran
*Information Processing and Communications Lab, Imperial College London, UK

***Abstract*—In this paper, we present a private voting system that consists of N voters who may vote to one of the K candidates or vote abstain. Each voter wants to compute the final tally, while staying private and robust against malicious voters, who try to gain information about the vote of the other voters beyond the final result, or send incorrect information to affect the final tally. We design an information-theoretic voting system that uses verifiable secret sharing and multi-party computation, which is secure and private as long as there are up to $\lfloor \frac{N-1}{3} \rfloor$ malicious voters.**

***Index Terms*—Private voting, multi-party computation, secret sharing.**

I. Introduction

Nowadays, with developments in technology, using electronic voting (e-voting) as an alternative to traditional paper voting has become common place, which is more efficient considering time and resources. Thanks to this efficiency, e-voting is now widely used in many different domains, for example, in multi-agent planning [1], federated learning [2], and collaborative filtering [3]. However, due to the lack of centralized vote collection and counting mechanism, distributed e-voting algorithms impose additional requirements compared to conventional paper voting.

The first e-voting protocol was proposed in 1981 by Chaum [4], which uses some trusted mixers to provide privacy of votes and digital pseudonym to preserve anonymity. Another tool that has been used in the e-voting systems is the blind signature [5]–[7]. The basis of the protocols using blind signature is that the authority signs the ballots blindly, and then each voter publishes its ballot through an anonymous channel, and thus, privacy is preserved. Some of the e-voting protocols use mix-nets to satisfy privacy [4], [8], [9]. These protocols use shuffle agents to mix the votes; and therefore, the authority is unable to find the relationship between the voter and the vote. There are also e-voting protocols that use the features of homomorphic encryption, which enables voters to do the computations over the encrypted data without access to the secret key [10]–[12]. With the recent interest in blockchain technology, it has been widely used in e-voting to provide security [13]–[15]. D-DEMOS [16] is another kind of secure e-voting protocol that is private under decisional Diffie-Hellman assumption that achieves end-to-end verifiability in the standard model. The protocol proposed in [17], also relies on the decisional Diffie-Hellman assumption. In [18], an unconditionally verifiable e-voting system is proposed, such that privacy is preserved under the decision linear assumption.

To preserve privacy, traditional e-voting protocols usually employ a centralized authority to collect the votes, which results in a single point of failure [4], [19]. Some protocols addressed this issue by replacing the trusted authority with a set of distributed authorities, and privacy is preserved if less than or equal to a certain number of them are malicious [20]–[23]. Many multi-authority e-voting protocols use non-interactive zero-knowledge proofs (NIZK) to provide verifiability [24]–[27]. The security of NIZK is based on a common reference string, which needs a trusted third party, or a random oracle, which has been shown to be unsound in [18].

Most of these protocols are cryptographic, that is, they provide privacy and security guarantees against adversaries with bounded computing power. If sufficient computing power becomes available in the future, e.g., with quantum computing, these schemes may be broken. As an alternative, information-theoretically secure e-voting protocols have been developed to preserve privacy against adversaries with unbounded complexity. The first such protocols were proposed in [28] and [29]. Information-theoretic secure voting protocols have also been presented in [19]–[23]. The scheme in [23] is based on Shamir secret sharing (SSS) [30], where each node shares its vote among a number of collection centers and they compute the sum of their shares in collaboration. This scheme is vulnerable to attacks by a malicious voter, which can send an invalid vote to make the final result incorrect, or vote for multiple candidates. In [19], a secure voting scheme based on a trusted authority is proposed. However, if this authority is compromised, the whole algorithm fails. Another information-theoretic private protocol is proposed in [20], which needs a trusted third party to distribute private keys between voters before the execution of the algorithm. The scheme in [21] uses a simultaneous broadcast channel, but may fail due to a corrupted participant. A fully private voting system is presented in [22]; however, it cannot tolerate Byzantine nodes.

In this paper, we propose a completely distributed and secure voting system with information-theoretic privacy guarantees, which does not rely on a trusted third party, or on pre-shared common information. We consider a private voting problem consisting of N authorized voters and K candidates. Each voter can vote for one of the candidates or abstain. The voters are interested in computing the final tally in a reliable manner, through interactions with each other, in the absence of a trusted authority. We further assume that up to T of the voters can be malicious; that is they may not follow the prescribed protocol in order to gain information about the vote of the other voters beyond the final result or to send incorrect information to affect the final tally beyond what is possible by their own vote. The objective is to propose a scheme such that the voters can all recover the final tally correctly and privately in the presence of such malicious voters.

To be reliable, secure and practical, the proposed scheme must satisfy some essential properties listed below:

1) **Correctness:** All valid votes must be counted correctly

and exactly once while the detected invalid votes should be ignored, and the final result must be effectively the real tally of the casted votes. Each voter should be able to to learn this result at the end of the algorithm.
2) **Unconditional privacy:** If any subset of T voters collude, they cannot gain any information beyond the result.
3) **Verifiability:** Each voter should be able to verify that her vote is counted correctly.
4) **Robustness:** Any adversarial behavior of at most T of the voters can be tolerated. No adversarial treatment can disrupt the voting and any cheating behavior will be detected or corrected.
5) **Non-reusability:** Each voter must be able to vote exactly once, and no voter can vote more than once.
6) **Fairness:** No one can achieve any information about the tally result before counting.

