A Format-Verifiable E-Voting Scheme Using Homomorphic Encryption

Yuhong Sun[1,3], Jiatao Wang[2], Fengyin Li[1,3]*

[1] School of Computer Science, Qufu Normal University, China
[2] Department of Civil Engineering, Shandong Water Conservancy Vocational College, China
[3] Rizhao-Qufu Normal University Joint Technology Transfer Center, China
sun_yuh@163.com, wy_wjt@126.com, Lfyin318@126.com

Abstract

Electronic-voting (e-voting) is taking place of the traditional paper voting, due to its efficiency and environmental protection. To protect the privacy of the votes, voters usually blind or encrypt the votes before submitting them. But the blinded or encrypted votes also hide the format of them and the receiver cannot distinguish if they are legal or not in format. To solve such a problem, this paper proposes a privacy-preserving e-voting scheme, that can validate the correctness of the cast ballots in format. Two protocols are designed based on homomorphic encryption to verify the format of the votes, without disclosing the content, and the designated verifier signature is adopted to obtain the receipt-freeness. The analysis shows the provable security of the protocols, and the proposed e-voting scheme achieves the format verifiability, in addition to meeting the requirements of other aspects in security.

Keywords: E-voting, Homomorphic encryption, Privacy-preserving, Verifiable format

1 Introduction

In modern society, voting is a common way to express people's willingness. For the cumbersome process and low efficiency, the traditional paper voting is being replaced by the electronic voting (e-voting) that may be a novel solution in the future. However, such as the eligibility of voters, the privacy of votes, the correctness and the verifiability of the results, are challenging in practical e-voting.

Among the challenges, privacy is the most basic requirement that requires the ballot content to be hidden, before it being counted or tallied. In condition of voters being honest or semi-honest, the ballot is legal in the form, such as the number of favorite candidates not greater than the prescribed number. However, a dishonest voter in practice may cast a ballot with a wrong format. An example is that a ballot consisting of 2 or more 'approval's for the same candidate, or the overall 'approval's greater than the prescribed number. It is bound to lead to unfairness if the illegal ballots are counted, along with the bribery, coercion and other problems.

In this paper, we focus on the above problem, that verification of the ballot format, in order to reject the invalid ballots before including them. At the same time, the scheme should meet the common requirements in security, thus improve the fairness and credibility of the e-voting.

1.1 Major Technologies

Depending on the technologies used, the existing e-voting schemes may be roughly categorized as: mix-net based schemes [1-5], blind signature based schemes [6-8], ring signature based schemes [9-11], and homomorphic encryption based schemes [12-15].

The first mix-net based scheme was proposed by Chaum [1], that allows a number of servers to shuffle the encrypted votes and hides the relationship between the voters and votes. In such kind of schemes, it is hard to convince the voter that his vote is re-encrypted and not replaced or discarded. The most recent mix-net cryptographic voting [5] proposed a verifiable secret shuffle for BGV ciphertexts and a compatible verifiable distributed decryption protocol. This scheme requires a lot of participants, including a trusted set of players to run the setup, a set of voters and their computers, a ballot box, a collection of shuffle servers, a collection of decryption servers and auditors.

The key idea behind the blind signature/ring signature based e-voting scheme is also to break the link between voters and votes. In blind signature based voting, the voter blinds his ballot and sends it to the authority for a signature. Next, the authority authenticates the voter and signs on the blind ballot. Then, the voter unblinds the ballot with the signature and casts it anonymously. Finally, the election authorities verify the signature on the ballot and include it in the tally. One of the earliest voting scheme based on blind signature is the FOO scheme [6]. The recent scheme based on blind signature, such as [16], provided an end-2-end verifiable e-voting using identity-based blind signature. The identity and the biometric feature can ensure the unreusability and be used to verify the final result. In such a kind of scheme, the ballot has to be disclosed for the tally before the end, while the leak of intermediate voting result may potentially lead to unfairness.

The ring signature based e-voting schemes [9-11, 17] utilized the anonymity of the signer in the signature to prevent the voter-vote relationship from been known. To prevent the repetition of voting, the linkable ring signature scheme [11] is adopted to link two ballots from the same

*Corresponding Author: Yuhong Sun; Email: sun_yuh@163.com
DOI: https://doi.org/10.70003/160792642026012701004

voter, where escrowed linkable ring signature is used to get the robustness and the receipt-freeness simultaneously [10]. In the ring signature based schemes, the length of public key depends on the size of the ring, and the voters who would vote must be determined to form a ring ahead, which is usually limited in practice.

The reason for homomorphic encryption used in the e-voting is that the operations on ciphertext can obtain the homomorphic encryption of plaintext computation, which preserves the privacy of votes perfectly. Voters usually need encrypt their votes before submitting them, and the counting center can count the votes on the ciphertexts directly. An inevitable point is that it is not easy to offer verifiability in this type of schemes, i.e. a voter cannot verify if his vote is counted for the candidate of his choice. Some secret sharing based e-voting schemes, such as Liu [18], are also roughly regarded as such a class.

In terms of system architecture, some schemes are built on the trusted authorities, including the e-voting using the DRE [19-20], of which the security depends on the DRE machine. With the popularity of the blockchain technology, many decentralized e-voting schemes arise [20-26] and achieve the decentralized system, most of which combine cryptographic technology and the blockchain to improve the independence on the trusted authority.

1.2 Security Requirements

The early research pointed out that an e-voting system should satisfy the general requirements in security [6], including completeness, soundness, privacy, unreusability, eligibility, fairness and verifiability. Recent research has proposed other desirable properties, such as the coercion resistance [21, 26-27], or the receipt-freeness [10-11, 28].

Among the requirements, some goals seem to be contradictory, such as the privacy and the correctness, the verifiability and the receipt-freeness. The authors [29] pointed out that one cannot get the universal verifiability and the receipt-freeness simultaneously unless the voting process involves interactions between voters and the voting authority. Similarly, the privacy of the ballot content and the format correctness are also hard to balance, especially in the e-voting schemes of multi-choice, based on blind signature or homomorphic encryption. Since the ballot is hidden for the content privacy, no one can know if the ballot is well-formed. The scheme of FOO [6] utilized the commitment to verify the ballot format, while the ballot content is also opened and leaked at the same time. The format verifiability is necessary in practice, for the voters may cheat in the blinded or encrypted ballot. In fact, the well-formed ballot is considered [19-20], where the proof of well-formedness is implemented by a non-interactive proof of knowledge. But the format in schemes [19-20] only considered each approval for one candidate.

1.3 Motivation and Contribution

In this paper, we mainly consider the problem of format-verifiability in multi-choice voting where voters may cheat when casting their ballots. For example, the voting organizer requires each voter choose m candidates at most, while the voter may choose more than m candidates in his ballot, that spoils the soundness. Another case is that each voter is only allowed to cast 1 'approval' for one candidate, while the voter may cast 2, or more 'approval's for his favorite candidate. Since the ballot is blinded or encrypted, the receiver cannot find the illegality of the ballots.

To solve such a problem, we propose an e-voting scheme to achieve the format verifiability without sacrificing the privacy of the votes based on the homomorphic encryption. To address the verifiability, we re-encrypt the ballot before publishing it, so that the receipt-freeness can also be obtained. The contributions of this paper can be summarized as follows:

(1) A novel e-voting scheme is proposed that can verify the format of ballot without disclosing its content, and meets the common requirements of security including the eligibility, uniqueness, correctness, verifiability, and receipt-freeness.

