Privacy-Preserving and Verifiable Electronic Voting Scheme Based on Smart Contract of Blockchain

Ting Liu, Zhe Cui, Hongjiang Du, and Zhihan Wu
(Corresponding author: Ting Liu)

Chengdu Institute of Computer Applications, Chinese Academy of Sciences
No. 9, Renmin South Road Section 4, Chengdu 610041, China
School of Computer and Control Engineering, University of Chinese Academy of Sciences
No. 19, Yuquan Road, Shijingshan District, Beijing 100049, China
(Email: liuting315@ mails.ucas.ac.cn)
(Received Sept. 2, 2019; Revised and Accepted Dec. 28, 2019; First Online Jan. 21, 2020)

Abstract

In this study, a privacy-preserving and verifiable electronic voting scheme is proposed based on a smart contract that is cost-effective and practical. The scheme uses electronic ballot as token for voting, and the smart contract verifies accuracy of the ballot. First, an agent generates electronic ballot via ElGamal encryption scheme, which is verified by the smart contract. The agent then generates decryption parameters based on the electronic ballot. Second, the agent assigns the electronic ballot to a voter and shares the decryption parameters to all voters with Shamir secret sharing scheme. Third, a voter generates and submits a vote that is the electronic ballot and a public parameter to the smart contract. Finally, the voter computes decryption data with the sum of decryption parameters restored by smart contract using shares summary submitted from voters. The voter then computes voting result via the homomorphic method with the decryption data. Experiment illustrates correctness and practicality of the proposed scheme.

Keywords: Block-Chain; Electronic Voting; Homomorphic Encryption; Smart Contract

1 Introduction

With the development and application of electronic technology, electronic voting (e-voting) has become an important method in various elections across the world [14,28]. E-voting uses computer and communication network technology to conduct voting activities with electronic ballot and digital vote instead of traditional paper printed ballot. It makes the voting more convenient and increases the efficiency of tallying votes with accuracy.

Various cryptography methods are implemented to preserve the privacy and verifiability of e-voting [17,18]. The first proposed method is termed as Mix-Nets that requires complex algorithms to protect voting privacy and realize public verification [7]. This was followed by blind signature based scheme, which depends on trusted signature institutions in the voting process. This type of scheme is not popular due to defects such as the complex of voting operation [1, 9, 13, 20]. Shamir secret sharing scheme (Shamir SSS) is another common cryptography method in e-voting. Shamir SSS splits secret digital information (an integer, for instance) into multiple shares, only some of which can restore the original secret information (the integer) [31]. In another study [21], the vote is encrypted via the ElGamal scheme, and the private key is shared to multiple authority centers by SSS to decipher the vote without restoring the key. Homomorphic cryptography is a common technology to protect voting privacy. By applying homomorphic cryptography, ciphertext of the votes tally are obtained via computing the ciphertext of votes, and the votes tally results are then decrypted. Hirt *et al.* used homomorphic cryptography to encrypt the vote, and verified the encryption by voters to ensure that the vote could not be tampered with [15]. In the scheme proposed by Ihsan Jabbar *et al.* [16], different servers encrypt the same ballot via homomorphic cryptography, then directly compute the encrypted ballots and decrypt the result to obtain the voting result. Literature [2] implements full homomorphic encryption based e-voting on cloud infrastructure. Liu *et al.* [24] took the votes of different candidates as Shamir SSS Lagrange polynomial coefficients and combined the homomorphic operation to verify tally result. Although the traditional cryptography schemes exhibit defects, such as high algorithm complexity, they are used to construct the e-voting schemes based on emerging technology to advance the progress in the field [3, 30].

Block-chain technology aims at that participants agree

on a series of consecutive blocks of transactions, invoke smart contract functions, and exchange assets [22, 36, 39, 42]. A few e-voting schemes apply block-chain to increase voting security and reduce the complexity combining with traditional cryptography methods. The schemes based on bitcoin protocol are studied as one of the main technical routes. This type of scheme necessitates the bitcoin for voting. Lee *et al.* [19] proposed a scheme conducting e-voting by means of bitcoin transaction with a third-party qualification audit mechanism. In 2017, Cruz *et al.* [10] proposed a block-chain e-voting scheme that adds ballot information to content of the transacted bitcoin. Zhao *et al.* [41] developed a voting system based on bitcoin with a mechanism to incentivize voting and zero-knowledge-proof on the vote commitment. Another type of e-voting scheme is based on Ethereum, which requires economy cost of Gas for the vote transaction [23,32,38]. The smart contract constitutes a main technology of block-chain voting system. In the protocol developed by McCorry *et al.* [26], the privacy of vote is protected via homomorphic encryption and all votes are tallied by a smart contract. Each voter broadcasts an encrypted vote , the legality of which is verified by a non-interactive zero-knowledge-proof. Literature [40] implements smart contract to verify the validity of encrypted votes during most voting stages. Currently, several voting systems based on block-chain have been developed such as BroncoVotes [11] and SecEVS [33].

