Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Multi party confidential verifiable electronic voting scheme based on blockchain

Xusheng Wang[1*], Tao Feng[2], Chunyan Liu[2] and Junli Fang[2]

Abstract

The current electronic voting system ensures the confidentiality of the ballot but does not explicitly address the anonymity of the voter's identity. The voting process is entirely managed by smart contracts, which diminishes the efficiency of contract execution and poses challenges for achieving large-scale voting. There is a lack of comprehensive explanation regarding the voting process and the verifiability of ballot information. This article introduces a multi-party secure verifiable electronic voting scheme based on blockchain technology. The solution uses IPFS distributed file system to alleviate the problem of limited block storage. The task allocation and vote counting processes are managed separately through management and computation contracts to improve efficiency. Ring signature technology is used to ensure the anonymity of voting identities. The vote counting process uses a key-sharing system to ensure the privacy of the votes, and the calculation results are encrypted and uploaded to the blockchain, making them verifiable. Finally, a security proof and performance analysis of the scheme were conducted.

Keywords Blockchain, Electronic voting, IPFS, Ring signature, Verifiable

Introduction

With the continuous development of information technology, electronic voting has become an important component of e-government. With the widespread application of Internet and cryptography technologies, the efficiency of electronic voting has been continuously improved, saving many costs and making the process more convenient. Building on this foundation, many researchers have proposed numerous electronic voting schemes based on techniques such as homomorphic encryption, hybrid networks, blind signatures, and ring signatures.

Chaum et al. achieved anonymous voting by associating voters with ballots through a hybrid network protocol [1], but this scheme cannot resist ciphertext attacks. To prevent ciphertext leakage, Boneh and Golle et al. improved the voting protocol based on hybrid network technology [2] by using probability theory to statistically verify the probability of successfully searching for cheating servers, but this increased the computational complexity of the protocol. Fujioka, Okamoto, and others proposed the famous FOO voting protocol [3]. This scheme is based on blind signature and bit commitment technology, suitable for large-scale voting, but it is inefficient and does not allow for vote abandonment. Roffen et al. proposed an electronic voting scheme without receipts based on the FOO voting protocol [4], which further improved the FOO voting protocol, ensuring anonymity, verifiability, and non-receipt of votes, and allowing for waivers. Reference [5] designed an application-oriented network electronic voting protocol based on homomorphic encryption. Chandra Priya, J et al. proposed an electronic voting scheme based on fully homomorphic encryption [6].

In recent years, with the application of blockchain technology in electronic voting, the voting process has been simplified, and reliance on third-party institutions

*Correspondence:
Xusheng Wang
wangxs@gsau.edu.cn
[1] School of Information Science and Technology, Gansu Agricultural University, Lanzhou 730070, China
[2] School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China

has been reduced. For the first time, an electronic voting protocol based on Bitcoin is proposed [7]. This scheme achieves voting fairness through an incentive mechanism. Many blockchain-based electronic voting protocols [8, 9], despite utilizing key blockchain technologies, still require third-party assistance for voting. The voting protocol proposed in reference [10] allows only a small number of votes and does not permit abstention. References [11, 12] utilize smart contracts instead of third-party institutions to conduct electronic voting, achieving a privacy-preserving, self-counting electronic voting scheme. Chondros et al. [13] adopted a distributed electronic voting scheme to meet the needs of large-scale voting. To provide accurate voting results, a practical Byzantine fault-tolerant (PBFT) consensus mechanism is used to ensure that votes are not modified or corrupted [14]. A distributed electronic voting scheme using blockchain technology replaces third-party institutions [15], with vote counting completed through smart contracts, ensuring the security and fairness of the voting process. To realize a voting mechanism for multiple candidates suitable for various voting scenarios, a privacy protection scheme for double-signature electronic voting based on consortium chains is proposed [16]. Taş, Baudier, and others conducted research and analysis on blockchain-based voting [17, 18], addressing problems under specific and general conditions, such as the need to improve the security and scalability of remote participation, and emphasized the importance of trust and human factors in voting.

