Blockchain-based Complete Self-tallying E-voting Protocol

Yikang Lin* and Peng Zhang*
* ATR Key Laboratory of National Defense Technology
College of Electronics and Information Engineering
Shenzhen University
Shenzhen, China
E-mail: linyikang2017@email.szu.edu.cn, zhangp@szu.edu.cn

***Abstract*—Electronic voting (E-voting) protocol is that voters can vote according to their wishes, and then the voting authority is responsible for collecting the votes and counting the final voting result. With the development of Blockchain, we tend to combine it with E-voting and propose Blockchain-based complete self-tallying E-voting protocol. Its distributed network makes the protocol more available than E-voting protocol based on centralized servers. In our protocol, Blockchain acts as bulletin board, and "Efficient One-out-of-T" zero knowledge proof (ZKP) is proposed to support multi-candidate voting. Moreover, the issues of abortive and adaptive are solved. The security analysis shows that our protocol meets the security requirements of E-voting, and it can be applied to small-scale and anonymous private scenario such as Corporate Board Voting. The performance analysis demonstrates that the proposed ZKP has low time consumption.**

I. Introduction

Voting (e. g., Corporate Board Voting, Country's Presidential Election) provides people an opportunity to express their opinions, which is of great significance for developing democratic society. E-voting [10], using electronic means to aid voting, is proposed for better performance on efficiency, security and fairness. In terms of security, in order to achieve verifiability of voting results as well as enhance anonymity of the voters, various cryptographic techniques like homomorphic encryption, zero knowledge proof (ZKP) and signature are usually applied in E-voting protocol. However, DEF CON [3], a security conference held in USA in 2017, proved that the US election's voting machines using cryptographic techniques could be broken in 90 minutes, with the result that the number of votes could be revised. It raises the public concerns on the security of E-voting [6][22], and it's a great challenge to design secure E-voting protocol.

Blockchain, as an emerging information technology, is regarded as a digital, decentralized and public ledger, where all transactions made by users are recorded in a public and secure way, without the control of a central entity. Blockchain was first introduced by Nakamoto [16] in 2008, and was described a peer to peer payment system Bitcoin, which allows E-cash transactions without relying on financial institutions. In 2014, Buterin proposed another Blockchain platform called Ethereum [1], where smart contract is designed automatically execute code on Blockchain, and enables interactions between end users and Blockchain.

The decentralized and distributed network of Blockchain has the following advantages for secure E-voting protocol: (1) Since the consensus mechanism of Blockchain makes all data is maintained and managed by public users, E-voting process can be executed without any privileged user. (2) Since Blockchain records are traceable and non-repudiable, each new voting record will be shared to other nodes in the whole network, and all nodes can receive the voting records and add the received records to the block. (3) Blockchain uses timestamps to provide proof of time. If a vote is fraudulent or tampering, it will be possible to backtrack the time and data of the fraudulent or tampering vote in Blockchain.

Recently, a number of E-voting protocols based on Blockchain have been developed by exploiting its inherent features. These protocols can be classified into three categories. (1) The E-voting protocols based on public chain: Due to the complete decentralization and low performance of the public chain, the E-voting protocols are suitable for small-scale voting scenario, such as Corporate Board Voting. Ref. [15] proposed a protocol and claimed it had maximum privacy for the voters. However, the protocol only supports two candidates voting, and raises abortive and adaptive issues. The protocol proposed in Ref. [25] requires a credible third party to ensure the privacy of voting. (2) The E-voting protocols based on alliance chain: On account of the partial decentralization and high performance in alliance chain, the E-voting protocols are suitable for large-scale voting scenario, like Country's Presidential Election. Ref. [24] proposed a protocol supporting multi-candidate voting, but it can not be self-tallying and requires a credible third party. (3) The E-voting protocols based on Blockchain in IoT (The Internet of Things): There are excellent performance in leader voting about wireless sensor networks based on Blockchain in IoT. Ref. [13] proposed an E-voting protocol supporting two candidates voting only, and can not solve abortive and adaptive issues fundamentally.

Our Contribution: In this paper, we discuss E-voting protocol based on Blockchain and propose a complete self-tallying E-voting protocol. Based on the E-voting protocol of McCorry et al. [15], this paper solves the following three problems: multi-candidate voting, abortive and adaptive issues, and completeness.

Multi-candidate voting. We propose an "Efficient One-out-of-T" ZKP to support multi-candidate voting without revealing any additional information about candidates.