The proposed scheme combines smart contract, Shamir SSS and homomorphic encryption to make e-voting privacy-preserving and verifiable. The bitcoin transaction has low time efficiency and necessitates cost economy. Thus the scheme using bitcoin as a token for voting is difficult to implement. E-voting based on Ethereum also necessitates cost economy. In this study, an agent generates electronic ballot as token for voting to a candidate that is similar to the scheme based on bitcoin. A voter transfers vote to candidate as bitcoin transaction that achieves the same credibility and lower cost. The voting scheme always uses a complex algorithm, such as zero-knowledge-proof, to provide proof of the validity of the encrypted vote during the voting process. Thus, the operating of voting becomes tedious and practically difficult. We avoided this problem by producing electronic ballot and verifying its correctness prior to voting. Additionally, by applying the designed electronic ballot and tally method, the implementation of the smart contract in the proposed scheme ensures reliability and efficiency of the block-chain operation. The proposed scheme needs further improvement for high efficiency when the voting has numerous voters.

The rest of this article is organized as follows. The next section gives the preliminaries used in the construction of our e-voting scheme. Section 3 describes technical route of the proposed e-voting scheme. Section 4 details the proposed scheme and the security analysis of it. Section 5 provides experiments on the scheme. Finally, conclusions are given in Section 6.

2 Preliminaries

In this section, we briefly introduce Shamir SSS, ElGamal encryption scheme, block-chain and bitcoin, smart contract, and cast-or-audit method.

2.1 Shamir SSS

Shamir SSS implements (k, n) threshold scheme that allows any k in all n secret shares to collaborate to retrieve the secret [31]. Shamir SSS shares secret polynomials

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}$$

to n shares $(x_i, f(x_i))$ such that $1 \le i \le n$, and restores the polynomials of $f(x)$ via Lagrange interpolation polynomial with k shares. In practical implementation, a_0 in $f(x)$ is shared as a secret, and it is restored as follows:

$$a_0 = (-1)^{k-1} \sum_{i=1}^{k} f(x_i) \prod_{j=1, j\neq i}^{k} \frac{x_j}{x_j - x_i}$$

The additive homomorphism of Shamir SSS is that the shares of multiple shared values are added together to restore the sum of all the values. The restoration of the sum of shared values are expressed by the formula as follows:

If $s = F_I(t_{i_1}, t_{i_2}, \ldots, t_{i_k})$ and $s' = F'_I(t'_{i_1}, t'_{i_2}, \ldots, t'_{i_k})$, then $s + s' = F_I(t_{i_1} + t'_{i_1}, t_{i_2} + t'_{i_2}, \ldots, t_{i_k} + t'_{i_k})$. F_I denotes Shamir SSS restoration algorithm, s and s' denote secret sharing values, $\{t_{i_1}, t_{i_2}, \ldots, t_{i_k}\}$ and $\{t'_{i_1}, t'_{i_2}, \ldots, t'_{i_k}\}$ denote the shares of s and s' [6].

2.2 ElGamal Encryption Scheme

ElGamal encryption scheme [12] is public-key cryptography and operated as follows.

1) Select a large prime q. Select numbers r and g that are both less than q. Compute $h = g^r \ mod \ q$;

2) Public key corresponds to h, private key corresponds to r, and public parameters corresponds to g and q;

3) Plain text M should be encrypted. Select a random integer k less than q, encrypt M to $\left(g^k, h^k M\right) \ mod \ q$;

4) Decrypt the ciphertext of M as $M = \left[h^k M (g^r)^{-k}\right] \ mod \ q$.

ElGamal encryption exhibits the homomorphic property as follows [8]. Character E symbolizes the encryption process, M_1 and M_2 are plain texts. The encryption of M_1 and M_2 corresponds to $E(M_1) = \left(g^{k_1}, h^{k_1} M_1\right)$ and $E(M_2) = \left(g^{k_2}, h^{k_2} M_2\right)$. It is defined that $k = k_1 + k_2$. Because $E(M_1) \times E(M_2) = \left(g^{k_1+k_2}, h^{k_1+k_2} M_1 M_2\right)$ and $E(M_1 M_2) = \left(g^k, h^k M_1 M_2\right)$, it is proven that $E(M_1 M_2) = E(M_1) \times E(M_2)$.

2.3 Block-Chain and Bitcoin

Block-chain was proposed by an anonymous scholar named "Satoshi" on the digital currency paper on bitcoin in 2007 [29]. Block-chain is an open ledger stored and maintained by different nodes. Block-chain technology realizes bitcoin trading without the participation of a trust center. The transaction is shared and stored by each node in the entire network. The bitcoin is the first application of block-chain, which safely and anonymously transacts electronic currency called bitcoin. Each bitcoin owner transfers the coin to the next by signing a hash of the previous transaction of the coin and the public key of the next owner. The public in the network can verify the signature to acquire bitcoin ownership. Sufficient computing power is necessary to transact the bitcoin, and the speed of transaction is low.