For the voting schemes discussed in the above literature, there are several issues: firstly, some schemes use third-party counting methods, which not only reduce voting efficiency but also introduce security risks. Secondly, while some schemes employ self-counting of votes, they store all voting information on the blockchain and rely on smart contracts for coordinated processing, This approach undoubtedly increases the cost of contracts and block storage, thereby limiting the scale of voting. Lastly, while most schemes focus primarily on ensuring the legitimacy and security of voting data, they pay little attention to the anonymity of voters' identities.

In light of the potential problems mentioned above, this paper proposes a secure, multi-party verifiable electronic voting scheme based on blockchain technology, establishes a voting model, and designs a voting algorithm. Smart contracts—encompassing management and computing contracts—replace third-party institutions, and the vote counting task is distributed across the computing network via the management contract to accommodate a certain scale of electronic voting. The management contract verifies the legitimacy and anonymity of voters during the registration stage. The computation contract calculates the vote results for each candidate, sorts them, and stores them on the blockchain, facilitating the search for and verification of votes. This scheme is suitable for elections with multiple candidates. Finally, the algorithmic complexity of this scheme is analyzed and compared, and security proofs and performance analyses are conducted based on the DBDH assumption and the random oracle model.

Related work

Ring signature

Ring signatures [19] are a special type of group signature. There is no trusted center and no group establishment process. For the verifier, the signer remains completely anonymous. The name reflects its unique ring structure. The real signer uses the public keys generated by other possible signers to create a ring with a break, and then connects the break with their own private key to form a complete ring. Any verifier can use the public keys of the ring members to verify who generated the ring signature. Based on the calculation result, they verify whether the signature is valid and then accept or reject the signature. Generally, ring signatures include the following three algorithms:

(1) Key generation algorithm $R-Gen$: The signer in the ring uses the key generation algorithm to generate a public–private key pair (pk, sk) and a set of public keys $PK = (pk_1, pk_2, \dots, pk_n)$.
(2) Ring Signature Algorithm $R-Sig$: This is the algorithm for generating ring signatures. The inputs include the message m to be signed, the public key set PK of n signers in the ring, and the private key sk of the actual signer within the ring. The algorithm then performs $\sigma \leftarrow sig(sk, m, PK)$ ring signature calculation on message m.
(3) Ring signature verification algorithm $R-V\mathrm{er}$: Input the signed message m and the result σ of the ring signature calculation performed on the message. The signature is verified and calculated $\mathrm{ver}ify(\sigma, m, PK)$, and the verification result is output. The verification passed is 1, and the failed verification is 0.

Assumptions for difficult problems and public key encryption

Definition 1: Assuming groups G_1 and G_2, and a mapping $e: G_1 \times G_2 \to G_2$, challenger B randomly selects $a, b, c, z \in Z_q$ and generates two quintuples, which are:

$$T_1 = (p, A = p^a, B = p^b, C = p^c, Z = e(p,p)^z)$$
$$T_2 = (p, A = p^a, B = p^b, C = p^c, Z = e(p,p)^{abc})$$

Randomly select a byte $\sigma \in \{0, 1\}$; if $\sigma = 0$, output the quintuple T_1; otherwise, output T_2.

Adversary A outputs σ^* as a conjecture about σ based on the obtained quintuples T_1, T_2. The advantage of adversary A is $Adv_A = |\Pr[\sigma^* = \sigma] - 1/2| \geq \varepsilon$ (an undeniable advantage ε), where the probability depends on the random selection of $a, b, c, z \in Z_q$ and the use of random bits.

Definition 2: A public key encryption scheme [20] is based on the probabilistic polynomial-time algorithm PPT and consists of the following four components:

System generation algorithm $Setup(1^\lambda)$: This algorithm takes safety parameter λ as input and outputs system parameter ω.

Key Generation Algorithm $Gen(\omega)$: This algorithm takes the system-generated parameter ω as input and outputs a public–private key pair (pk, sk).