Abortive and adaptive issues. Abortive issue that voter's abstention will lead to voting suspension is solved by introducing an option for abstention vote in "Efficient One-out-of-T" ZKP. Adaptive issue that the result of the vote can be known in advance by the last voter is solved by using Schnorr signature [19] for further encrypting the vote, so that no one can count the voting result in advance.

Completeness. Blockchain acts as a bulletin board, based on which we propose a complete self-tallying E-voting protocol. The completeness of our protocol is reflected in supporting multi-candidate voting, self-tallying and solving abortive and adaptive issues.

II. Related Work

According to the server platforms, E-voting protocols can be divided into two categories: centralized servers and Blockchain.

A. E-voting Protocols Based on Centralized Servers

E-voting protocols based on centralized servers mainly use the cryptographic techniques to ensure the privacy and robustness, including homomorphic encryption, zero knowledge proof and signature. The specific research status is described as follows:

Homomorphic encryption [18] is the feasibility to sum up data without decrypting them, i. e., without knowing the exact content of the data. Shinde et al. [20] proposed an E-voting protocol using homomorphic encryption, which allows the encrypted votes to be counted by any third party without leaking any information. However, this protocol does not support multi-candidate voting due to the low performance of homomorphic encryption and need trust centralized servers to record the voting result.

Zero knowledge proof [7] is essentially a protocol involving two or more parties, and a series of steps are required by two or more parties to complete a task. E-voting protocols using zero knowledge proof [2][17][23] prove that their votes are valid and unique without revealing any information about candidates. Nevertheless, it must ensure that ZKP is running correctly on the centralized servers.

Blind signature [5] is a form of digital signature in which the content of a message is blinded before it is signed. In the E-voting protocols using blind signature like [9][12], the tallying centre shows that the vote is from a valid voter, while the owner of the vote is not revealed. In such protocols, both the voter and the tallying center must trust the signer. If not, the signature scheme may stop working. Furthermore, linkable ring signature [14] is proposed to avoid untrusted signers. However, a certain number of voters are required to ensure their anonymity in the process of signing. For those E-voting protocols using signature, once the voting authority or centralized servers is compromised, the security and privacy can not be guaranteed.

B. E-voting Protocols Based on Blockchain

According to different application scenarios and user requirements, Blockchain can be roughly divided into three categories: public chain, private chain and alliance chain. At present, the researches on the E-voting protocol are mainly based on public chain, alliance chain and IoT.

E-voting protocols based on public chain. The public chain, represented by Bitcoin and Ethereum, is the most decentralized chain that can not be controlled by third parties, and everyone can read the data records and participate in the chain. That is to say, even the program developers have no right to interfere with the users, so that each participants (the nodes) can be freely in/out the network and perform the related operations. McCorry et al. [15] proposed an E-voting protocol using zero knowledge proof, based on Ethereum, and it was applied in small-scale voting scenarios. The advantage of this protocol is that it support self-tallying and no need credible third party. However, this kind of protocol only support two candidates voting and has abortive and adaptive issues. Zhu et al. [25] proposed another anonymous and decentralized E-voting protocol using blind signature and ring signature, based on Ethereum, and it was applied in small-scale voting scenarios. In addition, the protocol support multi-candidate voting and self-tallying. However, this kind of protocol need a credible third party for blind signature.

E-voting protocols based on alliance chain. The alliance chain is in between of the public chain and the private chain, which can achieve "partial decentralization". Compared with public chain, alliance chain has fewer nodes and faster transaction processing, however, the data on the alliance chain can only be read and modified by nodes on the alliance chain. Yu et al. [24] proposed an E-voting protocol using ring signature and homomorphic encryption, based on alliance chain, and it was applied in large-scale voting scenarios. The protocol supports multi-candidate and verifiable voting. However, this kind of protocol can not support self-tallying and need a credible third party for taking charge the process of the vote encryption and decryption. Therefore, the administrator can not disclose the secret key which is used for the voting process.

Blockchain-based E-voting protocols in IoT. In order to solve the issues of IoT about immeasurability and single-point-of-failure, Li et al. [13] proposed a Blockchain-based self-tallying E-Voting protocol in IoT. The protocol claimed to make a commitment for solving abortive issue and set time-locked for solving adaptive issue. However, in commitment phase, voter's abstention would lead to voting suspension, and time-lock means that users get the voting result only after a certain time, which would damage real-time voting for each voters. In addition, the protocol can not support multi-candidate voting.