2.4 Smart Contract

The concept of intelligent contracts was first proposed in 1994 by Nick Szabo, and defined as a set of digitally specified commitments, including agreements on which the contracting parties can enforce the commitments [34]. Block-chain provides a trusted computing environment, and thus a smart contract is widely studied and implemented. The essence of a smart contract corresponds to code with specific transaction logic that runs on a block-chain. The status and content of a smart contract are public, and users of the chain can review the code to confirm the function of the contract. If the contract is confirmed, then it is not possible to tamper with the content of the contract [25, 27]. The smart contract runs on all verification nodes in the block-chain. When compiled and deployed, it can accurately respond to any parameter input. The process of execution is irreversible and cannot be forced to stop or interrupted midway [35, 37].

2.5 Cast-or-Audit Method

Benaloh [4, 5] elaborated this auditing approach to a system where the verifier marks her choice, the prover prepares the ballot. The verifier then chooses to either decrypt the prepared ballot, or to cast the prepared ballot. Since the prover is irrevocably committed to a particular encryption, and as the prover cannot predict whether the verifier will choose to cast or audit, any cheating by the prover has 50% chance of being audited (and, thus, detected). By repeating the audit as often as desired, the verifier can test the prover as often as desired and increase its confidence in the correctness of the prover's operation. The cast-or-audit method processes as follows.

1) The verifier sends the ballot to the prover. The prover encrypts the ballot with parameter and shows the encrypted ballot to the verifier. Then the verifier can choose two options: Cast or audit.

2) If the verifier chooses to audit the encryption, the prover shows the encryption parameter to it, and the verifier checks the encryption correctness. Then the prover encrypts the ballot with another parameter and the verifier chooses the options again.

3) If the verifier chooses to cast the ballot, it finishes verifying the encrypted ballot. Then the encrypted ballot is ready to be cast.

3 Technical Route of the Proposed E-voting Scheme

In the study, the voting agent and the smart contract are abbreviated as AGT and SC. We use a vote v of value 1, -1 or 0 to present yes vote, no vote or abstention vote for a candidate, respectively. The value of v also presents the difference between yes and no vote for the candidate. When a voter votes yes to a candidate for which $v = 1$, the candidate get 1 yes vote and 0 no vote. The difference between yes and no votes of the candidate gotten from this vote is 1, which equals the value of v. When a voter votes no to a candidate for which $v = -1$, the candidate get 0 yes vote and 1 no vote. The difference between yes and no votes gotten from this vote is -1, which equals the value of v. When a voter votes abstention to a candidate for which $v = 0$, the candidate get 0 yes vote and 0 no vote. The difference between yes and no votes of the candidate gotten from this vote is 0, which equals the value of v. The sum of v of all votes for a candidate equals the sum of all difference between yes and no votes for the candidate, from which the final voting result can be computed correctly. The technical route is given as follows:

1) Agent AGT generates the parameters of vote. A vote v can be divided into two parameters, namely the secret parameter e and public parameter u, product of which equals to v. AGT randomly selects a number corresponding to 1 or -1 as secret parameters e of the vote. The public parameter u is generated by AGT based on e. The parameters e and u of vote number v is shown as Table 1.

2) Smart contract SC verifies the validation of the electronic ballot via the cast-or-audit method detailed in Section 2.5. SC acts as a verifier and AGT acts as a prover. AGT encrypts e with a parameter t to $EB(e, t)$ as an electronic ballot. AGT shows t and e to prove the accuracy of each encryption. At the end of verification, t and e in the encrypted ballot that is ready to cast are maintained secret from SC.

3) AGT signs ID of the voter with $EB(e, t)$, then sends them with the signature to the voter.

4) AGT generates different public parameters u of the vote based on e. AGT creates decryption parameters $DP(u, t)$ with t and all different u. AGT sends all u and shares the corresponding $DP(u, t)$ via Shamir SSS to all voters.

Table 1: Parameters and presentations of a vote

Vote	Yes vote	No vote	Difference between yes and no vote	$v = e \times u$	e	u
Yes	1	0	1	1	1	1
					-1	-1
No	0	1	-1	-1	1	-1
					-1	1
Abstention	0	0	0	0	1	0
					-1	