To the best of our knowledge, the proposed scheme is the first information-theoretically private voting system that satisfies all the above properties without relying on a trusted third party. One of the main challenges that we are faced in private voting is verifying the validity of the votes without gaining any information about it, while the other conditions are satisfied. To solve this problem, the proposed scheme exploits verifiable secret sharing (VSS) [31] and multi-party computation (MPC) [32] schemes. Using VSS enables voters to share their votes as a secret so that the privacy of their votes is preserved while the other voters can verify the consistency of the distributed shares. On the other hand, the MPC scheme is utilized in a way to make the voters sure that the votes are valid (a valid vote must correspond to exactly one candidate or abstain) and counted correctly, while providing robustness against malicious acts. The proposed voting scheme satisfies the aforementioned properties as long as $N \geq 3T + 1$.

The rest of the paper is organized as follows. In Section II, we introduce the problem setting. The main result is presented in Section III. Preliminaries are provided in Section IV. We illustrate the motivating example in Section V, and the proposed scheme is presented in Section VI. We conclude the paper in Section VII.

Notation: In this paper vectors are shown by boldface letters. We show the element-wise multiplication of two vectors $\mathbf{A}$ and $\mathbf{B}$ by $\mathbf{A}*\mathbf{B}$. $\mathbf{e}_k$ is a vector in $\mathbb{F}^n$ whose components are all zero, except the k-th one that is equal to 1. For each $N \in \mathbb{N}$, $[N]$ represents the set $\{1, 2, ..., N\}$ and $X_{[N]} = \{X_1, X_2, \ldots, X_N\}$. Also, for each $\mathbf{V} \in \mathbb{F}^n$, Sum($\mathbf{V}$) is is defined as $\sum_{i=1}^{n} V_i$. In addition, $\mathbf{1}_n$ is a vector in $\mathbb{F}^n$ whose components are all one and similarly, $\mathbf{0}_n$ is a null vector in $\mathbb{F}^n$.

II. Problem Setting

The private voting system (PVS) under consideration in this paper consists of N authorized voters and K candidates, $\mathcal{C} = \{C_1, C_2, ..., C_K\}$. Voter n can vote for one of the candidates, or abstain, which is shown by $\mathbf{V}^{(n)} \in \{0, 1\}^{K+1}, \forall n \in [N]$. $\mathbf{V}^{(n)}$ is a one-hot vector such that if voter n votes for C_k, then $\mathbf{V}^{(n)}$ is equal to $\mathbf{e}_k$, and $\mathbf{V}^{(n)} = \mathbf{e}_{K+1}$ if voter n votes abstain. There is no centralized authority to collect or count the votes, and voting is carried out through pair-wise communications among the voters who send functions of their votes to the other voters. The objective is, for each voter to be able to compute the final result of voting $\mathbf{R} = [R_1, R_2, \ldots, R_{K+1}]^T$, where R_k is the tally of casted votes corresponding to candidate C_k, $\forall k \in [K]$, and R_{K+1} shows the number of abstained voters. Also, assume that up to T of the voters are malicious. The malicious voters may send incorrect data to the other voters to affect the final result of voting. Besides, the malicious voters may try to violate the privacy constraint by acquiring information about the votes of the other voters. To achieve their goals, the malicious voters can collude; that is, share their data with each other, or deviate from the protocol. Note that the voters do not know in advance which of them are malicious. Therefore, the main challenges for the voters are to verify the validity of the votes and compute the final result correctly in the presence of the malicious voters, while keeping their votes private.

It is assumed that each pair of the voters are connected to each other with a point-to-point private link. Also, there is an authenticated broadcast channel among all the voters such that the identity of the broadcaster is known. All of these links are assumed to be error-free and secure. The proposed scheme consists of 3 steps:

1) **Sharing**: In this step, each voter $n \in [N]$ shares its vote, i.e., it sends a function of $\mathbf{V}^{(n)}$ to all of the other voters. Let $\mathcal{S}_{n,n'} \triangleq \mathbf{F}_{n,n'}(\mathbf{V}^{(n)}), \forall n' \in [N], n \neq n'$, be the set of messages that voter n' receives from voter n in this step, where $\mathbf{F}_{n,n'} : \mathbb{F}^{K+1} \to \mathbb{F}^{p \times q}$, for some $p, q \in \mathbb{N}$. Let us also define $\mathcal{S}_n \triangleq \cup_{n'=1}^{N} \mathcal{S}_{n',n}$ as the set of all messages voter n receives in this step.
2) **Verification**: In this step, the voters process their input messages from the previous step and communicate with each other to be able to verify the validity of each vote. A vote is valid if it is compatible with the voting system being used, e.g., the vote does not contain additional and surplus entries by the voter or more choice than permitted (overvoting). In this step, any adversarial behavior should be detected, corrected, or dropped. Let $\mathcal{M}_{n,n'}$ be the set of all messages that voter n' receives from voter n in the verification step, $\forall n, n' \in [N], n \neq n'$; and $\mathcal{M}_n \triangleq \cup_{n'=1}^{N} \mathcal{M}_{n',n}$ is the set of all messages voter n receives in this step.
3) **Counting**: After the verification step, each voter $n \in [N]$, broadcasts a message $\mathcal{B}_n$ to all the other voters.

The six requirements of a secure and private voting system that were specified in the previous section can be written in a more mathematical form as follows:

1- Correctness: All valid votes must be counted correctly, i.e., after the execution of the proposed algorithm, each voter must have sufficient information to be able to derive the final result $\mathbf{R}$ correctly; that is, $H(\mathbf{R}|\mathcal{S}_n, \mathcal{M}_n, \mathcal{B}_{[N]}) = 0, \forall n \in [N]$. Note that the correctness condition must be satisfied in the presence of at most T malicious voters.