III. Preliminaries

A. Zero Knowledge Proof

Zero knowledge proof [7] is an elegant technique to limit the amount of information transferred from the prover A to the

verifier B in a cryptographic protocol. The prover attempts to convince the verifier that the following NP (Non-deterministic Polynomial) statement is true, "there is $\widetilde{x}$ such that $\widetilde{y} = F(\widetilde{x})$ and $\widetilde{x}$ is a decommitment of commitment". If NP statement is false, the prover can not convince the verifier. If the NP statement is true, the prover can convince the verifier without leaking any information about $\widetilde{x}$. Assuming that the prover A and the verifier B are a pair of interaction Turing machines, we make the following definitions for ZKP. Let random variables $< A, B > (x)$ represent the output of interaction between A with B, where x is the input. Generally, the output comes in two forms: (1) $< A, B > (x) = 1$ indicates that B accepts the proof given by A. (2) $< A, B > (x) = 0$ indicates that B does not accepts the proof given by A.

There are two types of ZKP: interactive and non-interactive. Interactive ZKP performs both-way communication between the prover and the verifier. The prover A needs to accept the secret parameters from the verifier B during the proof process. Then A can pass the complete proof information to B for checking. On the contrary, non-interactive ZKP performs one-way communication between the prover and the verifier, but the prover and the verifier need to share random information.

Schnorr Zero Knowledge Proof [19] : By using Schnorr ZKP, the prover can prove the validity of its private key to the verifier without revealing any information about the private key. Suppose the prover has private key ε and its public key is g^{ε} where g is generator in finite cyclic group G. The prover sends g^{v} and $\varphi = v - \varepsilon\eta$ to the verifier where v is random integer and η is the value of hash function about g, g^{ε}, and g^{v}. The verifier can verify whether g^{v} and $g^{\varphi}g^{\varepsilon\eta}$ are equal. If yes, the verifier believes that the prover has valid private key.

B. E-voting Protocol Model

As shown in Fig. 1, E-voting protocol model mainly has four steps: SETUP, SIGNUP, VOTE, and TALLY, which are carried out by the following three partners.

Ethereum: Ethereum is a Blockchain platform supporting smart contract. It acts as a bulletin board for E-voting.

Administrator: Administrator administers E-voting, including establishing an eligible voter list, setting voting issues and voting time, and controlling the voting processes.

Voters: Voters register for E-voting and cast their votes in their own opinions.

E-voting protocol processes are described as follows.

SETUP: The administrator authenticates the voter identity and uploads the list of eligible voters to Ethereum. In addition, the administrator sets a timer list for voting timely.

$t_{finishRegistration}$: all voters must register their voting keys before this time.

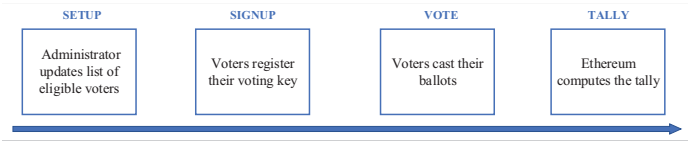

Fig. 1: E-voting protocol model

$t_{beginVoting}$: the administrator must notify Ethereum to begin the voting by this time.

$t_{finisnVoting}$: all voters must cast their votes before this time.

SIGNUP: All voters register for the voting after reviewing the parameters set by the administrator before $t_{finishRegistration}$. To register, the voter computes his voting key and detects the validity of the key by using ZKP. The administrator is responsible for notifying Ethereum to go to the next process VOTE.

VOTE: All voters publish their (encrypted) votes after $t_{beginVoting}$ and the corresponding ZKP showing the validity of the votes. The administrator notifies Ethereum to go to the next process TALLY when the final vote is voted before $t_{finishVoting}$ or $t_{finishVoting}$ is up.

TALLY: Ethereum compute the final voting result.

In addition, the ZKP in SIGNUP and VOTE are different. In SIGNUP, ZKP shows the validity of the key (See section III. A for details). In VOTE, ZKP (See seciton IV. A for details) shows that the vote is valid without revealing any information about the voter's choice.

IV. The Proposed Complete Self-tallying E-voting Protocol

In [15], McCorry proposed an E-voting protocol which only supports two candidates voting by using "One-out-of-Two" ZKP. In addition, the protocol has abortive and adaptive issues. In this section, we propose a complete self-tallying E-voting protocol. The protocol uses "Efficient One-out-of-T" ZKP to support multi-candidate voting in an efficient way. What's more, the issues of abortive and adaptive are solved.