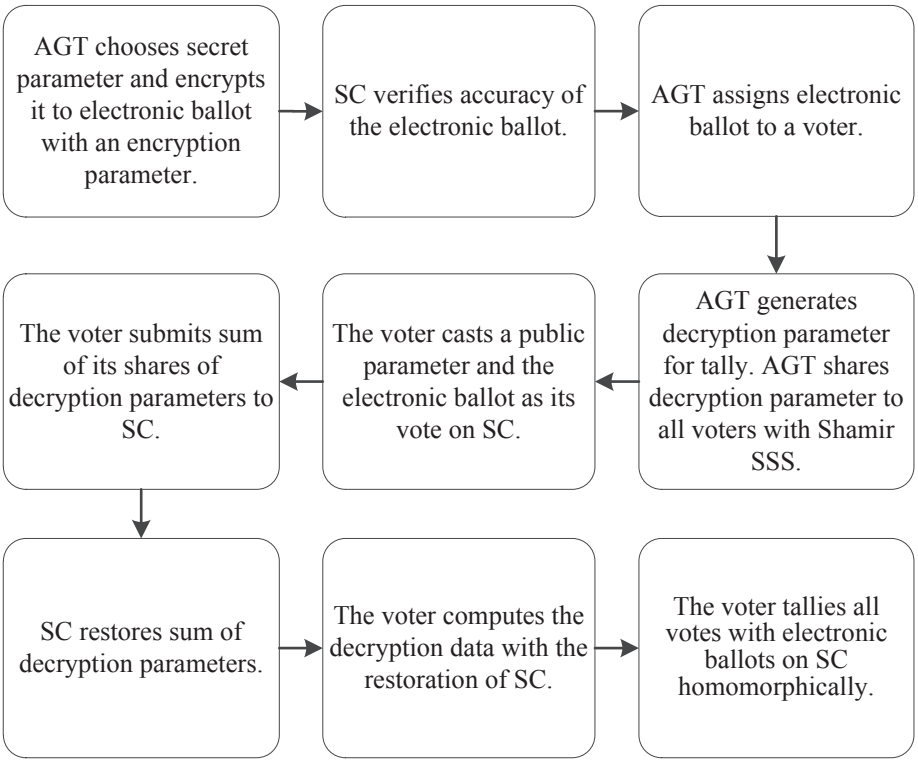

Figure 1: Technical route of the scheme

5) Each voter chooses and submits one $\{u, EB(e,t)\}$ as vote, and the signature of its ID and $EB(e,t)$ by AGT to SC.

6) Each voter submits its sum of shares to SC, and SC restores the sum of all shared $DP(u,t)$. Then the voter computes decryption data from restored sum of all $DP(u,t)$ on SC.

7) Voter tallies all votes saved on SC and decrypts the result with decryption data via homomorphic computation method. The tally result corresponds to the difference between the number of yes and no votes, which leads to the number of yes and no vote with the number of voters. The number of voters excluding the abandoning ones plus the sum of vote of this scheme, and the result summary divides 2 is the number of yes votes.

The technical route of the scheme is shown in Figure 1.

4 Implementation of the Proposed E-voting Scheme

The proposed scheme allows voters to vote from distance with the device of their own. In the proposed scheme, a voter can vote yes, no, and abstention to a candidate, and is allowed to give up voting. It is assumed that m voters are taking part in the voting. Each voter has its own public and private keys for encryption and signature. The signing operations by AGT and voter V_i are denoted as SIG_{AGT} and SIG_i respectively. The hash operation is denoted as $HASH$. This implementation shows the voting approach of V_i for C_j, and voting of V_i for others is the same.

4.1 Initialization

AGT generates its keys and parameters of the ElGamal scheme. AGT generates the secret parameters of votes

and encrypts them to electronic ballots, and the validity of which is verified by SC. Additionally, AGT prepares public parameters of different voting options for voters.

Step 1: As detailed in Section 2.2, AGT generates parameters and keys g, q, K_s, and K_p for ElGamal scheme.

Step 2: With respect to V_i, AGT selects a secret parameter $e_i \in \{-1, 1\}$ and generates random encryption parameter t_i. After encrypting e_i with t_i to electronic ballot that $EB(e_i, t_i) = g^{e_i}(K_p)^{t_i} \ mod\ q$, AGT invokes SC to verify the accuracy of the encryption with the method of cast-or-audit as shown in Section 2.5.

Step 3: The decryption parameter $DP(u_{i,j}, t_i)$ is defined as $DP(u_{i,j}, t_i) = k_s t_i u_{i,j}$. AGT computes 3 decryption parameters $\{DP(1, t_i), DP(-1, t_i), DP(0, t_i)\}$ with public parameters $\{1, -1, 0\}$.

4.2 Registration

The voter registers with AGT to prove its eligibility. AGT provides signed electronic ballot to the voter, and shares decryption parameters. V_i verifies the correctness of $EB(e_i, t_i)$ and AGT proves it via the cast-or-audit method detailed in Section 2.5.

Step 1: V_i sends its identification to AGT. If V_i is eligible, AGT continues the process, or else it rejects voting of V_i.