2- Unconditional privacy: If any arbitrary subset $\mathcal{X}$ of at most T voters collude, they cannot gain any additional information about the votes of the other voters beyond the tally result $\mathbf{R}$. It means that, for any $\mathcal{X} \subset [N], |\mathcal{X}| \leq T$, and each $n \in [N] \backslash \mathcal{X}$, $H(\mathbf{V}^{(n)}|\mathbf{R}, \mathcal{S}_\mathcal{X}, \mathcal{M}_\mathcal{X}, \mathcal{B}_{[N]}) = H(\mathbf{V}^{(n)}|\mathbf{R})$.

3- Verifiability: Each voter must be able to verify that his vote correctly included in the tally result.

4- Robustness: A voter's vote cannot be changed, duplicated, or removed by malicious voters. Any adversarial behavior of at most T of the voters, must be tolerated; that is, no adversarial

behavior should be able to disrupt the voting and any cheating behavior must be detected or corrected.

5- Non-reusability: Each voter must be able to vote exactly once, and no voter can vote more than once.

6- Fairness: No voter can gain any information about the tally result except their own vote before the counting phase, i.e., $H(\mathbf{R}|\mathcal{S}_n, \mathcal{M}_n) = H(\mathbf{R}), \forall n \in [N]$.

III. Main Result

The objective of PVS is to derive the tally result correctly while guaranteeing privacy, correctness and robustness against adversaries. We propose a new private voting scheme explained in Section VI without any third party, which is unconditionally private, robust against adversarial behavior, and satisfies all of the conditions in Section II. The main result is stated in the following theorem.

Theorem 1. *Given K candidates and N voters, up to T of which are malicious. There exists a private voting scheme without a trusted third party that enables unconditionally private voting, while guaranteeing correctness, robustness, verifiability, non-reusability, and fairness conditions as defined in Section II, as long as $N \geq 3T + 1$.*

Remark 1: The proposed scheme, whose details are provided in Section VI uses VSS and MPC. VSS enables voters to share their votes as a secret such that the privacy of the votes is preserved and the other voters can verify consistency of the distributed shares. MPC is used in a way to enable each voter to verify the validity of the votes, while providing robustness against malicious acts.

Remark 2: The minimum number of voters needed depends linearly on the number of malicious voters with a coefficient 3. The upper bound $3T + 1$ is a common phenomenon in distributed computation with malicious nodes. Also, one can see that the number of candidates K, has no impact on on this condition.

Remark 3: In the proposed framework, the voters perform all the computing, and each of them can compute the final result, i.e., the voting is performed completely among the group of voters by secure pair-wise links and a public broadcast channel. Another setting that can be considered for the voting problem, is that there exist worker nodes and an authority in addition to the voters. In this formation, the voters only send their shares and required data to the worker nodes, and they perform the computing and send the required information to compute the final result to the authority. As long as a certain number of the workers collude, the privacy is preserved and the workers cannot gain any information about the votes of the voters. Using the received information, the authority can derive the final tally while gaining no information about the votes beyond the result. This framework of the voting system can be handled with a slight modification in our proposed scheme.

IV. Preliminaries

Before describing the achievable scheme, we need some preliminaries.

A. Polynomial Interpolation and Reed-Solomon Codes

Constructing a polynomial that passes through a given set $\mathcal{S}$ of points is called polynomial interpolation. Lagrange theorem, stated as Theorem 2 below, is used to identify the minimum-degree polynomial that goes through the points in $\mathcal{S}$.

Theorem 2 (Lagrange Theorem)**.** *Assume that $x_1, x_2, \ldots, x_{T+1}$ are distinct elements of $\mathbb{F}$ and $y_1, y_2, \ldots, y_{T+1}$ are elements of $\mathbb{F}$ (not necessarily distinct). There exists a unique polynomial $p(x)$ of degree at most T, such that $p(x_i) = y_i$, $\forall i \in [T + 1]$.*

A natural consequence of Lagrange Theorem is that any polynomial of degree T can be uniquely represented by $T + 1$ distinct points that lie on it.

Remark 4: Suppose that $x_1, x_2, \ldots, x_N$ are distinct elements of $\mathbb{F}$ and $y_1, y_2, \ldots, y_N$ are elements of $\mathbb{F}$ (not necessarily distinct). Also, assume that $N - E$ elements of the set $\mathcal{P} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ lie on a polynomial $p(x)$ of degree T while the remaining E points, corresponding to erroneous data, do not lie on $p(x)$. Reed-Solomon decoding [33] guarantees that $p(x)$ can be reconstructed by using the points of the set $\mathcal{P}$ as long as $E \leq \lfloor \frac{N-T-1}{2} \rfloor$. As a result, if the number of errors is less than T, Reed-Solomon decoding procedure guarantees that polynomial $p(x)$ of degree T can be reconstructed uniquely by using elements of set $\mathcal{P}$ as long as $N \geq 3T + 1$.