Assume there are N voters and T candidates. In our voting, we assume the voter V_i votes for the candidate C_k. The voter V_i tries to use ZKP to convince others that the vote is credible. Some notations are defined in Table I.

TABLE I: Notations

Notation	Description
N	The number of voters
T	The number of candidates, $T \geq 3$
V_i	The i^{th} voter
x_i	The secret key for V_i
C_k	The k^{th} candidate
m_j	The value of $2^{qj} (j = k, \alpha, \beta, \gamma)$ for C_j, $2^q > N$
g	The generator in finite cyclic group G
h	The value $\Pi_{j=1}^{i-1} g^{x_j} / \Pi_{j=i+1}^{n} g^{x_j}$
H	Hash funciton where $\|\|$ represents the link symbol
X_i	The value of public key g^{x_i} for V_i
Y_i	The value of vote $h^{x_i} g^{m_k}$ for V_i

A. One-out-of-T and Efficient One-out-of-T ZKP

"One-out-of-T" ZKP is extended from "One-one-of-Two" ZKP [15], which proves that a candidate is one of all candidates without revealing any information about his identity. Based on it, to improve the algorithm performance, we propose "Efficient One-out-of-T" ZKP.

One-out-of-T ZKP: The prover creates random integer w, r_α, d_α ($\alpha = \{1, 2, \ldots, k-1, k+1, \ldots, T\}$). After that, the prover computes the vote $(X_i, Y_i) = (g^{x_i}, h^{x_i} g^{m_k})$, $(a_\alpha, b_\alpha) = (g^{r_\alpha} X_i^{d_\alpha}, h^{r_\alpha} (\frac{Y_i}{g^{m_\alpha}})^{d_\alpha})$ and $(a_k, b_k) = (g^w, h^w)$. And then, let $(a_j, b_j) = (g^{r_j} X_i^{d_j}, h^{r_j} (\frac{Y_i}{g^{m_j}})^{d_j})$, $j = 1, 2, \ldots, T$, and $c = H(x_i || X_i || Y_i || \{a_j, b_j\}_{j=1}^{T})$ for non-interaction.

Next, the prover computes $d_k = c - \Sigma_\alpha d_\alpha$ and $r_k = w - x_i \cdot d_k$. Finally, the prover sends $(\{a_j, b_j, d_j, r_j\}_{j=1}^{T}, X_i, Y_i)$ to the verifier for checking.

The verifier verifies correctness of $c = \sum_j d_j$ and (a_j, b_j) $=(g^{r_j} X_i^{d_j}, h^{r_j} (\frac{Y_i}{g^{m_j}})^{d_j})$ where $j = \{1, 2, \ldots, T\}$. If yes, the voting would continue. Otherwise, it fails.

Efficient One-out-of-T ZKP: We divided the prover and the verifier into two parts respectively according to the parity of k (even or odd).

The prover creates random integer w. Depending on k is even or odd, the prover creates random integer r_β, d_β ($\beta = \{2, 4, \ldots, k-2, k+2, \ldots, T\}$) or r_γ, d_γ ($\gamma = \{1, 3, \ldots, k-2, k+2, \ldots, T-1\}$) in group G. After that, the prover computes the vote $(X_i, Y_i) = (g^{x_i}, h^{x_i} g^{m_k})$ and $(a_k, b_k) = (g^w, h^w)$. And then, let $(a_j, b_j) = (g^{r_j} X_i^{d_j}, h^{r_j} (\frac{Y_i}{g^{m_j}})^{d_j})$ where $j = 2, 4, \ldots, T$ or $j = 1, 3, \ldots, T-1$. For non-interaction, let $c_{all} = H(x_i || X_i || Y_i)$, $c_{even} = H(x_i || X_i || Y_i || \{a_j, b_j\}_{j=2(j \in even)}^{T})$ or $c_{odd} = H(x_i || X_i || Y_i || \{a_j, b_j\}_{j=1(j \in odd)}^{T-1})$ when k is even or odd.