Step 2: V_i chooses an encryption $EB(e_i, t_i)$ recorded in SC. AGT provides t_i and e_i of the $EB(e_i, t_i)$ to V_i. V_i verifies the accuracy of $EB(e_i, t_i)$ and records e_i. AGT makes signature $SIG_{AGT}(HASH(V_i, EB(e_i, t_i)))$ and provides it to V_i for further operation. Then AGT saves $SIG_{AGT}(HASH(V_i, EB(e_i, t_i)))$ and $\{V_i, EB(e_i, t_i)\}$ on SC for public checking.

Step 3: AGT shares all three decryption parameters $\{DP(1, t_i), DP(-1, t_i), DP(0, t_i)\}$ with Shamir SSS to all voters via network. The shares for voters are denoted with corresponding public parameters 1, -1, and 0.

Step 4: AGT permanently deletes shares of decryption parameters, $DP(u_{i,j}, t_i)$, and $\{t_i, e_i\}$.

4.3 Voting

After registration, V_i submits its vote to the smart contract on its own device.

Step 1: V_i considers $\{V_i, EB(e_i, t_i), u_{i,j}\}$ as its vote for C_j. V_i generates the transaction of vote $Trans_{i,j} = \{V_i, EB(e_i, t_i), u_{i,j}, C_j\}$. V_i signs the hash of $Trans_{i,j}$ to $SIG_i(HASH(Trans_{i,j}))$, then submits $Trans_{i,j}$ and the signature to SC.

Step 2: After checking validity of $\{V_i, EB(e_i, t_i)\}$ saved by AGT in step 2 of the registration phase, SC verifies the signature, hash, and whether $u_{i,j} \in \{-1, 1, 0\}$. If the verification is successful, SC records $Trans_{i,j}$ and $SIG_i(HASH(Trans_{i,j}))$.

4.4 Tally Preparation

According the additive homomorphism of Shamir SSS, SC restores the sum of decryption parameters with the sum of shares submitted by voters. V_i computes decryption data with the restoration of SC.

Step 1: Based on the vote transaction saved on SC, each voter sums all the shares of $DP(u_{i,j}, t_i)$ of voters who cast votes, and submits the summary to SC.

Step 2: SC restores the sum of decryption parameter DP_j for C_j with the shares summary submitted by different voters so that

$$DP_j = \sum_{i=1}^{m} DP(u_{i,j}, t_i) = K_s \sum_{i=1}^{m} (t_i u_{i,j}).$$

Step 3: V_i checks correctness of the restoration with the shares on SC, and computes decryption data DD_j with DP_j.

$$DD_j = g^{DP_j} \ mod\ q = g^{K_s \sum_{i=1}^{m} (t_i u_{i,j})} \ mod\ q.$$

4.5 Tally

Voter V_i tallies the votes on SC with the decryption data.

Step 1: V_i tallies saved votes of C_j on SC by computing the following:

$$\begin{aligned} EB_j &= \prod_{i=1}^{m} (EB(e_i, t_i)^{u_{i,j}}) \\ &= g^{\sum_{i=1}^{m} e_i u_{i,j}} g^{K_s \sum_{i=1}^{m} (t_i u_{i,j})} \bmod q \\ EB_j / DD_j &= g^{\sum_{i=1}^{m} e_i u_{i,j}} \ mod\ q. \end{aligned} \quad (1)$$

Because the absolute value of $\sum_{i=1}^{m} e_i u_{i,j}$ is not exceeding the number of all voters, it is easy to compute

$$X_j = g^{\sum_{i=1}^{m} e_i u_{i,j}}$$

from the result of EB_j / DD_j computed from Equation (1).

Step 2: The vote which is the difference between yes and no votes of V_i voting for C_j is denoted as $v_{i,j}$. The sum of votes which is the sum of difference between yes and no votes to C_j from all voters is denoted as v_j. SC computes v_j as follows:

$$v_j = \sum_{i=1}^{m} v_{i,j} = \sum_{i=1}^{m} e_i u_{i,j} = \log_g X_j.$$

Step 3: V_i counts the number a_j of voters who voted abstention or abstain from voting for C_j on the SC. It is assumed that C_j gets the number of y_j yes votes and n_j no votes from all voters. As detailed in Section 3, because $m - a_j = y_j + n_j$, V_i obtains $y_j = \frac{m-a_j+v_j}{2}$ and $n_j = \frac{m-a_j-v_j}{2}$.

4.6 Scheme Analysis

In this section, based on all aforementioned methods, the primary information security of the proposed scheme is summarized as follows.

1) Eligibility: The voter registers its IDs with AGT, so that only eligible voter can get the electronic ballot signed by AGT to vote.

2) Privacy: At the end of the registration stage, AGT permanently deletes the parameters and shares which can reveal the vote or decrypt the information in the transaction. The voting privacy is provided by Shamir SSS and ElGamal encryption.

3) Verifiability: All encryptions of secret parameters are verified by SC without revealing the encryption parameters. Voter can check the verification programme deployed on SC. The vote with its voter and candidate are saved on the SC with signature of AGT, and it is not possible to tamper with any of the records without being discovered. According to the smart contract records, each voter can verify whether the voting result is correct.