B. Verifiable Secret Sharing (VSS)

Assume that a node in a system, called as *the dealer* wants to share a secret $s \in \mathbb{F}$ with other nodes, such that any fewer than or equal to T colluding nodes cannot gain any information about s, while any subset of the nodes more than T, can recover it perfectly. Secret sharing was first introduced by Shamir [30] and Blakley [34], independently, in 1979. It is a basic tool in cryptography and has been used in many applications such as e-voting schemes, crypto-currencies, and access control systems. In the Shamir secret sharing, the dealer constructs a polynomial $f(x) = s + c_1 x + c_2 x^2 + \cdots + c_T x^T$ of degree T such that $f(0)$ is equal to the secret and the other coefficients are chosen uniformly and randomly from field $\mathbb{F}$. Each participant n is assigned a distinct and nonzero α_n, chosen independently and uniformly at random from $\mathbb{F}$. The dealer sends $f(\alpha_n)$ to participant n, $\forall n \in [N]$. One can see that any arbitrary subset $\mathcal{X}$ of at least $T + 1$ participants can find the secret s in collaboration with each other, but if the size of $\mathcal{X}$ is at most T, they cannot gain any information about the secret. It is shown that this scheme is information-theoretically secure [30]. In this scheme, we assume that the dealer is trusted and always sends *consistent* shares to the other nodes, i.e., it chooses points on a polynomial of degree T.

On the other hands, the dealer may be malicious and sends non-consistent shares to the other nodes. In such a case, we need a mechanism that is able to verify the consistency of the shares. Chor et al. [31] introduce VSS, which enables nodes to verify whether their shares are consistent or not. In the context of Shamir secret sharing, VSS has the following properties:

- If the dealer is malicious, and the shares that it sends to the other nodes are not consistent, i.e., are not on a polynomial of specified degree, then the honest nodes in collaboration with each other will realize that and reject the shares.
- If the dealer is honest, then the malicious node cannot deceive the honest nodes and convince them that the dealer is malicious; thus, each honest node accepts its share.

In the original form of VSS [31], to share a secret $s \in \mathbb{F}$, the dealer chooses a bivariate polynomial $S(x,y)$, uniformly at random from the set of all bivariate polynomials of degree T, with respect to each of the variables x and y, with coefficients from $\mathbb{F}$, subject to $S(0,0) = s$. Then, the dealer sends $f_n(x) \triangleq S(x,\alpha_n)$ and $g_n(y) \triangleq S(\alpha_n,y)$ to node n, $\forall n \in [N]$, for some distinct $\alpha_n \in \mathbb{F}$. One can see that, $\forall n,n' \in [N]$, $f_n(\alpha_{n'}) = g_{n'}(\alpha_n)$. Therefore, the redundancy in this scheme allows the honest nodes to verify the consistency of shares through communication with other nodes. For detailed description of VSS, refer to [35]. By using VSS, in [32], an information-theoretic secure distributed multiplication protocol is proposed, called *BGW* algorithm. This scheme became the basis of many distributed secure computation algorithms such as large-scale matrix multiplication [36]–[39].

V. Motivating Example

For ease of understanding, we first illustrate the main idea of the proposed PVS through a simple example. Assume that in this election each voter n can vote only Yes or No. We denote the No and Yes votes by 0 and 1, respectively. The vote of voter n is denoted by $V^{(n)} \in \{0,1\}$, and the complement of $V^{(n)}$ is defined as $1-V^{(n)}$ and shown by $V'^{(n)}$. The steps of the proposed algorithm are as follows.

A. Sharing

In this step, each voter n shares both $V^{(n)}$ and $V'^{(n)}$ using the VSS algorithm [31]. In order to do that, voter n constructs two polynomials $F^{(n)}(x) = V^{(n)} + R_1^{(n)}x + R_2^{(n)}x^2 + \cdots + R_T^{(n)}x^T$ and $G^{(n)}(x) = V'^{(n)} + Z_1^{(n)}x + Z_2^{(n)}x^2 + \cdots + Z_T^{(n)}x^T$, and sends $F^{(n)}(\alpha_{n'})$ and $G^{(n)}(\alpha_{n'})$ to voter n', $\forall n,n' \in [N]$, $n' \neq n$, where $R_k^{(n)}$ and $Z_k^{(n)}$ are chosen uniformly and independently at random from field $\mathbb{F}$, $\forall k \in [T]$. Also, distinct and non-zero $\alpha_1,\alpha_2,\ldots,\alpha_N$ are chosen uniformly and independently at random from field $\mathbb{F}$ and they are known by all the voters.

If $N \geq 3T+1$, using VSS [31] ensures the voters that the shared values by voter n are consistent, i.e., they lie on a polynomial of degree T, otherwise, honest voters can identify malicious voters who have adversarial behavior and omit them from the remaining part of the algorithm. It addition, if voter n is honest, the malicious voters cannot gain any information about $F^{(n)}(0)$ and $G^{(n)}(0)$ except that one of them is 0 and the other one is 1, but they cannot understand which is which.

B. Verification

In this step, each voter needs to be assured that $F^{(n)}(0) = V^{(n)}$ is equal to 1 or 0. In order to do this, we perform 2-phase verification. In the first phase, called verification of summation, voters verify whether $F^{(n)}(0)+G^{(n)}(0)$ is equal to 1 or not, and in the second phase, called verification of product, they verify whether $F^{(n)}(0)G^{(n)}(0)$ is equal to 0 or not. If both of the aforementioned conditions are satisfied, then we can conclude that $\{F^{(n)}(0),G^{(n)}(0)\} = \{0,1\}$.