Next, the prover computes parameters based on the parity of k:

k is even: The prover computes $(a_\beta, b_\beta) = (g^{r_\beta} X_i^{d_\beta}, h^{r_\beta} (\frac{Y_i}{g^{m_\beta}})^{d_\beta})$, $c_{odd} = c_{all} - c_{even}$ and $d_k = c_{even} - \Sigma_\beta d_\beta$, $r_k = w - x_i \cdot d_k$.

k is odd: The prover computes $(a_\gamma, b_\gamma) = (g^{r_\gamma} X_i^{d_\gamma}, h^{r_\gamma} (\frac{Y_i}{g^{m_\gamma}})^{d_\gamma})$, $c_{even} = c_{all} - c_{odd}$ and $d_k = c_{odd} - \Sigma_\gamma d_\gamma$, $r_k = w - x_i \cdot d_k$.

Finally, the prover sends $(\{a_j, b_j, d_j, r_j\}_{j=2(j \in even)}^{T}, X_i, Y_i, c_{odd})$ or $(\{a_j, b_j, d_j, r_j\}_{j=1(j \in odd)}^{T-1}, X_i, Y_i, c_{even})$ to the verifier for checking.

The verifier verifies correctness of c_{even} or $c_{odd} = \sum_j d_j$, $c_{all} = c_{even} + c_{odd}$ and (a_j, b_j)=$(g^{r_j} X_i^{d_j}, h^{r_j} (\frac{Y_i}{g^{m_j}})^{d_j})$ where $j = \{1, 3, \ldots, T-1\}$ or $\{2, 4, \ldots, T\}$. If yes, the voting would continue. Otherwise, it fails.

B. Solution to Abortive and Adaptive Issues

In this part, we propose the solution to abortive and adaptive issues by using "Efficient One-out-of-T" ZKP and Schnorr signature [4], respectively.

Solution to abortive issue. The abortive issue is that if the voter abstains from vote, the voting will be suspended. McCorry et al. [15] sets a deposit for voting in order to solve abortive issue. However, it need an additional system budget and the voting will still be suspended when someone abstains from the vote, which reduces the robustness of the protocol. Therefore, we set candidate C_0 for abstention votes. Candidate C_0 is expressed by vote $m_0 = 2^0 = 1$, using "Efficient One-out-of-T" ZKP to ensure its privacy and security. In other words, there are $T+1$ candidates, and one of them is an abstention vote.

Solution to adaptive issue. The adaptive issue is that the result of the vote can be known in advance by the last voter. McCorry et al. [15] proposed an optional process that requires all voters to hash their encrypted vote and store it in Ethereum as a commitment. However, this method increases E-voting processes and complicates the protocol. Therefore, we make the voters transforms their vote Y_i to $Y_i' = Y_i g^e$ by using Schnorr signature [19] in order to further encrypt the vote in VOTE, where e is a random number and g^e is in group G. In TALLY, Y_i' can be decoded to Y_i and everyone can check the validation of g^e.

C. Complete Self-tallying Protocol Supporting Multi-candidate Voting

Based on protocol proposed in [15], we proposed a complete self-tallying protocol supporting multi-candidate voting, which is self-tallying without any credible third party and solves abortive and adaptive issues. The followings are the detailed steps of our protocol.

SETUP: Let G denote a finite cyclic group of prime order q in which the decision Diffie-Hellman problem is intractable [21]. In addition, g is the generator of G and e is a random integer. Administrator's private key is s and its public key is g^s. Let $n = H(g || g^e || g^s)$ and $u = e - sn$. There are N voters in list of eligible votes and T candidates. Candidate C_k is expressed by vote $m_k = 2^{qk}, 2^q > N$, $k = 0, 1, \ldots, T$ where $m_0 = 2^0$ represents abstention vote. Thus, we can assume that there are $T+1$ candidates.

SIGHUP: Each V_i selects their private key x_i and computes their voting key $X_i = g^{x_i}$ by using Schnorr ZKP(x_i) [19] to verify the validity of x_i. The protocol computes reconstruction of the key $h = \Pi_{j=1}^{i-1} g^{x_j} / \Pi_{j=i+1}^{n} g^{x_j}$ for each voter after passing Schnorr ZKP.

VOTE: If voter V_i chooses the candidate C_k , the protocol would publish $Y_i' = h^{x_i} g^{m_k} g^e$ to Ethereum after $Y_i = h^{x_i} g^{m_k}$ passing "Efficient One-out-of-T" ZKP. After the VOTE step is over, the protocol would cast (n, u, g^s, g^e) to Ethereum.