Encryption Algorithm $En(pk, M, r)$: This algorithm takes the public key pk, plaintext message M, and system parameter ω as inputs, and outputs the ciphertext C.

Decryption Algorithm $De(sk, C)$: This algorithm takes the private key sk and the ciphertext C as inputs, outputs the plaintext M, otherwise it fails.

Reference [21] demonstrated the correctness of the aforementioned public key encryption scheme, which is grounded in number-theoretic assumptions, and performed a security analysis of the scheme under the framework of adversary attack games within a random oracle model. An advantage function was defined for the scheme, and it was shown that adversaries operating within the random oracle model possess negligible advantages during two stages of interrogation, thereby satisfying the indistinguishability under chosen-ciphertext attacks (IND-CCA) security, also referred to as IND-CCA2 security.

Security requirements for electronic voting

The electronic voting system conserves numerous resources, ensures fairness and transparency, supplants methods like dictatorship and drawing lots, and has become the most critical decision-making method within e-government. It encapsulates democracy and transforms individual values into collective ones. Electronic voting is conducted via a computer network. There are inherent risks in the network environment, making security a crucial aspect of electronic voting. A secure electronic voting protocol should satisfy the following key requirements:

1). Privacy: To protect the privacy of voters, including the identity of voters and candidates, as well as ballot data, ensuring that this information is accessible only to the voter.

2). Legality: Voters can only participate if their identity has been confirmed through registration and authentication.

3). Correctness: Ballots must have a uniform format to ensure accurate counting.

4). Completeness: All legal votes from voters must meet correctness criteria before proceeding to the vote counting stage.

5). Uniqueness: Each voter has one opportunity to vote, with no allowance for repeated voting.

6). Verifiability: Voting results and voter identities must be verifiable to prevent unauthorized individuals from falsifying or tampering with the outcomes, in accordance with regulations.

7). No Receipts: To prevent alterations to the voting results and ensure the legality of the vote, candidates should not receive proof of their votes, and voters should not be able to create receipts that contradict their actual votes.

8). Anti-Attack: This includes preventing collusion and attacks among voters to view ballot information, preventing candidates from conspiring with voters for canvassing, and destroying ballot information to render voting impossible. In these attack scenarios, the attacker is considered a malicious participant.

9). Fairness: The voting outcome must remain unaffected by any of the aforementioned factors.

System model

Traditional voting model

As the scale of voting continues to expand, traditional voting methods reveal numerous shortcomings, such as the low efficiency and high cost associated with manual vote counting, and the accuracy of the results cannot be guaranteed. The privacy of traditional voting and the accuracy of vote counting both rely on the integrity of the voter (or a trusted third party). Due to the significant power vested in the voter in traditional voting schemes, the voter's identity and ballot information may be compromised, affecting their willingness to participate and providing candidates with opportunities for canvassing, which can compromise the fairness of the election results. The traditional election model is illustrated in Fig. 1 below:

BC-MPC voting model

To prevent fraudulent conduct in voting and ensure the fairness and impartiality of electronic voting, an electronic voting scheme based on the LUC secret system and secret sharing is proposed [22]. This scheme meets