(2) An interactive comparison protocol is proposed to test if an encrypted value is less than a prescribed value, without disclosing the value encrypted in ciphertext.

(3) An interactive protocol is proposed to verify each element in an encrypted vector no more than '2' without disclosing the plaintext of the elements.

(4) The security of the comparison protocols is proved. In addition, we analyzed the common requirements in security, and tested the performance of the proposed e-voting scheme.

2 Preliminaries

2.1 Homomorphic Encryption

Roughly, a homomorphic encryption scheme enables (certain) computations to be performed on encrypted data, yielding a ciphertext containing the encrypted result. Stated differently, the direct computation on the ciphertext can get the same result as encrypting the computation of plaintext. In this paper, we focus on the additive homomorphic cryptosystem.

The additive homomorphic cryptosystem (e.g. Paillier cryptosystem [30]) has the additive homomorphism property. Suppose $[a_1]_{pk}$ and $[a_2]_{pk}$ are two ciphertexts under the same public key pk in a homomorphic cryptosystem, $D_{sk}([a_1]_{pk} \oplus [a_2]_{pk}) = a_1 + a_2$ holds, where "$\oplus$" indicates operation on the ciphertext, and $D_{sk}(.)$ means the decryption under the secret key sk.

2.2 Paillier Cryptosystem

Paillier cryptosystem is semantically secure under the assumption of the hardness of residue class of composite power [30]. The Paillier cryptosystem utilizes the group $Z^*_{N^2}$, the multiplicative group of elements in the range $\{1, \ldots N^2\}$ that are relatively prime to N^2, where N is a product of two distinct primes. The cryptosystem consists of phases of **KeyGen, Encrypt**, and **Decrypt** after the system setup. The algorithms can be depicted as follows.

KeyGen. Randomly select two large prime numbers p, q, let $N = p \times q$, $\lambda = lcm((p-1), (q-1))$, select a generator g of

$Z^*_{N^2}$, define $L(\mu)=\frac{\mu-1}{N}$. The public key is (N, g), and the secret key is λ.

Encrypt. Randomly select an integer $r \in Z^*_N$, for $a \in Z_N$, the ciphertext is $c = Enc(a) = g^a \times r^N \bmod N^2$.

Decrypt. Given the ciphertext c, and the secret key λ, the plaintext is $a = Dec(c) = \frac{L(c^\lambda \bmod N^2)}{L(g^\lambda \bmod N^2)} \bmod N$.

As an additive homomorphic cryptosystem, it is easy to compute the encryption of ka as $Enc(a)^k \bmod N^2$, given the $Enc(a)$ and integer k. In addition, the scheme suffices to re-encryption for the same message without any decryption, i.e. given $a \in Z_N$, and $r \in Z^*_N$, $Enc(a) = Enc(a) \times Enc(0) \bmod N^2 = Enc(a) \times r^N \bmod N^2$, and $Dec(Enc(a)\, r^N \bmod N^2) = a$.

2.3 Designated-verifier Signature

The designated-verifier signature (DVS) is a special kind of signature such that the verifier can be convinced by the signer of the message, while the verifier cannot prove the fact to others. There exist many DVS schemes, from it being proposed by Jakobsson et al. [31]. Specifically, it can be implemented by setting the size of the group to two in a ring signature [32-34]. Generally, three algorithms including **DVKGen**, **DVSign**, and **DVVerify** are mainly involved in a DVS scheme. Here, a DVS scheme derived from ring signature [34] is depicted as follows. Assume group G (generator g) and G_T with order q, a hash function $H: \{0,1\}^* \rightarrow G$, and a secure bilinear map $e: G \times G \rightarrow G_T$.

DVKGen. Randomly select x_S, $x_D \in Z^*_q$, compute $y_S = g^{x_S}$ and $y_D = g^{x_D}$, where y_S, x_S are public/secret keys for the signer, and y_D, x_D are public/secret keys for the verifier.

DVSign. Given message α, the signer signs on it using his secret key $x_S \in Z^*_q$ and the verifier's public key $y_D \in G$: randomly choose $r \in Z^*_q$, compute $h \leftarrow H(\alpha) \in G$, $\beta \leftarrow (h/y_D^r)^{1/x_S}$ and $\gamma = g^r$; output the signature $\sigma = (\beta, \gamma)$.

DVVerify. Given the message α and $\sigma = (\beta, \gamma)$, compute $h \leftarrow H(\alpha) \in G$, if $e(\beta, y_S)e(\gamma, y_D) = e(g, h)$ holds, output 1; otherwise output 0.

The above signature is actually a DVS, for the signature can also be produced by the verifier. The scheme is proved to be existential unforgeable against chosen-message attacks (EU-CMA) [34] under the assumption of hardness of co-CDH problem and random oracle model.

3 Model, Assumptions, and Goals

3.1 System Model

The participants of the system include the voting managing center (MC), the voting center (VC), the counting center (CC), the voters, and the bulletin board (BB). The MC is the organizer or launcher of the voting, who is responsible for authorizing the legality of the voters, and initializing the voting. The VC receives and aggregates the votes, confirms the legality of the voters, and verifies the votes in format. The CC counts the legal votes and publishes the intermediate and the final voting result on the BB. Each legal voter has his unique identity information (ID) that can prove the legality to MC, and he casts his ballot to VC by a temporary voting ID issued by MC. The BB is used to publish information accessed by all participants, including the voting result. Same as the usual settings, the BB is read-only and the data on it cannot be modified or deleted.

The model can be illustrated as Figure 1.

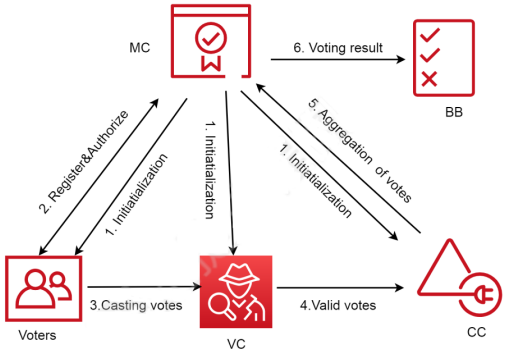


Figure 1. System model

3.2 Security Assumptions

The security assumptions include the trust assumption and the channel assumption. In terms of trust assumption, we assume the MC, VC and CC are all semi-honest. In addition, the MC and VC are not assumed to collude, similar as most existing schemes [6, 10, 16, 18], while the CC is assumed not to collude with MC and VC simutaneously. The voter may create and cast a ballot with incorrect format, or he may be corrupted and sell his votes to others. The voter is not allowed to collude with MC or VC. Aside frome these, we do not consider other malicious behaviors of the voters.

For the channels, we assume the adversary cannot prevent the communication between parties, or the adversary may endeavor or tamper with the data but he cannot stop the communication. To obtain the receipt-freeness, the anonymous channel is the minimum requirement [27].

3.3 Design Goals

In this paper, we focus on the vote format-verifiability or cheat-resistance, along with the soundness or vote eligibility. Thus, the design goals of the proposed scheme include the following requirements.

(1) Cheat-resistance. The tallied ballots must be in correct format. Any cheating in the ballot would be detected and rejected.

(2) Correctness. All valid ballots are counted correctly to reflect the willingness of the voters.