*1) **Verification of summation**:* Let us define $S^{(n)}(x) \triangleq F^{(n)}(x) + G^{(n)}(x)$. In this phase, $\forall n,n' \in [N]$, each voter n' broadcasts $S^{(n)}(\alpha_{n'}) = F^{(n)}(\alpha_{n'}) + G^{(n)}(\alpha_{n'})$. If all of the voters were honest, after this phase each voter has access to $\{S^{(n)}(\alpha_1),S^{(n)}(\alpha_2),\ldots,S^{(n)}(\alpha_N)\}$. But, some of the voters can be malicious and not broadcast correct information. One can see that $\deg(S^{(n)}(x)) = T$, thus, due to Remark 4, voters can correct up to $\lfloor \frac{N-T-1}{2} \rfloor$ errors. Since the number of malicious voters is at most T, we need to have $\lfloor \frac{N-T-1}{2} \rfloor \geq T$, or equivalently, $N \geq 3T+1$. If $N \geq 3T+1$, each voter can recover the correct set of $\{S^{(n)}(\alpha_1),S^{(n)}(\alpha_2),\ldots,S^{(n)}(\alpha_N)\}$. Thus, each voter can calculate $S^{(n)}(x)$, and then derive $S^{(n)}(0) = F^{(n)}(0)+G^{(n)}(0)$, and verify whether $F^{(n)}(0)+G^{(n)}(0)$ is equal to 1 or not, $\forall n \in [N]$.

*2) **Verification of product**:* In this phase, each voter needs to verify whether $F^{(n)}(0)G^{(n)}(0)$ is 0 or not, $\forall n \in [N]$. For this purpose, we use the *BGW* scheme for the secrets multiplication as explained in [35]. To be self-contained, the following is a brief overview of the scheme.

Theorem 3. *[35, Subsection 6.6] For an arbitrary pair of polynomials $A(x)$ and $B(x)$, each of degree T, there exist T polynomials $O_1(x),O_2(x),\ldots,O_T(x)$ of degree T such that the degree of $A(x)B(x) - \sum_{i=1}^{T} x^i O_i(x)$ is at most T.*

According to Theorem 3, each voter n can find polynomials $O_1^{(n)}(x),O_2^{(n)}(x),\ldots,O_T^{(n)}(x)$, such that $\deg(F^{(n)}(x)G^{(n)}(x) - \sum_{i=1}^{T} x^i O_i^{(n)}(x)) \leq T$. Let us define

$$C^{(n)}(x) \triangleq F^{(n)}(x)G^{(n)}(x) - \sum_{i=1}^{T} x^i O_i^{(n)}(x). \tag{1}$$

One can see that $C^{(n)}(0) = F^{(n)}(0)G^{(n)}(0)$. This is due to the fact that each $O_i^{(n)}(x)$ is multiplied by x^i, $i \geq 1$. Thus, the constant term of $F^{(n)}(x)G^{(n)}(x)$ cannot be affected by $O_i^{(n)}(x)$, $\forall i \in [T]$. Constructing $C^{(n)}(x)$ enables the other voters to compute the value of $F^{(n)}(0)G^{(n)}(0)$ without violating the privacy, i.e., malicious voters cannot get any additional information about the polynomials $F^{(n)}(x)$ and $G^{(n)}(x)$.

After constructing $O_1^{(n)}(x),O_2^{(n)}(x),\ldots,O_T^{(n)}(x)$, voter n shares $O_i^{(n)}(x)$ with all the other voters by using the VSS algorithm, i.e., it sends $O_i^{(n)}(\alpha_{n'})$ to voter n', $\forall n,n' \in [N]$, and $\forall i \in [T]$. In addition, voter n shares $C^{(n)}(x)$, i.e., it sends $C^{(n)}(\alpha_{n'})$ to voter n', $\forall n,n' \in [N]$. Until now, voter n' has the values of $C^{(n)}(x),F^{(n)}(x),G^{(n)}(x),O_i^{(n)}(x)$ at point $\alpha_{n'}$, $\forall n,n' \in [N]$ and $\forall i \in [T]$. Hence, each voter n' can directly verify whether (1) holds for $\alpha_{n'}$ or not. If (1) does not hold for $\alpha_{n'}$, voter n' broadcasts a Complaint messages. Then the other voters compute the values $C^{(n)}(\alpha_{n'}),F^{(n)}(\alpha_{n'}),G^{(n)}(\alpha_{n'}),O_1^{(n)}(\alpha_{n'}),O_2^{(n)}(\alpha_{n'}),\ldots,O_T^{(n)}(\alpha_{n'})$ in collaboration with each other, to identify the malicious voter among voter n and voter n' and omit the malicious one from the remaining part of the algorithm. Please see [35] for more details.

Then, to verify $F^{(n)}(0)G^{(n)}(0) = 0$, each voter n' broadcasts $C^{(n)}(\alpha_{n'})$. Hence, each voter has access to the value of $C^{(n)}(x)$ at more than $3T$ points. Thus, due to Remark 4, each voter can compute $C^{(n)}(x)$ and verify if $C^{(n)}(0) = F^{(n)}(0)G^{(n)}(0)$ is equal to 0 or not, even if the malicious voters send incorrect data.