TALLY: Ethereum collects all vote Y_i' and anyone can verify if $g^e = g^u g^{sn}$. If yes, the protocol would compute g^{-e} and tally $\prod_i Y_i' g^{-eN} = g^{\sum M} (\prod_i g^{x_i y_i} = 1$, process of proof in Ref. [8]), where $\sum M = 2^0 \cdot c_0 + 2^q \cdot c_1 + \cdots + 2^{Tq} \cdot c_T$, c_0 to c_T are the counts of votes for $T+1$ candidates correspondingly, including the count of abstention vote. Since $\sum M$ is normally a small number for Corporate Board Voting, it is not difficult to compute the discrete logarithm of $g^{\sum M}$, for example, by using exhaustive search or baby-step giant-step algorithm [11].

V. Security Analysis

Complete self-tallying E-voting protocol supporting multi-candidate mainly relies on Ethereum and cryptography schemes to ensure its security and robustness. Smart contracts on Ethereum ensure the correct execution of the cryptography

schemes. The attacks on Ethereum are beyond the scope of our research. The security analysis of our protocol is as follows.

Anonymity: The voters' identities are encrypted by using cryptographic algorithm, like Schnorr ZKP. In addition, these data are recorded on Ethereum to ensure their security and verification.

Pretended-voting-avoided: All voting keys g^{x_i} and their ZKP are publicly sent to Ethereum. A potential attack is that another eligible voter can attempt to register the same voting keys by replaying g^{x_i} and ZKP(x_i). This would also let them later copy the targeted voters vote. The hash function of Schnorr ZKP in SIGHUP includes Ethereum account and Ethereum will not accept ZKP(x_i) if Ethereum account does not match the account that is calling the contract. As such, it is not possible to replay another voters key g^{x_i} without their co-operation.

Privacy: Each voter's vote $(X_i, Y_i') = (g^{x_i}, h^{x_i} g^{m_k} g^e)$ is encrypted and verified by using ZKP. In SIGNUP, each voter chooses his secret key x_i where x_i is a random integer and casts his public key X_i to Ethereum. In VOTE, each voter casts his vote Y_i' to Ethereum.

The public information for attackers are g^{x_i}, g^s, Schnorr ZKP in SIGNUP, and "Efficient-One-out-of-T" ZKP in VOTE. Attackers can not get x_i , s from g^{x_i}, g^s under Diffie-Hellman assumption. In addition, Schnorr ZKP reveals that whether the voter knows the discrete logarithm x_i of g^{x_i}. Also, "Efficient One-out-of-T" ZKP only reveals that whether the message m_k is one of $m_0, m_1, \ldots, m_T$. Thus, Schnorr ZKP and "Efficient One-out-of-T" ZKP do not reveal any more information than what is intended, and the attackers can not get private m_k from $h^{x_i} g^{m_k} g^e$.

Self-tallying: Our protocol is self-tallying and anyone can check the final voting result. In TALLY, the random number on each voter's vote can be offset by using proof of theorem to achieve self-tallying. What's more, zero knowledge proof protects each voter against attacks in SIGNUP and VOTE.

Dispute-freeness: Our protocol consider Blockchain as a credible bulletin board. Each voter and their vote will be verified by ZKP to ensure their qualification. Thus, our protocol is dispute freeness.

Man-in-the-Middle (MITM) Attack: MITM is an attack where the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other. Our portocol has strong resistance to MITM attack, since each administrator, voter and candidate use public key encryption scheme which is not the target of MITM.

Denial-of-Service (DoS) Attack: DoS attack is a cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet. Attacking nodes is the main attack method of DoS, however the opponent cannot have all the nodes of Blockchain, so the DoS attack is almost impossible to succeed.

TABLE II: Comparison of E-voting protocols based on Blockchain

Blockchain-based E-voting protocol	Ref. [15]	Ref. [24]	Ref. [25]	Ours
Ciphergraph	Zero knowledge proof	Ring signature Homomorphic encryption	Blind signature Ring signature	Zero knowledge proof
Voting scenarios	Small-scale	Large-scale	Small-scale	Small-scale
Support multi-candidate	NO	YES	YES	YES
Self-tallying	YES	NO	YES	YES
Credible third party	YES	NO	NO	YES
Security issues	Abortive and Adaptive issues	Administrator can not disclose the secret key	Signer can not disclose the secret key used in blind signature	Abortive and Adaptive issues have been solved

VI. Performance Analysis

A. Theoretical Analysis

In this section, we compare our E-voting protocol with some existing E-voting protocols based on Blockchain in terms of performance. In addition, we analyze "One-out-of-T" ZKP and "Efficient One-out-of-T" ZKP theoretically.