(3) Privacy. Each ballot is private to the voter himself and cannot be known by any other members until the end of voting.

(4) Voter's eligibility. Only the voter who has registered with the MC can cast a ballot.

(5) Uniqueness (unreusability). No voter can vote twice.

(6) Verifiability. Any voter can verify whether his ballot is counted correctly. The public can also verify the final voting result.

(7) Receipt-freeness. The voter cannot make a proof to others on which candidate he has voted after the voting is finished.

3.4 Notations

For the sake of simplicity, we list the notations used in this paper as Table 1.

Table 1. Notations and descriptions

Notations	Descriptions
pk_{ent} , sk_{ent} , $ent \in \{MC, VC, CC\}$	The public key and secret key of the entity in encryption
ps_{ent} , ss_{ent} , $ent \in \{MC, VC, CC\}$	The public key and secret key of the entity in signature
pk_i , sk_i	The public key and secret key of voter v_i in encryption
ps_i , ss_i	The public key and secret key of voter v_i in signature
$[a]_{pk}$	Encryption of message a with the public key pk, abbreviated as $[a]_{ent}$, if pk is pk_{ent}
$\sigma_{ss}(\alpha)$	Signature of message α by the secret key ss, abbreviated as $\sigma_{ent}(\alpha)$, if ss is ss_{ent}
$[x]\times_c[y]$	$[x] \times [y] \bmod N^2$
n	The number of candidates
V_n	The number of legal voters
m	The maximum of 'approval's in a ballot
m'	The actual number of 'approval's in a ballot
blt_i	The ballot of v_i including n encrypted choices

4 Proposed Scheme

The proposed e-voting scheme consists of the phases of initialization, registration and authorization, casting the votes, checking the votes, counting the votes, publishing the results, and verification.

4.1 Initialization

The MC declares the information of all (n) candidates, the identities of legal voters, and other rules of the voting, such as each ballot should include no more than $m (m < n)$ 'approval's, and only one 'approval' for a candidate is allowed.

Given the security parameter κ, MC generates the public/secret key pair according to the algorithm KeyGen of Pailliar encryption system: pk_{MC} and sk_{MC}. At the same time, MC generates his public/secret key pair for signature as ps_{MC}, ss_{MC}. The VC, CC and all voters generate their public/secret key pairs for encryption and signature respectively. In the end of initialization, the public keys are published.

In this scheme, MC is assumed to know the identities of all legal voters before registration. To verify the legality of the voters, MC initializes a list $list_V = [(ID_i, 0)]_{i\in[1, V_n]}$ to avoid the repetition of registration, where ID_i is the identity of each legal voter. At the same time, an empty list named $list_C$ is prepared by MC to record the registered voters.

4.2 Registration and Authorization

To guarantee only legal voters cast votes, each voter must be registered with MC before the voting, to get a unique credential signed by the MC.

Firstly, the voter v_i signs on his ID, combined with his public keys pk_i, ps_i, and encrypts them with pk_{MC}. Then he sends $[(ID_i, pk_i, ps_i, \sigma_{v_i}(ID_i, pk_i, ps_i))]_{\mathrm{MC}}$ to MC. The MC decrypts and verifies the signature after receiving the ciphertext, and then checks $list_V$ to find whether $(ID_i, 0)$ exists in the list. If it does, MC accepts the request and encrypts a signed credential $[Cert_i = (TID_i, \sigma_{MC}(TID_i))]_{pk_i}$ and sends it to the voter v_i, where TID_i is a temporary ID, or a pseudonym of the voter in this voting. Meanwhile, the relationship between ID_i and TID_i is private to MC, and no one except MC can infer one from the other. Finally, MC updates $list_V$ by replacing $(ID_i, 0)$ as $(ID_i, 1)$, and appends $(TID_i, 0, pk_i, ps_i)$ to $list_C$. In the end of registration, MC sends the signed $list_C$ that involves all the TIDs to VC.

4.3 Creating Votes

According to his own willingness, each voter makes his ballot. For n candidates, the voter v_i expresses his vote as a bit vector $(b_{i1}, b_{i2}, \ldots, b_{in})$, where the element $b_{ij} = 1$ means 'approval' and $b_{ij} = 0$ means 'disapproval' or abstain. The vote is then encrypted through pk_{MC} bit by bit: randomly choose $r_{ij} \in Z_N^*$, and compute $[b_{ij}]_{MC} = g^{b_{ij}} r_{ij}^{N} \bmod N^2$ to form the encrypted vote as $blt_i = ([b_{ij}]_{MC})_{j=[1,n]}$. Then, the voter concatenates his vote blt_i and $Cert_i$ to form the ballot $B_i = (blt_i \,||\, Cert_i)$, and signs on it to get $\sigma_i = \sigma_{v_i}(B_i)$. Last, the voter encrypts the (B_i, σ_i) using pk_{VC} to get $EB_i = [B_i || \sigma_i]_{VC}$, and sends EB_i to VC.

4.4 Checking Votes

Receiving EB_i, VC decrypts it to get B_i and σ_i. Next, VC verifies the validity of the ballot from three steps: verifying the legality of the voter, checking the number of the 'approval's, and checking legality of each bit in the ballot.

(1) Verify the legality of the voter: Firstly, VC separates out $Cert_i$ from B_i, and TID_i from $Cert_i$. Then, he checks $list_C$ to find if $(TID_i, 1, pk_i, ps_i)$ exists. If it does, this is a repetition vote and should be rejected. Or else, he takes out the corresponding ps_i from $list_C$, verifies σ_i and the legality of $Cert_i$. When both the verifications pass, go to step (2) for further check.

(2) Check the format to find whether there are more than m 'approval's in this ballot. The main idea is to summarize the items of the ballot and to judge if the sum m' is greater than m by a comparison protocol between two parties. Note that if we compare m' and m directly, one party will know a valid ballot consists m 'approval's if the '=' holds. Instead, we compute $m' < m+1$ to avoid the leakage of the valid ballot.

Firstly, VC firstly separates out $blt_i = ([b_{ij}]_{MC})_{j=[1,n]}$ from B_i, and computes $x = (\prod_{j=1}^{n} [b_{ij}]_{MC} \bmod N^2)$ (x is just encryption of m'). Then, VC chooses a linear function $f(z) = k_1 z + k_2$, where k_1, k_2 are private non-zero integers satisfying $k_1 n + k_2 \in Z_N$, and $k_2 > 0$. Note that the requirement $k_1 n + k_2 \in$

Z_N still can guarantee a high security for a sufficiently large N. The encryption of $f(m')$ and $f(m+1)$ can be obtained easily as $[f(m+1)]_{MC} = [k_1(m+1)+k_2]_{MC}$, $[f(m')]_{MC} = x^k \times_c [k_2]_{MC}$. Then, VC set (e_1, e_2) as $([f(m+1)]_{MC}, [f(m')]_{MC})$, and signs on the tuple (e_1, e_2) before sending it to MC.

Receiving (e_1, e_2), MC verifies the signature and decrypts them. Then he compares the decryption result to get $com = D_{sk_{MC}}(e_1) \geq D_{sk_{MC}}(e_2)$, and sends *com* back to VC.

When the VC receives *com* from MC, the validity of the ballot can be determined: if $(com = 1 \ \& \ k_1 > 0) \| (com = 1 \ \& \ k_1 < 0)$, the ballot is invalid, where '&' means 'AND', and '||' means 'OR' in logic.