C. Counting

Assume that $\mathcal{I}$ is the set of all the malicious voters that are identified by the other voters. So far, each voter n'

has $F^{(n)}(\alpha_{n'})$, $\forall n, n' \in [N]\backslash\mathcal{I}$. Also, it is verified that $F^{(n)}(0) \in \{0,1\}$. Let us define $F(x) \triangleq \sum_{n\in[N]\backslash\mathcal{I}} F^{(n)}(x)$. Each voter n' computes $F(\alpha_{n'}) = \sum_{n\in[N]\backslash\mathcal{I}} F^{(n)}(\alpha_{n'})$ and broadcasts the result. Ideally, after this step, each voter has access to $\{F(\alpha_1), F(\alpha_2), \ldots, F(\alpha_N)\}$. But, some of the voters may act maliciously. Since $\deg(F(x)) = T$, by using the Reed-Solomon decoding procedure, voters can correct up to $\lfloor \frac{N-T-1}{2} \rfloor$ errors. Since the number of malicious voters is at most T, we need to have $\lfloor \frac{N-T-1}{2} \rfloor \geq T$, or equivalently, $N \geq 3T+1$. If $N \geq 3T+1$, each voter can recover the correct set of $\{F(\alpha_i)\}_{i\in[N]\backslash\mathcal{I}}$, and calculate $F(x)$, and finally derive $F(0) = \sum_{n\in[N]\backslash\mathcal{I}} F^{(n)}(0) = \sum_{n\in[N]\backslash\mathcal{I}} V^{(n)}$, which is the total number of 1 votes casted excluding the votes of the identified malicious voters in set $\mathcal{I}$.

As described above, as long as $N \geq 3T+1$, the correctness and robustness properties are satisfied and the privacy is assured thanks to VSS and *BGW*. Also, each voter can vote once and all of them are sure that their votes are counted correctly, and none of them can gain any information about the tally result before the counting phase. The detailed proofs of these claims will be provided in the longer version [40].

VI. General Scenario

Consider now an election among K candidates $\mathcal{C} = \{C_1, C_2, ..., C_K\}$, where voter n may vote for one of the candidates or abstain, which is shown by $\mathbf{V}^{(n)} \in \{0,1\}^{K+1}$, $\forall n \in [N]$. The voters aim to compute the final result $\mathbf{R} = [R_1, R_2, \ldots, R_{K+1}]^T$, where R_k is the tally of votes casted for candidate C_k, $\forall k \in [K]$, and R_{k+1} shows the number of abstained votes. Also, assume that distinct $\alpha_1, \alpha_2, \ldots, \alpha_N$ are chosen uniformly and independently at random from field $\mathbb{F}$, which are known by all the voters.

In this section, we follow the same protocol as Section V with some modifications to handle more candidates.

A. Sharing

In this step, each voter n wants to share its vote $\mathbf{V}^{(n)}$, which is a one-hot vector in $\{0,1\}^{K+1}$. Let us define $\mathbf{V'}^{(n)}$ as the complement of $\mathbf{V}^{(n)}$, or equivalently, $\mathbf{V'}^{(n)} = \mathbf{1}_{K+1} - \mathbf{V}^{(n)}$, where $\mathbf{1}_{K+1} \triangleq [1, 1, \ldots, 1]^T$.

In this step, each voter n shares both $\mathbf{V}^{(n)}$ and $\mathbf{V'}^{(n)}$ using the VSS algorithm [31]. In order to do that, voter n constructs polynomials $\mathbf{F}^{(n)}(x) = \mathbf{V}^{(n)} + \mathbf{R}_1^{(n)}x + \mathbf{R}_2^{(n)}x^2 + \cdots + \mathbf{R}_T^{(n)}x^T$ and $\mathbf{G}^{(n)}(x) = \mathbf{V'}^{(n)} + \mathbf{Z}_1^{(n)}x + \mathbf{Z}_2^{(n)}x^2 + \cdots + \mathbf{Z}_T^{(n)}x^T$, then sends $\mathbf{F}^{(n)}(\alpha_{n'})$ and $\mathbf{G}^{(n)}(\alpha_{n'})$ to voter n', $\forall n, n' \in [N]$, where $\mathbf{R}_j^{(n)}$ and $\mathbf{Z}_j^{(n)}$ are chosen uniformly and independently at random from field $\mathbb{F}^{K+1}$, $\forall j \in [T]$.

Using VSS ensures the voters that if $N \geq 3T+1$, shared values by voter n are consistent, i.e., they lie on a polynomial of degree T, otherwise, honest voters can identify malicious voters who have adversarial behavior and omit them from the remaining part of the algorithm. It must be mentioned that if voter n is honest, the malicious voters cannot gain any information about $\mathbf{F}^{(n)}(0)$, which is the vote of voter n.

B. Verification

In this step, each voter wants to be assured that $\mathbf{V}^{(n)} = \mathbf{F}^{(n)}(0)$ is a one-hot vector. This would then verify that voter n followed the protocol and voted for exactly one candidate. In order to do that, we propose a 3-phase verification approach: 1) Verification of summation: All the voters verify whether $\mathbf{V}^{(n)} + \mathbf{V'}^{(n)} = \mathbf{1}_{K+1}$ or not, $\forall n \in [N]$. 2) Verification of product: All the voters verify whether $\mathbf{V}^{(n)} * \mathbf{V'}^{(n)}$ is equal to $\mathbf{0}_{K+1}$ or not, $\forall n \in [N]$. 3) Verification of entities: All the voters verify whether $\text{Sum}(\mathbf{V}^{(n)}) = \sum_{i=1}^{K+1} V_i^{(n)} = 1$, or not.

If the first two conditions are satisfied, then we can conclude that $\mathbf{V}^{(n)} \in \{0,1\}^{K+1}$. The last condition makes the other voters sure that $\mathbf{V}^{(n)} = [V_1^{(n)}, V_2^{(n)}, \ldots, V_{K+1}^{(n)}]$ is a one-hot vector. Thus, if all of the aforementioned conditions are satisfied, then we can conclude that the vote of voter n is valid, i.e., voter n votes for one candidate, or abstain, $\forall n \in [N]$.