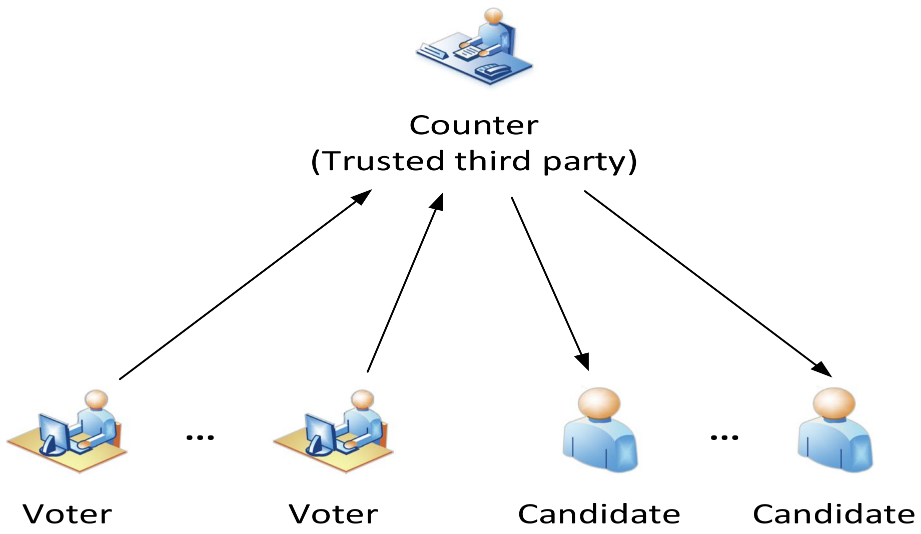

Fig. 1 Model of traditional electronic voting scheme

the security requirements of anonymity, no receipt, verifiability, and fairness, and enhances voting efficiency. Given the characteristics of decentralization, openness, tamper resistance, anonymity, and traceability, blockchain technology can address current electronic voting issues related to anonymity, transparency, and public verification.

This section introduces our proposed blockchain-based secure multi-party voting calculation model, which primarily involves six entities: voters, candidates, MPC contracts (including management and computing contracts), computing networks, the blockchain, and the IPFS distributed file system. The MPC contract is divided into two parts: the management contract and the counting contract. Typically, the electronic voting scheme encompasses four stages: registration, voting, counting, and verification. The specific structure of the program is depicted in Fig. 2 below.

First, candidates and voters are registered, and the management contract verifies the legitimacy of their identities based on the format and length of the input characters to ensure that each registered individual is genuine. The contract uses the voter's registered account number and password as a public–private key pair to construct a ring signature, calculate the signature, and return a unique identity post-registration. Voters use their identities to encrypt and store voting information in the IPFS distributed file system. The returned storage location is the index information obtained via the distributed hash table. Voters store the index information on the blockchain, and a consensus mechanism is employed to reach an agreement among nodes. The authorizer obtains the decryption key to view the voting information and verify its legality and validity.

Secondly, because of the synchronization mechanism, once the registration period ends, the management contract processes the voting information based on the MPC parameters (including participants' public identities, secret data, etc.), compiles the calculation script according to the data summation and sorting algorithm, carries out task allocation, and sends calculation requests to the MPC nodes within the computing network. Each node in the computing network sequentially processes the votes through multi-party computation for summation and sorting, then returns the calculation results to the management contract. The management contract encrypts the candidate's identity and the corresponding ballot information using a public key and stores them on the blockchain. The consensus mechanism verifies and updates the block information to maintain consistency, facilitating easy search and verification.

Finally, the candidate or other authorized individuals can decrypt and verify the identity using the private key and view the vote results.

Electronic voting protocol design

Protocol parameter definition

The role definitions and corresponding symbols in the voting algorithm are presented in Table 1. The roles and their functions are described as follows:

> v_i (Vote): Voters are associated with voting data, and candidates are associated with the voting options.
>
> V_i (Voters): Voters register to participate in voting after being verified by identity ID_i. The ring signa-