The process can be depicted by Protocol 1. And the analysis of the security is in section 5.1.

Protocol 1. Comparison of an encrypted value and a plaintext

Inputs: P_1 has the ciphertext $[x]_{pk}$ and P_2 has the decrypting key $sk = \lambda$.

Auxiliary input: The value y, n, and the public key $pk = (N, g)$.

Output: The result of whether $x \geq y$ for P_1.

The protocol:

@P_1:

• Choose $f(z) = k_1 z + k_2$, where k_1, k_2 are private non-zero integer satisfying $k_1 n + k_2 \in Z_N$ and $k_2 > 0$;

• Compute $[f(x)] = [x]^{k_1} \times_c [k_2]_{pk}$ and $[f(y)]) = [k_1 y + k_2]_{pk}$ according to the homomorphism property of the Paillier encryption.

• Send $(e_1, e_2) = ([f(x)], [f(y)])$ to P_2.

@ P_2:

• With the secret key, decrypt e_1 and e_2 to get $d_1 = D_{sk}(e_1)$, and $d_2 = D_{sk}(e_2)$.

• Compute $com = (d_1 \geq d_2)$ and sends *com* to P_1.

@ P_1:

• Determine the comparison result $c = (x \geq y)$ based on the sign of k_1 and *com*: c=*com* if $k_1 > 0$, and c=1-*com* if $k_1 < 0$.

If the ballot passes this check, go to step (3) for further check.

(3) Check the validity of the ballot by finding if there exists an element greater than 1. The main idea is to compare each element with '2' in ciphertext according to Protocol 2 between MC and VC. To prevent MC knowing the vote when CC publishes them on the BB, VC shuffles the elements before inputting the vector to the protocol. Same as Protocol 1, a linear function $f_{ij}(z)$ is used to hide the value of b_{ij} and '2'. However, if b_{ij} is compared with '2' directly, we find that the $f_{ij}(b_{ij}) \leq f_{ij}(2)$ would lead to $b_{ij} \leq 2$ when $k_1 > 0$ and cannot determine whether $b_{ij} \leq 1$ or not. Similarly, $f_{ij}(b_{ij}) \geq f_{ij}(2)$ when $k_1 < 0$ also produces such a puzzle. To avoid dealing with the '=', $2b_{ij}$ is used to compared with '3' to get: $2b_{ij} > 3$ meaning $b_{ij} > 1$, and $2b_{ij} < 3$ meaning $b_{ij} \leq 1$. To simplify expression, most subscripts 'i' indicating the ballot from voter v_i are omitted in the rest paper.

For $blt = ([b_j]_{MC})_{j=[1,n]}$, VC privately chooses a reversible permutation π to shuffle the elements to get $([b_{\pi(j)}]_{MC})_{j=[1,n]}$ firstly. Then, he computes each encrypted $2b_{\pi(j)}$ to get $([2b_{\pi(j)}]_{MC}) = ([b_{\pi(j)}]^2_{MC}) \bmod N^2$, written as $([b^{(2)}_{\pi(j)}]_{MC})$. Next, VC privately chooses a random bit vector $C = (c_1, \ldots, c_n)$, $c_j \in \{0,1\}$, and creates the following two vectors: $A = (a_1, \ldots, a_n)$, $E = (e_1, \ldots, e_n)$, where $a_j = c_j[b^{(2)}_{\pi(j)}]_{MC} + (1-c_j)[3]_{MC}$, $e_j = c_j[3]_{MC} + (1-c_j)[b^{(2)}_{\pi(j)}]_{MC}$, and $[3]_{MC}$ is the encryption of '3' by pk_{MC}. Last, the VC chooses functions $f_j(z) = k_{1j} z + k_{2j}$, where k_{1j}, k_{2j} are private non-zero integers satisfying $k_{1j} n + k_{2j} \in Z_N$, $k_{2j} > 0$, and computes the encrypted vectors $[F(A)]_{MC} = (a_j^{k_{1j}} \times_c [k_{2j}])_{j=[1,n]}$, and $[F(E)]_{MC} = (e_j^{k_{1j}} \times_c [k_{2j}])_{j=[1,n]}$. Finally, he signs on the two vectors and sends them to MC.

Upon receiving $[F(A)]_{MC}$ and $[F(E)]_{MC}$, MC verifies the signature, decrypts the vector, to obtain $k_{1j} a_j + k_{2j}$ and $k_{1j} e_j + k_{2j}$, written as 'd_{1j}' and 'd_{2j}' respectively. Then he computes $g_j = (d_{1j} > d_{2j})$ for all bits to get the comparison result as $G = (g_j)_{j=[1,n]}$. Finally, MC signs on G and sends it to VC.

With the comparison result G, VC determines g_j for $j \in \{1, \ldots, n\}$: if there exists a j that satisfies $((c_j = g_j) \ \& (k_{1j} > 0) \| (c_j \neq g_j) \ \& (k_{1j} < 0))$, the ballot is invalid. Otherwise, the ballot is regarded valid and the $(TID_i, 0, pk_i, ps_i)$ is updated as $(TID_i, 1, pk_i, ps_i)$ in the $list_C$. The process of the check can be depicted as protocol 2.

Protocol 2. Determine whether an element is greater than 1 in an encrypted vector

Inputs: P_1 has an encrypted vector $([b_j]_{pk})_{j=[1,n]}$, and P_2 has the secret key $sk = \lambda$.

Auxiliary input: The number n and the $pk = (N, g)$.

Output: The result of whether there exists any $b_j > 1$ for P_1.

The protocol:

@ P_1:

• For $j \in \{1, \ldots, n\}$, compute the encrypted $2b_j$ as $[b_j^{(2)}]_{pk} = [b_j]^2_{pk} \bmod N^2$;

• Randomly choose $C = (c_1, \ldots, c_n)$, $c_j \in \{0,1\}$;

• Define two vectors of ciphertext $A = (a_1, \ldots, a_n)$ and $E = (e_1, \ldots, e_n)$:

$a_j = c_j[b^{(2)}_{\pi(j)}]_{MC} + (1-c_j)[3]_{MC}$,

$e_j = c_j[3]_{MC} + (1-c_j)[b^{(2)}_{\pi(j)}]_{MC}$ for $j \in \{1, \ldots, n\}$;

• Choose n private functions $f_j(z) = k_{1j} z + k_{2j}$, $j \in \{1, \ldots, n\}$, and encrypt $F(A)$ and $F(E)$ as:

$[F(A)]_{pk} = (k_{1j} a_j \times_c [k_{2j}]_{pk})_{j=[1,n]}$,

$[F(E)]_{pk} = (k_{1j} e_j \times_c [k_{2j}]_{pk})_{j=[1,n]}$;

• Send $([F(A)]_{pk}, [F(E)]_{pk})$ to P_2.

@P_2:

• Decrypt $[F(A)]_{pk}$ and $[F(E)]_{pk}$ to get

$F(A) = (k_{1j} a_j + k_{2j} = d_{1j})_{j=[1,n]}$

and $F(E) = (k_{1j} e_j + k_{2j} = d_{2j})_{j=[1,n]}$;

Compute $(g_j = (d_{1,j} > d_{2j}))_{j=[1,n]}$ to get $G(g_j)_{j=[1,n]}$;

• Send G to P_1.

@P_1:

• Get $H = (h_j)_{j=[1,n]}$ according to Table 2 to determine whether there is an element greater than 1: $h_j = 1$ implies $b_j > 1$.