*1) **Verification of summation**:* Similar to the verification of summation phase in Section V, each voter n' broadcasts $\mathbf{F}^{(n)}(\alpha_{n'}) + \mathbf{G}^{(n)}(\alpha_{n'})$, and finally all the voters can verify whether $\mathbf{F}^{(n)}(0) + \mathbf{G}^{(n)}(0)$ is equal to $\mathbf{1}_{K+1}$ or not, $\forall n \in [N]$.

*2) **Verification of product**:* Similar to the verification of product phase in Section V, the *BGW* scheme enables the voters to verify whether $\mathbf{F}^{(n)}(0) * \mathbf{G}^{(n)}(0)$ is $\mathbf{0}_{K+1}$ or not, $\forall n \in [N]$.

*3) **Verification of entities**:* Here, the aim is to verify that $\text{Sum}(\mathbf{F}^{(n)}(0)) = \text{Sum}(\mathbf{V}^{(n)}) = 1$. In order to do that, $\forall n, n' \in [N]$, each voter n' broadcasts $\text{Sum}(\mathbf{F}^{(n)}(\alpha_{n'}))$. If all of the voters were honest, after this phase each voter has access to $\{\text{Sum}(\mathbf{F}^{(n)}(\alpha_1)), \text{Sum}(\mathbf{F}^{(n)}(\alpha_2)) \ldots, \text{Sum}(\mathbf{F}^{(n)}(\alpha_N))\}$, which indeed lie on a $T-$degree polynomial $\text{Sum}(\mathbf{F}^{(n)}(x))$. But, some of the voters may act maliciously. As it is mentioned in Remark 4, voters can correct up to $\lfloor \frac{N-T-1}{2} \rfloor$ errors, or equivalently, if $N \geq 3T+1$, each voter can calculate $\text{Sum}(\mathbf{F}^{(n)}(0))$, and then derive $\text{Sum}(\mathbf{V}^{(n)})$. As a result, the other voters can verify that $\mathbf{F}^{(n)}(0) = \mathbf{V}^{(n)}$ is a one-hot vector.

C. Counting

Let us define $\mathbf{F}(x) \triangleq \sum_{n\in[N]\backslash\mathcal{I}} \mathbf{F}^{(n)}(x)$. Similar to the counting step in Section V, each voter n' computes $\mathbf{F}(\alpha_{n'}) = \sum_{n\in[N]\backslash\mathcal{I}} \mathbf{F}^{(n)}(\alpha_{n'})$ and broadcasts the result. As explained in Section V, after some computations, each voter can derive $\mathbf{F}(0) = \sum_{n\in[N]\backslash\mathcal{I}} \mathbf{F}^{(n)}(0) = \sum_{n\in[N]\backslash\mathcal{I}} \mathbf{V}^{(n)}$, which is equal to our final result $\mathbf{R} = [R_1, R_2, \ldots, R_{K+1}]^T$ without counting the votes of identified malicious voters in set $\mathcal{I}$.

As described in Section V as long as $N \geq 3T+1$, all of the conditions mentioned in Section II are satisfied.

VII. Conclusion

In this paper, we proposed an information-theoretically secure and private voting system that does not relay on a trusted authority. We exploited MPC and VSS schemes to detect, correct, or drop malicious voters. We showed that if the total number of voters is more than the three times of malicious voters, the system can guarantee that adversarial behavior cannot compromise the election result.

REFERENCES

[1] E. Ephrati and J. S. Rosenschein, "Deriving consensus in multiagent systems," *Artificial intelligence*, vol. 87, no. 1-2, pp. 21–74, 1996.

[2] J. Bernstein, J. Zhao, K. Azizzadenesheli, and A. Anandkumar, "SignSGD with majority vote is communication efficient and fault tolerant," *arXiv preprint arXiv:1810.05291*, 2018.

[3] D. M. Pennock, E. Horvitz, C. L. Giles, *et al.*, "Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering," *AAAI/IAAI*, vol. 30, pp. 729–734, 2000.

[4] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[5] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982* (D. Chaum, R. L. Rivest, and A. T. Sherman, eds.), pp. 199–203, Plenum Press, New York, 1982.

[6] A. Fujioka, T. Okamoto, and K. Ohta, "A practical secret voting scheme for large scale elections," in *Advances in Cryptology - AUSCRYPT '92, Workshop on the Theory and Application of Cryptographic Techniques, Gold Coast, Queensland, Australia, December 13-16, 1992, Proceedings* (J. Seberry and Y. Zheng, eds.), vol. 718 of *Lecture Notes in Computer Science*, pp. 244–251, Springer, 1992.

[7] S. Ibrahim, M. Kamat, M. Salleh, and S. R. A. Aziz, "Secure e-voting with blind signature," in *4th National Conference of Telecommunication Technology, 2003. NCTT 2003 Proceedings.*, pp. 193–197, IEEE, 2003.

[8] D. Boneh and P. Golle, "Almost entirely correct mixing with applications to voting," in *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002* (V. Atluri, ed.), pp. 68–77, ACM, 2002.

[9] R. Aditya, B. Lee, C. Boyd, and E. Dawson, "An efficient mixnet-based voting scheme providing receipt-freeness," in *Trust and Privacy in Digital Business, First International Conference, TrustBus 2004, Zaragoza, Spain, August 30 - September 1, 2004, Proceedings* (S. K. Katsikas, J. López, and G. Pernul, eds.), vol. 3184 of *Lecture Notes in Computer Science*, pp. 152–161, Springer, 2004.