In Table II, we evaluate the performances of our proposed E-voting protocol, compared with the protocols in [15][24][25]. Compared with Ref. [15], our protocol supports multi-candidate voting by using "Efficient One-out-of-T ZKP", besides, abortive and adaptive issues are solved by using 'Efficient One-out-of-T ZKP" and Schnorr signature. Compared with Ref. [24], our protocol supports self-tallying by using "complete self-tallying protocol", and thus it doesn't need credible third party. Involving the security issues of administrator disclosing their key, our protocol can maintain robustness. Compared with Ref. [25], our protocol doesn't need credible third party. In summary, our protocol has better performance.

In Table III, we theoretically analyze the costs of the prover and the verifier in "One-out-of-T" ZKP and "Efficient One-out-of-T" ZKP. Assume there are $T(T \geq 3)$ candidates, and we conduct theoretical analysis based on parity of T. We set an exponential operation denoted as E, and a multiplication operation denoted as M. Since hash funcitons are stable in quantity and pseudo-random functions are not consuming operation, so we do not include them in theoretical analysis.

Due to the parity, the prover and verifier part are divided into two parts in "Efficient One-out-of-T" ZKP, thus the computational overhead of "One-out-of-T" ZKP is significantly larger than "Efficient One-out-of-T" ZKP. The number of exponential and multiplication operations in "Efficient One-out-of-T" ZKP is nearly half less than "One-out-of-T" ZKP.

B. Experiment Analysis

Experiment analysis was performed on a Lenovo Thinkstation running window 7 equipped with 4 cores, 3. 2 GHz Intel Core i5 and 8 GB DDR3 RAM. All time measurements are rounded up to the next whole millisecond. We choose

TABLE III: Theoretical Analysis of One-out-of-T ZKP and Efficient One-out-of-T ZKP

Number of candidates:T ≥3	Exponential operation:E	Multiplication operation:M
Scheme	**One-out-of-T ZKP**	**Efficient One-out-of-T ZKP**
The Prover	$5TE + (3T-2)M$	$\frac{5T}{2}E + \frac{3T-4}{2}M$ (If T is even) $\frac{5T+5}{2}E + \frac{3T-1}{2}M$ (If T is odd)
The Verifier	$5TE + 3TM$	$\frac{5T}{2}E + \frac{3T}{2}M$ (If T is even) $\frac{5(T-1)}{2}E + \frac{3(T-1)}{2}M$ (If T is odd)

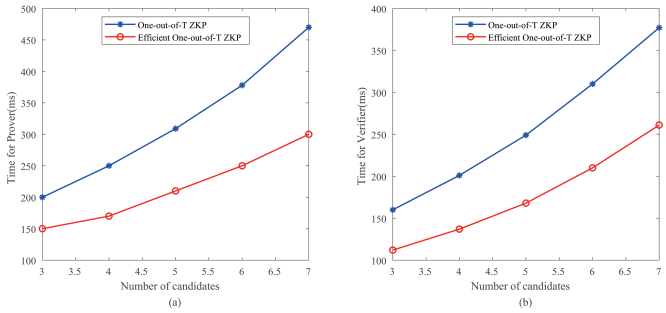

Fig. 2: Time cost for One-out-of-T and Efficient One-out-of-T ZKP

Ethereum as the framework of Blockchain, and built a private chain based on geth-1. 8. 3 and EthereumWallet-0. 8. 10. We test a hundred sets of data and take the average. In addition, our protocol is for small-scale E-voting, so only up to 7 candidates have been tested.

Fig. 2 shows that "Efficient One-out-of-T" ZKP has better efficiency than "One-out-of-T" ZKP in the prover and the verifier part. As pseudo-random functions are computed in the prover part, time consumption of the prover is more than the verifier. Moreover, according to the theoretical analysis in Table III, "Efficient One-out-of-T" ZKP is nearly twice as efficient as "One-out-of-T" when the number of candidates is large enough.

VII. Conclusion

In this paper, we propose a Blockchain-based complete self-tallying E-voting protocol in which Blockchain is used as a bulletin board. The protocol uses an "Efficient One-out-of-T" ZKP for achieving multi-candidate voting, and the issues of abortive and adaptive are solved. The security analysis shows that our proposed protocol meets the security requirements and the performance analysis confirms that our protocol is efficient and practical.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (61702342) and the Science and Technology Innovation Projects of Shenzhen (JCYJ20170302151321095).