Table 2. Relations of g_j, k_{1j}, c_j and h_j

g_j =1		g_j = 0	
$k_{1j} > 0$	$k_{1j} < 0$	$k_{1j} > 0$	$k_{1j} < 0$
$h_j = c_j$	$h_j = 1-c_j$	$h_j = 1-c_j$	$h_j = c_j$

The security analysis of Protocol 2 is in section 5.1.

4.5 Re-encrypting the Valid Ballots

Any stateless encryption with sematic security, like Paillier encryption, involves randomness in the ciphertext, and the random factor can be used by the voter as a piece of evidence that a certain vote has been cast [10]. To prevent the voter from proving the third party what he has voted, the VC re-encrypts the ballot before forwarding it.

For the valid ballot from v_i, VC chooses n random numbers $(r'_{ij} \in Z^*_N)_{j=[1,n]}$ and generates the ciphertexts of 0: $[0_j]_{MC} = r'^N_{ij} \bmod N^2$, written as R_{ij}, to be used to refresh the permutated ballot as $[b_{\pi(j)}]' = [b_{\pi(j)}] \times_c [0_{\pi(j)}]$. Then, VC signs on the fresh ballot and encrypts it with pk_{CC} and sends it to CC. Next, an order number t_i, just the index of the valid ballot, and the random numbers $R_i = (R_{ij})_{j=[1,n]}$ are packed as a 'receipt', and signed by the VC through DVS. Finally, the receipt and the signature are packed and encrypted with pk_i and sent to the voter v_i.

4.6 Counting the Votes

The CC publishes an initialized list $P_0 = \{1, \ldots, 1\}$ on the BB before he receives the first ballot. When CC receives the i-th encrypted fresh ballot, he decrypts it firstly to get $([b_{\pi(j)}]')_{j=[1,n]}$, and verifies the signature. If it is verified, he publishes the ballot on the BB, and computes $P_i = (P_{i-1} \times_c [b_{\pi(j)}]')_{j=[1,n]}$ by element. When the number of voters reaches the specified number, or the voting deadline arrives, CC signs on the aggregated ballots as $P_f = (\prod [b_{\pi(j)}]')_{j=[1,n]}$, and sends it to MC.

The MC decrypts P_f by element after verifying the signature and gets $dP_{j,j\in\{1,\ldots,n\}} = D_{sk}(P_f)$. The result is published on the BB, and VC is informed to announce the permutation π. The reverse of π on the decryption result $\pi^{-1}(dP_j)$ is exactly the final voting result for j-th candidate. The MC also publishes the decryption key on BB in the end.

4.7 Verification

After the permutation and the decryption key are published on the BB, the voting result can be verified. Verification includes universal verification and individual verification.

In the proposed scheme, the universal verification is carried out on the BB when the result is published, anyone can aggregate the votes by homomorphic operation, and decrypt the voting result with the permutation π and the decryption key.

For the individual verification, when the voter v_i who has cast a valid ballot receives a 'receipt', he decrypts it and verifies the signature. If it is verified, the 'receipt' including $R_i = (R_{ij})_{j=[1,n]}$ and t_i is taken out. The voter only needs to encapsule his original ballot as $([b_j]')_{j=[1,n]} = ([b_{ij}] \times_c R_{ij})_{j=[1,n]}$ and shuffle it using π to get $([b_{\pi(j)}]')_{j=[1,n]}$, and check whether the t_i-th item published on the BB, is identical to $([b_{\pi(j)}]')_{j=[1,n]}$ he computed. Then he can verify whether his ballot is counted correctly.

5 Security Analysis

According to the design goals (section 3.3) of the e-voting scheme, we prove the correctness and privacy of the **Protocol** 1 and **Protocol** 2 firstly. Then, we analyze the proposed e-voting system satisfying the general requirements of security and the receipt-freeness.

5.1 Cheat-resistance/Format Verifiability

The cheat-resistance or format verifiability means that voters cannot cheat in casting their ballots, and the ballots with incorrect format will be rejected. According to **Protocol** 1 and **Protocol** 2, the ballot that consists of more 'approval's or with a choice $b_j > 1$ will be detected. We have the following theorems on the protocols.

Theorem 1. Assume the cryptosystem is semantically secure, **Protocol 1** securely computes the functionality $F_{com}([x]_{pk}, sk) = (c, \perp)$ in the presence of semi-honest parties without any collision, where '⊥' denotes 'no output'.

Proof. The correctness is immediate. The comparison $com = 1$ means $d_1 \geq d_2$, or $k_1x+k_2 \geq k_1y+k_2$. So $x \geq y$, or $c = com = 1$ holds if $k_1 > 0$, and $x < y$, or $c = 1\text{-}com = 0$ holds if $k_1 < 0$. Similarly, $com = 0$ means $d_1 < d_2$, or $k_1x+k_2 < k_1y+k_2$, so $x < y$, or $c = com$, is concluded if $k_1 > 0$, and $x \geq y$, or $c = 1\text{-}com$ holds if $k_1 < 0$.

We proceed to prove the privacy. Since the comparison result of x and y is determined, and thus it suffices to use a simpler formulation of security [35] in a simulation-based paradigm. We construct a separate simulator for each party: S_1 for P_1's view and S_2 for P_2's view, defined as

$$\{(S_1(1^\mu, [x], c))\}_{[x],sk;\mu\in N} \underline{\overset{c}{=}} \{(view_1^\pi([x], sk), c)\}_{[x],sk;\mu\in N}$$

$$\{(S_2(1^\mu, [x], \perp))\}_{[x],sk;\mu\in N} \underline{\overset{c}{=}} \{(view_2^\pi([x], sk), \perp)\}_{[x],sk;\mu\in N}$$

Here, $\{(view_i^\pi([x], sk), c/\perp)\}_{[x],sk;\mu\in N}$ with $i \in \{1,2\}$ denotes the view of the i-th party in the real execution of the protocol, μ is the security parameter, and '$\underline{\overset{c}{=}}$' means "computationally indistinguishable".

Since the party P_2 has no output, it is easier to simulate the view and we firstly consider the construction of S_2. From the protocol, the view of P_2 is exressed as {sk, ([f(x)], [f(y)])}. In the semi-honest model, the task is to simulate the view ([f(x)], [f(y)]). So, S_2 needs to choose a random $x' \in Z_N$, encrypts x' and y, and chooses a linear function $f'(z) = k_1'z + k_2'$ to generate $[f'(x')]$ and $[f'(y)]$ as prescribed in the protocol, and output the encryption ($[f'(x')]$, $[f'(y)]$) to his view.

Analysis: Although the ciphertext tuple ($[f'(x')]$, $[f'(y)]$) is not necessarily identical to ([f(x)], [f(y)]), they are indistinguishable due to the semantic security of encryption system and the same construction.

Next, we consider the case that P_1 is corrupted. Observe that P_1's view includes its input, the incoming messages, and the output, which can be expressed as {[x],

com, *c*}. Given P_1's input [*x*] and output *c*, S_1 only needs to simulate *com*, so that *c* can be derived from *com* correctly. S_1 can work as follows: flip a coin to get σ; and if σ=1, set *com′* = *c*, else set *com′* = 1 − *c*; *com′* is output to his view finally.