4) Reliability: The scheme realizes a decentralized voting that is safe from attacks via internet. In the tally preparation stage, SC can compute the decryption data with submissions from only part of all voters, number of which is equal or exceeding the sharing threshold. Voters less than the threshold are unable to effect the tally result without being discovered.

5) Efficiency: In the electronic ballot generation, the encryption parameter, the secret parameter, and the public parameter are chosen to be positive or negative. So the data size in the tally stage is limited as the offset of positive or negative number via the additive homomorphic computation. The tally efficiency of the proposed scheme ensures its practical implementation.

5 Experiments

We considered a voting that 7 voters vote for a candidate as example of the proposed scheme. Because that the large size of data is difficult to be published in the article, the experiment was performed with short parameters. We implemented the proposed scheme based on Hyperledger Fabric 1.4 and the smart contract via the chaincode mechanism. All the symbols in this section have the same meaning with those in Section 4. The experiment of all stages in the proposed scheme is detailed as follows.

5.1 Initialization

Specifically, AGT chooses $q = 10007$, $g = 1009$, and $K_s = 1317$ for ElGamal encryption scheme. Seven voters, namely V_1, V_2, ..., V_6, and V_7 vote for candidate C_1. After verification of SC, the secret parameter e_i, encryption parameter t_i and electronic ballot $EB(e_i, t_i)$ of each voter are listed in Table 2.

Table 2: Parameters and encryptions in the initialization stage

i	V_i	e_i	t_i	$EB(e_i,t_i)$
1	V_1	1	76176	6107
2	V_2	-1	-73426	6799
3	V_3	1	-45241	562
4	V_4	-1	-88765	1013
5	V_5	1	84314	7430
6	V_6	-1	47838	2268
7	V_7	-1	81653	6522

5.2 Registration

AGT generates decryption parameter $DP(u_{i,1}, t_i)$ with public parameter $u_{i,1}$ such that $u_{i,1} \in \{-1, 1, 0\}$ for 3 types of options. The decryption parameter $DP(u_{i,1}, t_i)$ that V_i and public parameter $u_{i,1}$ of each voter voting for C_1 are listed in Table 3.

Table 3: Public parameters of votes and decryption parameters of votes

i	V_i	$u_{i,1}$	$DP(u_{i,1},t_i)$
1	V_1	1	100323792
2	V_2	-1	96702042
3	V_3	1	-59582397
4	V_4	0	0
5	V_5	1	111041538
6	V_6	1	63002646
7	V_7	1	107537001

The votes and their parameters are listed in Table 4.

Table 4: The votes and their parameters

i	V_i	$u_{i,1}$	e_i	$v_{i,1}$	Vote
1	V_1	1	1	1	Yes
2	V_2	-1	-1	1	Yes
3	V_3	1	1	1	Yes
4	V_4	0	-1	0	Abstention
5	V_5	1	1	1	Yes
6	V_6	1	-1	-1	No
7	V_7	1	-1	-1	No

5.3 Voting

Each vote $\{V_i, EB(e_i, t_i), u_{i,j}\}$ cast for C_1 is listed in Table 5.

Table 5: Votes cast for C_1

i	V_i	$EB(e_i, t_i)$	$u_{i,1}$
1	V_1	6107	1
2	V_2	6799	-1
3	V_3	562	1
4	V_4	1013	0
5	V_5	7430	1
6	V_6	2268	1
7	V_7	6522	1

5.4 Tally Preparation

The SC restores the sum of decryption parameters DP_1 for C_1 as $DP_1 = 89766720$. V_i computes the decryption data $DD_1 = 3353$.

5.5 Tally

For C_1, V_i computes the sum of difference between yes and no votes v_1, the abstention vote a_1, the yes vote y_1, and the no vote n_1 with decryption data DD_1 and the ballots multiplicative value EB_1. These data are listed in Table 6.

Table 6: Data and result of the tally

DD_1	EB_1	v_1	a_1	y_1	n_1
3353	8508	2	1	4	2

Because $y_1 = 4$, $n_1 = 2$, and $a_1 = 1$, the voting result is obtained as 4 yes votes, 2 no votes and 1 abandon vote, which are in consistent with all votes from voters shown in Table 4. So the correctness of the proposed scheme is experimented.

6 Conclusion

In the study, we proposed a novel e-voting scheme based on smart contract designed to ensure voting privacy and verifiability. In this scheme, the electronic ballot is generated as the token for voting to a candidate. Hence, the scheme is more cost-effective than the bitcoin-based and Ethereum-based scheme. Using the electronic ballot generated from encrypted secret parameter of vote with verification by a smart contract, this scheme does not necessitate complex zero-knowledge-proof during the voting period and is more practical. The proposed scheme realizes decentralization during vote casting and tallying, therefore attacking via network become very difficult. It achieves primary key security requirements of e-voting.