References

[1] Buterin, V., et al.: A next-generation smart contract and decentralized application platform. white paper **3**, 37 (2014)
[2] Chow, S.S., Liu, J.K., Wong, D.S.: Robust receipt-free election system with ballot secrecy and verifiability. In: NDSS. vol. 8, pp. 81–94 (2008)
[3] CON, D.: Def con@ hacking conference (2017)
[4] ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory **31**(4), 469–472 (1985)
[5] Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: International Workshop on the Theory and Application of Cryptographic Techniques. pp. 244–251. Springer (1992)
[6] Gibson, J.P., Krimmer, R., Teague, V., Pomares, J.: A review of e-voting: the past, present and future. Annals of Telecommunications **71**(7-8), 279–286 (2016)
[7] Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM Journal on computing **18**(1), 186–208 (1989)
[8] Hao, F., Zieliński, P.: A 2-round anonymous veto protocol. In: International Workshop on Security Protocols. pp. 202–211. Springer (2006)
[9] Joaquim, R., Zúquete, A., Ferreira, P.: Revs–a robust electronic voting system. IADIS International Journal of WWW/Internet **1**(2), 47–63 (2003)
[10] Kiayias, A., Yung, M.: Self-tallying elections and perfect ballot secrecy. In: International Workshop on Public Key Cryptography. pp. 141–158. Springer (2002)
[11] Lenstra, A.K., Lenstra Jr, H.W.: Algorithms in number theory. In: Algorithms and Complexity, pp. 673–715. Elsevier (1990)
[12] Li, C.T., Hwang, M.S., Lai, Y.C.: A verifiable electronic voting scheme over the internet. In: 2009 Sixth International Conference on Information Technology: New Generations. pp. 449–454. IEEE (2009)
[13] Li, Y., Susilo, W., Yang, G., Yu, Y., Liu, D., Guizani, M.: A blockchain-based self-tallying voting scheme in decentralized iot. arXiv preprint arXiv:1902.03710 (2019)
[14] Liu, J.K., Wong, D.S.: Linkable ring signatures: Security models and new schemes. In: International Conference on Computational Science and Its Applications. pp. 614–623. Springer (2005)
[15] McCorry, P., Shahandashti, S.F., Hao, F.: A smart contract for boardroom voting with maximum voter privacy. In: International Conference on Financial Cryptography and Data Security. pp. 357–375. Springer (2017)
[16] Nakamoto, S., et al.: Bitcoin: A peer-to-peer electronic cash system (2008)
[17] Neff, C.A.: A verifiable secret shuffle and its application to e-voting. In: Proceedings of the 8th ACM conference on Computer and Communications Security. pp. 116–125. ACM (2001)
[18] Ryan, P.Y.: Prêt à voter with paillier encryption. Mathematical and Computer Modelling **48**(9-10), 1646–1662 (2008)
[19] Schnorr, C.P.: Efficient signature generation by smart cards. Journal of cryptology **4**(3), 161–174 (1991)
[20] Shinde, S.S., Shukla, S., Chitre, D.: Secure e-voting using homomorphic technology. International Journal of Emerging Technology and Advanced Engineering **3**(8), 203–206 (2013)
[21] Stinson, D.R.: Cryptography: theory and practice. Chapman and Hall/CRC (2005)
[22] Wang, K.H., Mondal, S.K., Chan, K., Xie, X.: A review of contemporary e-voting: Requirements, technology, systems and usability. Data Science and Pattern Recognition **1**(1), 31–47 (2017)
[23] Weber, S.: A coercion-resistant cryptographic voting protocol-evaluation and prototype implementation. Darmstadt University of Technology, http://www. cdc. informatik. tudarmstadt. de/reports/reports/StefanWeber. diplom. pdf (2006)
[24] Yu, B., Liu, J.K., Sakzad, A., Nepal, S., Steinfeld, R., Rimba, P., Au, M.H.: Platform-independent secure blockchain-based voting system. In: International Conference on Information Security. pp. 369–386. Springer (2018)
[25] Zhu, Y., Zeng, Z., Lv, C.: Anonymous voting scheme for boardroom with blockchain. International Journal of Performability Engineering **14**(10), 2414–2422 (2018)