Analysis: The above simulation for S_1's view is {[*x*], *c*, *com′*}, with the only difference from the real view *com* and *com′*. In the real protocol, we know that the condition k_1>0 & *com*=1 plus k_1<0 & *com*=0 means *c*=1, while k_1>0 & *com*=0 plus k_1<0 & *com*=1 means *c*=0. The selection of k_1 without bias means the probability $P[k_1>0]=P[k_1<0]=0.5$. Therefore, the probability $P[c=com]=P[k_1>0 \ \& \ com=1]+P[k_1>0 \ \& \ com=0]=P[k_1>0]=0.5$ holds. On the other hand, S_1 sets *com′*=*c* with flipping a coin randomly, and the probability $P[c=com]=P[c=1-com]=0.5$ also holds. In summary, the view of the real adversary and the simulator S_1 yields the indistinguishability.

Thus, the proof is completed.

Theorem 2. Assume that the cryptosystem is semantically secure, **Protocol 2** securely computes the functionality $F_{\det}(([b_j]_{pk})_{j\in\{1,\ldots,n\}}, sk)=(H, \perp)$ in the presence of semi-honest parties, without any collision.

The proof of Theorem 2 is similar as that of the Theorem 1, and is omitted here.

5.2 Correctness

The correctness means the valid ballots being counted reflect the willingness of legal voters, and the voting result is correct. The proposed scheme meets the requirement.

Assume the valid ballots are cast by registered voters, each of which is in form of $blt=([b_j]_{MC})_{j=[1,n]}$. When *blt* is checked in Protocol 1, it is just held in VC and the content is unchanged. When the ballot is checked in Protocol 2, the encrypted elements are permutated by π firstly. Since all ballots are permutated as the same way, the aggregation is exactly the permutation of the tally result. The reverse π^{-1} is used to permutate and recover the aggregation result after all ballots are counted. That implies the permutation would not change the votes.

When VC is convinced that a ballot is correct in format, he re-encrypts the ballot $[b_{ij}]'=[b_{ij}]\times_c[0_{ij}]$, which does not change the value of the b_{ij}. The above operations are correct, due to the homomorphism property of the Paillier cryptosystem. In the end, the voting result is achieved by decrypting the aggregation of ballots by element, which is just the sum of 'approval's of each candidate.

5.3 Privacy

The privacy means each vote content being private to the voter himself and cannot be known by any other members until the voting end.

When a ballot is cast, it is encrypted and packed in the ciphertext EB_i, anyone including the eavesdropper cannot know the content. Even the VC cannot know the value of b_j, for each element is encrypted by pk_{MC}. Thanks for the privacy of the Protocol 1 and Protocol 2, the ballot content is private during the protocols of checking the format. The re-encrypted ballot sent to CC is encrypted by the pk_{CC} in the communication, meaning no one can decrypt the ballot except CC. And the re-encrypted ballot may appear on the BB, but MC cannot know the real votes of each candidate although he holds the secret key, for the permutation π. The CC also cannot know each ballot content before the end of the voting for the encryption. In such a way, the privacy of the vote content is guaranteed during the voting process.

5.4 Voter's Eligibility and Uniqueness

The voters' eligibility means only the legal voter can vote, and uniqueness requires each voter can vote only once. The properties depend on the security of signature scheme used in the voting credential. If the signature is unforgeable, the proposed scheme meets the requirements of eligibility and uniqueness.

In this scheme, a voter has to register with MC before casting a ballot. During the registration, the message from the voter to MC is encrypted by pk_{MC}, which means no one except MC can know the content, or the legal ID cannot be stolen. The voting credential to the voter is signed by MC and encrypted by pk_i, which means eavesdroppers cannot get the credential or *TID*. Anyone who pretends to be a legal voter has to forge a valid signature of MC, while the security of signature guarantees the forgery is hard.

The uniqueness is dependent on the security of $list_C$, which is created by MC and updated only by VC. A voter who tries to vote twice by setting the corresponding TID_i as 0 in $list_C$, or by making a fake credential, will fail, because the element of $list_C$ is also signed by MC. In the scheme, we assume that VC and the voters cannot collude. Thus, the uniqueness is satisfied.

5.5 Verifiability

The verifiability implies a legal voter can verify whether his ballot being counted correctly, and the final voting result can be verified publicly.

Since the permutation π and the decryption key are announced in the voting end, anyone can aggregate the published ballots and verify the final result by decryption and permutation. That means the universal verifiability is fulfilled. On the other hand, the encryption of the initial ballot $([b_j]_{MC})_{j=[1,n]}$ is through the random number $r_{ij}\in Z^*_N$, i.e. $[b_j] = g^{b_j}r_{ij}^N \bmod N^2$. If the ballot is regarded valid, the permutated ballot is re-encrypted as $([b_{\pi(j)}]'_{MC} = [b_{\pi(j)}]\times_c r_{i\pi(j)})_{j=[1,n]}$. Since the fresh ballot is published on the BB, it is easy for a legal voter to verify that the computed $[b_{\pi(j)}]'_{MC}$ is just the t_i-th item on the BB.

5.6 Receipt-freeness

Formally, the definition of receipt-freeness was given in [29] as following.

Definition (Receipt-Freeness) [29]. A receipt is a witness w' which allows a third party to verify, in an unambiguous way, the vote of a voter v_i:

$$\exists! v_i, s.t. \exists w', s.t. R'(B, V, v_i, w') = 1 \quad \textbf{(1)}$$

A voting scheme achieves the receipt-freeness property if there is no such a relation *R′*, or the witness *w′* is hard to compute.

The receipt-freeness in this scheme is achieved mainly because the 'receipt' R_i from the VC is signed with DVS, such that the voter v_i himself can generate a valid 'receipt' based on a published ballot $([b_{pj}])_{j=[1,n]}$. Although the randomness in the encryption can be a proof for the voter to show to others, the re-encryption eliminates the initial randomness. And the fresh random number is signed by VC with DVS, thus the proof becomes ambiguous.

Informally, when the voting result is announced, followed with the permutation π and the decryption key, the voter v_i can deduce each ballot and decrypt it to get the corresponding content. Thereby, an arbitrary published ballot $([b_{pj}])_{j=[1,n]}$ can be claimed to be cast by him easily, which implies the receipt-freeness.

It is noticed that we also assume the voter-buyer is not allowed to communicate with the voter during the voting process, similar as [10]. The voter who wants to sell his ballot has to require additional channel to communicate with the buyer. Therefore, even an encrypted ballot is sent to the buyer ahead of the voting result being published, and the buyer will decrypt it later, the buyer cannot believe the received ballot is just the one that the voter cast to VC.

6 Performance

The costs of computation and communication are mainly analyzed. Since the voters register before the voting, the initialization or preparation for the voting is excluded from consideration. We only consider the cost during the voting process.

6.1 Computation Cost

In this scheme, the main computation cost involves creating a ballot for each voter and checking the ballots for the VC and MC. Since voters can create their votes in a distributed manner, we just consider the time cost for one ballot. The time includes encrypting each choice in creating a ballot, and checking the format of the ballot by operations of decryption and comparison, and re-encrypting the ballot. For n candidates, each voter will compute n ciphertexts. The time cost in verification is at least $2n$ times of decryption due to the permutation and the comparison. Since the number of candidates affects the times of encryption and decryption, we consider the time cost with the number of candidates. For Paillier system, the length of the big integer N will influence the time cost in encryption and decryption, and we also consider impact of N on the time cost. We evaluate the cost via a custom simulator built in Java, and the evaluations were performed on a PC with 3.60 GHz ten-cores processor and 32G memory. The number of candidates and the length of N affect the time cost, as shown in Figure 2.