On-site registration should be discussed if there is a need for increasing information security such as receipt-freeness. On a physical site, it is possible to achieve more protection for voters against leaking voting information. The proposed scheme needs further improvement on the secret sharing scheme to be applied to large scale e-voting of numerous voters with high efficiency.

Acknowledgments

This study was financially supported by the National Natural Science Foundation of China under grant No. 61501064, Sichuan Technology Support Program under grant No. 2015GZ0088, Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis Open Fund under grant No. HCIC201502 and No. HCIC201701. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] K. M. R. Alam, A. Maruf, Md. R. R. Rakib, G. G. Md. N. Ali, P. H. J. Chong, and Y. Morimoto, "An untraceable voting scheme based on pairs of signature," *International Journal of Network Security*, vol. 20, no. 4, pp. 774–787, 2018.

[2] A. A. A. Aziz, H. N. Qunoo, and A. A. A. Samra, "Using homomorphic cryptographic solutions on e-voting systems," *International Journal of Computer Network and Information Security*, vol. 10, no. 1, pp. 44–59, 2018.

[3] S. Bartolucci, P. Bernat, and D. Joseph, "SHARVOT: Secret SHARe-based VOTing on the blockchain," in *IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*, pp. 30–34, May 2018.

[4] J. Benaloh, "Simple verifiable elections," in *Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology*, pp. 5–14, June 2006.

[5] J. Benaloh, "Ballot casting assurance via voter-initiated poll station auditing," in *Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology*, pp. 14–20, June 2007.

[6] J. C. Benaloh, "Secret sharing homomorphisms: Keeping shares of a secret (extended abstract)," in *Conference on the Theory and Application of Cryptographic Techniques (CRYPTO'86)*, pp. 251–260, Aug. 1986.

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

[8] J. C. Corena and J. A. Posada, "Multiplexing schemes for homomorphic cryptosystems," *Elementos*, vol. 1, no. 1, pp. 21–32, 2013.

[9] L. F. Cranor and R. K. Cytron, "Sensus: A security-conscious electronic polling system for theinternet,"

in *The 30th Hawaii International Conference on System Sciences (HICSS'97)*, vol. 3, pp. 561–570, Jan. 1997.

[10] J. P. Cruz and Y. Kaji, "E-voting system based on the bitcoin protocol and blind signatures," *Transactions on Mathematical Modeling and Its Applications*, vol. 10, no. 1, pp. 14–22, 2017.

[11] G. G. Dagher, P. B. Marella, M. Milojkovic, and J. Mohler, "BroncoVotes: Secure voting system using ethereum's blockchain," in *The 4th International Conference on Information Systems Security and Privacy (ICISSP'18)*, pp. 96–107, Jan. 2018.

[12] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[13] A. Fujioka, T. Okamoto, and K. Ohta, "A practical secret voting scheme for large scale elections," in *The Workshop on the Theory & Application of Cryptographic Techniques (AUSCRYPT'92)*, pp. 244–251, Dec. 1992.

[14] J. P. Gibson, R. Krimmer, V. Teague, and J. Pomares, "A review of e-voting: The past, present and future," *Annals of Telecommunications*, vol. 71, no. 7-8, pp. 279–286, 2016.

[15] M. Hirt and K. Sako, "Efficient receipt-free voting based on homomorphic encryption," in *International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'00)*, pp. 539–556, May 2000.

[16] I. Jabbar and N. A. Saad, "Design and implementation of secure remote e-voting system using homomorphic encryption," *International Journal of Network Security*, vol. 19, no. 5, pp. 694–703, 2017.

[17] R. Jardí-Cedó, J. Pujol-Ahulló, J. Castellí-Roca, and A. Viejo, "Study on poll-site voting and verification systems," *Computers & Security*, vol. 31, no. 8, pp. 989–1010, 2012.

[18] H. Jonker, S. Mauw, and J. Pang, "Privacy and verifiability in voting systems: Methods, developments and trends," *Computer Science Review*, vol. 10, pp. 1–30, 2013.

[19] K. Lee, J. I. James, T. G. Ejeta, and H. J. Kim, "Electronic voting service using block-chain," *The Journal of Digital Forensics, Security and Law*, vol. 11, no. 2, pp. 123–136, 2016.

[20] C. T. Li and M. S. Hwang, "Security enhancement of chang-lee anonymous e-voting scheme," *International Journal of Smart Home*, vol. 6, no. 2, pp. 45–52, 2012.

[21] Y. J. Li, C. G. Ma, and L. S. Huang, "An electronic voting scheme(in chinese)," *Journal of Software*, vol. 16, no. 10, pp. 1805–1810, 2005.

[22] I. C. Lin and T. C. Liao, "A survey of blockchain security issues and challenge," *International Journal of Network Security*, vol. 19, no. 5, pp. 653–659, 2017.