[10] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," in *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding* (W. Fumy, ed.), vol. 1233 of *Lecture Notes in Computer Science*, pp. 103–118, Springer, 1997.

[11] S. S. Chow, J. K. Liu, and D. S. Wong, "Robust receipt-free election system with ballot secrecy and verifiability.," in *NDSS*, vol. 8, pp. 81–94, 2008.

[12] H. Li, Y. Sui, W. Peng, X. Zou, and F. Li, "A viewable e-voting scheme for environments with conflict of interest," in *2013 IEEE Conference on Communications and Network Security (CNS)*, pp. 251–259, IEEE, 2013.

[13] A. B. Ayed, "A conceptual secure blockchain-based electronic voting system," *International Journal of Network Security & Its Applications*, vol. 9, no. 3, pp. 01–09, 2017.

[14] A. Ometov, Y. Bardinova, A. Afanasyeva, P. Masek, K. Zhidanov, S. Vanurin, M. Sayfullin, V. Shubina, M. Komarov, and S. Bezzateev, "An overview on blockchain for smartphones: State-of-the-art, consensus, implementation, challenges and future trends," *IEEE Access*, vol. 8, pp. 103994–104015, 2020.

[15] U. C. Çabuk, E. Adiguzel, and E. Karaarslan, "A survey on feasibility and suitability of blockchain techniques for the e-voting systems," *arXiv preprint arXiv:2002.07175*, 2020.

[16] N. Chondros, B. Zhang, T. Zacharias, P. Diamantopoulos, S. Maneas, C. Patsonakis, A. Delis, A. Kiayias, and M. Roussopoulos, "D-demos: A distributed, end-to-end verifiable, internet voting system," in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, pp. 711–720, IEEE, 2016.

[17] S. J. Bouck, *A Verifiable Distributed Voting System Without a Trusted Party*. PhD thesis, Arizona State University, 2021.

[18] V. Iovino, A. Rial, P. B. Rønne, and P. Y. Ryan, "Universal unconditional verifiability in e-voting without trusted parties," in *2020 IEEE 33rd Computer Security Foundations Symposium (CSF)*, pp. 33–48, IEEE, 2020.

[19] Y. Liu and Q. Zhao, "E-voting scheme using secret sharing and k-anonymity," *World Wide Web*, vol. 22, no. 4, pp. 1657–1667, 2019.

[20] A. Otsuka and H. Imai, "Unconditionally secure electronic voting," in *Towards Trustworthy Elections*, pp. 107–123, Springer, 2010.

[21] A. Broadbent and A. Tapp, "Information-theoretic security without an honest majority," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 410–426, Springer, 2007.

[22] J. Van De Graaf, "Voting with unconditional privacy: CFSY for booth voting," *Cryptology ePrint Archive*, 2009.

[23] V. Binu, D. G. Nair, and A. Sreekumar, "Secret sharing homomorphism and secure e-voting," *arXiv preprint arXiv:1602.05372*, 2016.

[24] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," in *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pp. 329–349, 2019.

[25] C. Rackoff and D. R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," in *Annual International Cryptology Conference*, pp. 433–444, Springer, 1991.

[26] A. D. Santis, S. Micali, and G. Persiano, "Non-interactive zero-knowledge proof systems," in *Conference on the Theory and Application of Cryptographic Techniques*, pp. 52–72, Springer, 1987.

[27] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp. 62–73, 1993.

[28] J. Bos and G. Purdy, "A voting scheme," *Rump session of Crypto*, vol. 88, 1988.

[29] D. Chaum, "Elections with unconditionally-secret ballots and disruption equivalent to breaking rsa," in *Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 177–182, Springer, 1988.

[30] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[31] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults," in *Foundations of Computer Science, 1985., 26th Annual Symposium on*, pp. 383–395, IEEE, 1985.

[32] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pp. 1–10, 1988.

[33] S. B. Wicker and V. K. Bhargava, *Reed-Solomon codes and their applications*. John Wiley & Sons, 1999.

[34] G. R. Blakley, "Safeguarding cryptographic keys," in *Managing Requirements Knowledge, International Workshop on*, pp. 313–313, IEEE Computer Society, 1979.

[35] G. Asharov and Y. Lindell, "A full proof of the BGW protocol for perfectly-secure multiparty computation.," in *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 18, pp. 10–1007, 2011.

[36] S. R. H. Najarkolaei, M. A. Maddah-Ali, and M. R. Aref, "Coded secure multi-party computation for massive matrices with adversarial nodes," in *2020 Iran Workshop on Communication and Information Theory (IWCIT)*, pp. 1–6, IEEE, 2020.

[37] H. A. Nodehi, S. R. H. Najarkolaei, and M. A. Maddah-Ali, "Entangled polynomial coding in limited-sharing multi-party computation," in *2018 IEEE Information Theory Workshop (ITW)*, pp. 1–5, IEEE, 2018.

[38] H. A. Nodehi and M. A. Maddah-Ali, "Secure coded multi-party computation for massive matrix operations," *IEEE Transactions on Information Theory*, 2021.

[39] S. R. Hoseini Najarkolaei, M. A. Maddah-Ali, and M. R. Aref, "Coded secure multi-party computation for massive matrices with adversarial nodes," *arXiv e-prints*, pp. arXiv–2004, 2020.

[40] S. R. Hoseini Najarkolaei, N. Kazempour, D. Gunduz, and M. R. Aref, "Information theoretically private and secure distributed voting without a trusted authority," *under preperation*, 2022.