In this figure, the re-encrypting time is almost same as creating a ballot. That is because the re-encryption actually encrypting n '0', and the time in creating a vote is also encrypting n bits. The checking time is double the creating time or so, that is because the checking involves the decrypting the double number of elements and comparing them. The impact of length of N is obvious. When the length is 1024, and for 100 candidates, the checking time is no more than 500ms, while the time increases to 2300ms or so when the bit length is set as 2048. The underlying reason is that the larger N, the time in the modular computation in decryption will be longer. Another phenomenon that can be seen from this figure is that the time cost is almost linear with the number of the candidates, which is expected.

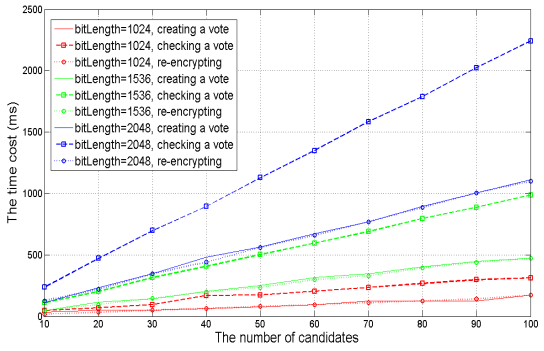


Figure 2. Time cost with the number of candidates

6.2 Communication Cost

In terms of communication, we mainly address the communication cost during the e-voting process. The communication in the scheme may occur between the voters and VC, VC and MC, VC and CC, CC and MC. Since the voters have to cast their ballots to VC, the transmitted data involves n encrypted choices, a credential, and a signature. During the voting format check, VC sends $2n + 2$ ciphertexts in Protocol 1 and Protocol 2 and the signature to MC, and MC will send back n+1 bits for the comparison results. For a valid ballot, VC sends back a ciphertext of n random numbers to each voter, and sends the re-encrypted ballot to CC for n ciphertexts. For the size of the ciphertexts and signature depends on the schemes and the parameters chosen, we roughly list the communicated data related to the number of encryption and signature used, as Table 3.

Table 3. Communication overhead during the voting phase

Cost	Casting a vote	Checking a vote	Counting votes
Encryption (E)	n+1	2n+2	n
Signature (S)	1	1	1
Others (bits)	0	n+1	0

7 Conclusion

In this paper, we propose an e-voting scheme that can verify the ballot in format without disclosing its content using the homomorphic encryption. The legal voter is registered and authenticated by the managing center and issued a unique temporary voting credential, that is used to ensure the legality during the voting. The ballot is encrypted with homomorphic cryptosystem and the voting center can check whether the ballot contains more 'approval's than the specified number, or if there exists more than one

'approval' for the same candidate in a ballot. In such a way, the fraud behavior from the voters can be detected and the invalid ballots will be rejected. The voting center sends the valid ballots to the counting center for tallying the voting result. The managing center can decrypt the final voting result correctly without knowing each ballot. In addition to the correctness, the legality, the privacy, the verification, and the receipt-freeness, the proposed scheme achieves the format verifiability for each ballot.

The possible improvement for our scheme lies in the coding of the ballot and the non-interactive proofs for the ballot format. First, if the ballot of n choices is coded as a single number, then only one encryption is needed for creating a ballot and the efficiency will be improved. Another improvement is the non-interactive proofs to verify the format of the ballots. Both of them are the future work of this paper.

References

[1] D. Chaum, Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms, *Communications of the ACM*, Vol. 24, No. 2, pp. 84-90, February, 1981. https://doi.org/10.1145/358549.358563

[2] P. Golle, S. Zhong, D. Boneh, M. Jakobsson, A. Juels, Optimistic Mixing for Exit-polls, *Advances in Cryptology-8th International Conference on the Theory and Application of Cryptology and Information Security*, Queentown, New Zealand, 2002, pp. 451-465.

[3] K. Peng, A general and efficient countermeasure to relation attacks in mix-based e-voting, *International Journal of Information Security*, Vol. 10, No. 1, pp. 49-60, February, 2011. https://doi.org/10.1007/s10207-010-0122-1

[4] J. P. Allepuz, S. G. Castell, Universally Verifiable Efficient Re-encryption Mixnet, *4th International Conference on Electronic Voting 2010*, Castle Hofen, Bregenz, Austria, 2010, pp. 241-254.

[5] D. Aranha, C. Baum, K. Gjøsteen, T. Silde, Verifiable Mix-Nets and Distributed Decryption for Voting from Lattice-Based Assumptions, *Proceeding of 2023 Conference on Computer and Communications Security (CCS)*, Cophenhagen, Denmark, 2023, pp. 1467-1481. https://doi.org/10.1145/3576915.3616683

[6] A. Fujioka, T. Okamoto , K. Ohta, A Practical Secret Voting Scheme for Large Scale Elections, *Workshop on the Theory and Application of Cryptographic Techniques*, Gold Coast, Queensland, Australia, 1992, pp. 244-251. https://doi.org/10.1007/3-540-57220-1_66

[7] D. Chaum, Blind Signature System, *Advances in Cryptology, Proceedings of CRYPTO'83*, Santa Barbara, California, USA, 1983, pp. 153.

[8] M. Kumar, C. P. Katti, P. C. Saxena, A Secure Anonymous E-Voting System Using Identity-Based Blind Signature Scheme, *International Conference on Information Systems Security-13th International Conference*, Mumbai, India, 2017, pp. 29-49. https://doi.org/10.1007/978-3-319-72598-7_3

[9] J. K. Liu, D. S. Wong, A Restricted Multi-show Credential System and Its Application on E-voting, *Information Security Practice and Experience, First International Conference*, Singapore, 2005, pp. 268-279. https://doi.org/10.1007/978-3-540-31979-5_23

[10] S. S. M. Chow, J. K. Liu, D. S. Wong, Robust Receipt-Free Election System with Ballot Secrecy and Verifiability, *Proceedings of the Network and Distributed System Security Symposium*, San Diego, California, USA, 2008, pp. 81-94.

[11] S. S. M. Chow, W. Susilo, T. H. Yuen, Escrowed Linkability of Ring Signatures and Its Applications, *Progress in Cryptology - First International Conference on Cryptology, Vietnam*, Hanoi, Vietnam, 2006, pp. 175-192. https://doi.org/10.1007/11958239_12

[12] X. Yang, X. Yi, S. Nepal, A. Kelarev, F. Han, A Secure Verifiable Ranked Choice Online Voting System Based on Homomorphic Encryption, *IEEE Access*, Vol. 6, pp. 20506-20519, March, 2018. https://doi.org/10.1109/ACCESS.2018.2817518

[13] I. Chillotti, N. Gama, M. Georgieva, M. Izabachne, A Homomorphic LWE Based E-voting Scheme, *Post-Quantum Cryptography-7th International Workshop*, Fucuoka, Japan, 2016, pp. 245-265. https://doi.org/10.1007/978-3-319-29360-8_16

[14] W. Qu, L. Wu, W. Wang, Z. Liu, H. Wang, An Electronic Voting Protocol Based on Blockchain and Homomorphic Signcryption, *Concurrency and Computation: Practice and Experience*, Vol. 34, No. 16, Article No. e5817, July, 2022. https://doi.org/10.1002/cpe.5817