[23] V. C. T. Linh, C. M. Khoi, D. L. B. Chuong, and A. N. Tuan, "Votereum: An Ethereum-based e-voting system," in *IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF'19)*, pp. 1–6, Mar. 2019.

[24] Y. N. Liu and Q. Y. Zhao, "E-voting scheme using secret sharing and k-anonymity," *World Wide Web*, vol. 2, pp. 1657–1667, 2018.

[25] D. Macrinici, C. Cartofeanu, and S. Gao, "Smart contract applications within blockchain technology: A systematic mapping study," *Telematics and Informatics*, vol. 35, pp. 2337–2354, 2018.

[26] 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 (FC'17)*, pp. 357–375, Apr. 2017.

[27] W. Z. Meng, J. F. Wang, X. M. Wang, J. Liu, Z. X. Yu, J. Li, Y. J. Zhao, and S. S. M. Chow, "Position paper on blockchain technology: Smart contract and applications," in *The 12th International Conference on Network and System Security (NSS'18)*, pp. 474–483, Aug. 2018.

[28] M. F. M. Mursi, G. M. R. Assassa, A. Abdelhafez, and K. M. A. Samra, "On the development of electronic voting: A survey," *International Journal of Computer Applications*, vol. 61, no. 16, pp. 1–11, 2013.

[29] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2019. (`https://bitcoin.org/bitcoin.pdf`)

[30] K. Nir and V. Jeffrey, "Blockchain-enabled e-voting," *IEEE Software*, vol. 35, no. 4, pp. 95–99, 2018.

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

[32] S. Shukla, A. N. Thasmiya, D. O. Shashank, and H. R. Mamatha, "Online voting application using ethereum blockchain," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI'18)*, pp. 873–880, Sep. 2018.

[33] A. Singh and K. Chatterjee, "SecEVS: Secure electronic voting system using blockchain technology," in *International Conference on Computing, Power and Communication Technologies (GUCON'18)*, pp. 863–867, Sep. 2018.

[34] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, pp. 1–21, 1997.

[35] F. Tariq and R. Colomo-Palacios, "Use of blockchain smart contracts in software engineering: A systematic mapping," in *The 19th International Conference on Computational Science and Its Applications (ICCSA'19)*, pp. 327–337, Oct. 2019.

[36] S. Underwood, "Blockchain beyond bitcoin," *Communications of the ACM*, vol. 59, no. 11, pp. 15–17, 2016.

[37] S. Wang, L. Ouyang, Y. Yuan, X. C. Ni, X. Han, and F. Y. Wang, "Blockchain-enabled smart contracts: Architecture, applications, and future trends," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, pp. 1–12, 2019.

[38] X. Yang, X. Yi, S. Nepal, and F. L. Han, "Decentralized voting: A self-tallying voting system using a smart contract on the ethereum blockchain," in *The 19th Web Information Systems Engineering (WISE'18)*, pp. 18–35, Nov. 2018.

[39] Y. Yong and F. Y. Wang, "Blockchain: The state of the art and future trends," *Acta Automatica Sinica (in chinese)*, vol. 42, no. 4, pp. 81–94, 2016.

[40] W. Zhang, S. Huang, Y. Yuan, Y. Y. Hu, S. H. Huang, S. J. Cao, and A. Chopra, "A privacy-preserving voting protocol on blockchain," in *IEEE 11th International Conference on Cloud Computing (CLOUD'18)*, pp. 401–408, June 2018.

[41] Z. Zhao and T. H. Chan, "How to vote privately using bitcoin," in *The 17th International Conference on Information and Communications Security*, pp. 82–96, Dec. 2015.

[42] Z. Zheng, S. Xie, H. Dai, X. P. Chen, and H. M. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *IEEE International Congress on Big Data (BigData Congress)*, pp. 557–564, June 2017.

Biography

Ting Liu received the M.S degree in Computer Software and Theory from Xi'an Technological University in 2011. He is currently a Ph.D candiadate in University of Chinese Academy of Sciences. His research interests include Electronic Voting, Block-chain and Secret Sharing.

Zhe Cui received the Ph.D degree in Computer Software and Theory from University of Chinese Academy of Sciences in 2011. He is Ph.D supervisor and researcher of Chinese Academy of Sciences. His research interests include Pattern Recognition and Information Security.

Hongjiang Du received the M.S degree of Engineering in Computer Science and Technology from Sichuan University in 2006. He is currently a Ph.D candidate in University of Chinese Academy of Sciences. His research interests include Electronic Voting, Coding Theory, and Information Security.

Zhihan Wu received the Bachelor degree of Engineering in Information Scecurity from Sichuan University in 2017. She is currently a M.S candiadate in University of Chinese Academy of Sciences. Her research interests include Electronic voting, Block-chain, Cryptography.