[15] Y. Yang, Y. Zhao, Q. Zhang, Y. Ma, Y. Gao, Weighted Electronic Voting System with Homomorphic Encryption Based on SEAL, *Chinese Journal of Computers*, Vol. 43, No. 4, pp. 711-723, April, 2020. https://doi.org/10.11897/SP.J.1016.2020.00711

[16] M. Kumar, S. Chand, C. P. Katti, A Secure End-to-End Verifiable Internet-voting System Using Identity-based Blind Signature, *IEEE Systems Journal*, Vol. 14, No. 2, pp. 2032-2041, June, 2020. https://doi.org/10.1109/JSYST.2019.2940474

[17] A. Russo, A. F. Anta, M. I. G. Vasco, S. P. Romano, Chirotonia: A Scalable and Secure E-voting Framework Based on Blockchains and Linkable Ring Signatures, *2021 IEEE International Conference on Blockchain*, Melbourne, Australia, 2021, pp. 417-424. https://doi.org/10.1109/Blockchain53845.2021.00065

[18] Y. Liu, Q. Zhao, E-voting Scheme Using Secret Sharing and K-anonymity, *World Wide Web*, Vol. 22, No. 4, pp. 1657-1667, July, 2019. https://doi.org/10.1007/s11280-018-0575-0

[19] S. F. Shahandashti, F. Hao, DRE-ip: A Verifiable E-voting Scheme Without Tallying Authorities, *21st European Symposium on Research in Computer Security*, Heraklion, Greece, 2016, pp. 223-240. https://doi.org/10.1007/978-3-319-45741-3_12

[20] S. Panja, B. Roy, A Secure End-to-End Verifiable E-voting System Using Blockchain and Cloud Server, *Journal of Information Security and Applications*, Vol. 59, Article No. 102815, June, 2021. https://doi.org/10.1016/j.jisa.2021.102815

[21] C. Spadafora, R. Longo, M. Sala, A Coercion-Resistant Blockchain-based E-voting Protocol with Receipts, *Advances in Mathematics of Communications*, Vol. 17, No. 2, pp. 500-521, April, 2023. https://doi.org/10.3934/amc.2021005

[22] H. Li, Y. Li, Y. Yu, B. Wang, K. Chen, A Blockchain-based Traceable Self-tallying E-voting Protocol in AI Era, *IEEE Transactions on Network Science and Engineering*, Vol. 8, No. 2, pp. 1019-1032, April-June, 2021.

https://doi.org/10.1109/TNSE.2020.3011928

[23] Y. Abuidris, R. Kumar, T. Yang, J. Onginjo, Secure Large-scale E-voting System Based on Blockchain Contract using a Hybrid Consensus Model Combined with Sharding, *ETRI Journal*, Vol. 43, No. 2, pp. 357-370, April, 2021. https://doi.org/10.4218/etrij.2019-0362

[24] H. Yi, Securing E-voting Based on Blockchain in P2P Network, *EURASIP Journal on Wireless Communications and Networking*, Vol. 2019, Article No. 137, May, 2019. https://doi.org/10.1186/s13638-019-1473-6

[25] J. C. P. Carcía, A. Benslimane, S. Boutalbi, Blockchain-Based System for E-voting Using Blind Signature Protocol, *IEEE Global Communications Conference*, Madrid, Spain, 2021, pp. 1-6. https://doi.org/10.1109/GLOBECOM46510.2021.9685189

[26] R. Longo, C. Spadafora, Multiple Candidates Coercion-Resistant Blockchain-Based E-voting Protocol with Receipts, *IACR Cryptology ePrint Archive*, No. 851, pp. 1-13, June, 2021. https://ia.cr/2021/851

[27] A. Juels, D. Catalano, M. Jakobsson, Coercion-Resistant Electronic Elections, *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, Alexandria, VA, USA, 2005, pp. 61-70. https://doi.org/10.1145/1102199.1102213

[28] M. Hirt, K. Sako, Efficient Receipt-Free Voting Based on Homomorphic Encryption, *Advances in Cryptology - International Conference on the Theory and Application of Cryptographic Techniques*, Bruges, Belgium, 2000, pp. 539-556. https://doi.org/10.1007/3-540-45539-6_38

[29] B. Chevallier-Mames, P. Fouque, D. Pointcheval, J. Stern, J. Traore, On Some Incompatible Properties of Voting Schemes, in: D. Chaum (Eds.), *Towards Trustworthy Elections, New Directions in Electronic Voting*, Springer, Berlin, Heidelberg, 2010, pp. 191-199. https://doi.org/10.1007/978-3-642-12980-3_11

[30] P. Paillier, Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, *Advances in Cryptology - International Conference on the Theory and Application of Cryptographic Techniques*, Prague, Czech Republic, 1999, pp. 223-238. https://doi.org/10.1007/3-540-48910-X_16

[31] M. Jakobsson, K. Sako, R. Impagliazzo, Designated Verifier Proofs and Their Applications, *Advances in Cryptology - International Conference on the Theory and Application of Cryptographic Techniques,* Saragossa, Spain, 1996, pp. 143-154. https://doi.org/10.1007/3-540-68339-9_13

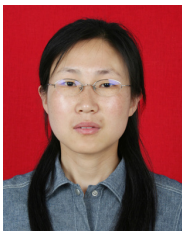
[32] J. Wang, X. Xie, Y. Chen, A Generic Construction of Designated Verifier Signature from Standard Cryptographic Algorithms, *Journal of Information Science and Engineering*, Vol. 38, No. 5, pp. 1051-1063, September, 2022. https://doi.org/10.6688/JISE.202209_38(5).0011

[33] D. Balla, P. Behrouz, P. Grontas, A. Pagourtzis, M. Spyrakou, G. Vrettos, Designated-verifier Linkable Ring Signatures with Unconditional Anonymity, *Algebraic Informatics - 9th International Conference*, Virtual Event, 2022, pp. 55-68. https://doi.org/10.1007/978-3-031-19685-0_5

[34] D. Boneh, C. Gentry, B. Lynn, H. Shacham, Aggregate and Verifiably Encrypted Signatures from Bilinear Maps, *Advances in Cryptology - International Conference on the Theory and Applications of Cryptographic Techniques*, Warsaw, Poland, 2003, pp. 416-432. https://doi.org/10.1007/3-540-39200-9_26

[35] Y. Lindell, How to simulate it - A tutorial on the simulation proof technique, in: Y. Lindell (Eds.), *Tutorials on the Foundations of Cryptography*, Springer, Cham, 2017, pp. 277-346. https://doi.org/10.1007/978-3-319-57048-8_6

Biographies



Yuhong Sun received the M. E. degree in Shandong University, Jinan, China, in 2005. Currently, she is an associate professor at Qufu Normal University, Rizhao, China. Her current research interests include information security theory and application.

Jiatao Wang received the M. E. degree in Qingdao University of Technology, Qingdao, China, in 2005. Currently, he is an associate professor at Shandong Water Conservancy Vocational College, Rizhao, China. His current research interests include geotechnical engineering.

Fengyin Li received the Ph. D. degree in Shandong Normal University, Jinan, China, in 2014. Currently, she is a professor at Qufu Normal University, Rizhao,China. Her current research interests include information security theory and application, privacy protection technology.