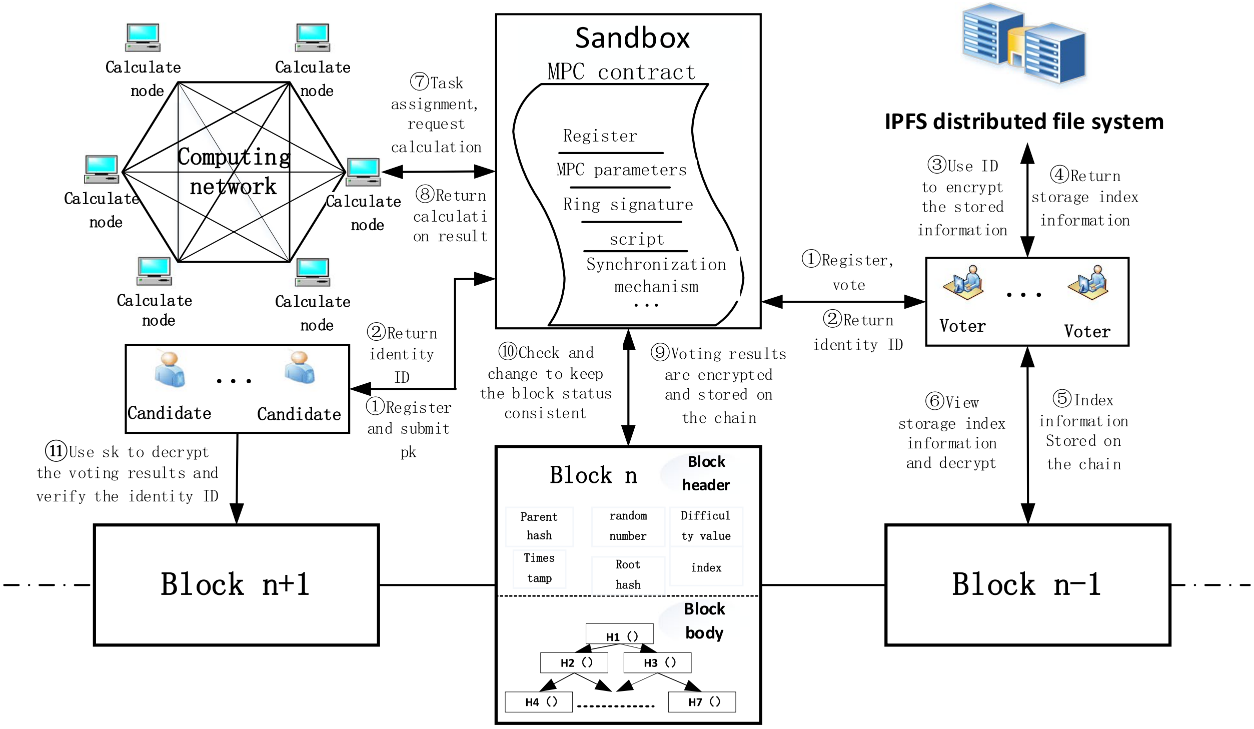
Calculate node
Calculate node
Calculate node
Calculate node
Calculate node
Calculate node
Computing network
⑦Task assignment, request calculation
⑧Return calculati on result
Sandbox
MPC contract
Register
MPC parameters
Ring signature
script
Synchronization mechanism
...
IPFS distributed file system
③Use ID to encrypt the stored information
④Return storage index information
①Register, vote
②Return identity ID
Voter
Voter
②Return identity ID
①Register and submit pk
Candidate
Candidate
⑩Check and change to keep the block status consistent
⑨Voting results are encrypted and stored on the chain
⑥View storage index information and decrypt
⑤Index information Stored on the chain
⑪Use sk to decrypt the voting results and verify the identity ID
Block n
Block header
Parent hash
random number
Difficul ty value
Times tamp
Root hash
index
Block body
H1 ()
H2 ()
H3 ()
H4 ()
H7 ()
Block n+1
Block n−1

Fig. 2 BC-MPC program model

Table 1 Symbols and definitions of voting roles

Symbol	Role definition
v_i	vote
V_i	Voter
C_i	Candidate
SC_m	Management contract
SC_t	Counting Contract
Peer	blockchain node
MPC_n	MPC node
ID_i	Unique identification

ture anonymizes the identity and interacts with the computing protocol through the contract, without human intervention.

C_i (Candidate): After the same registration as the voter is verified by identity ID_i, and the public key *pk* possessed is stored in the contract, the management contract uses the corresponding public key *pk* to encrypt the voting result on the chain, and the candidate can use it The private key *sk* is decrypted to view one's own votes.

SC_m(Management contract): Check whether the registration of the voter's identity information is legal, sign the voting information, and calculate the management of the contract. Use candidate public key *pk* to encrypt the result of the ballot and publish it on the chain.

SC_t (Vote counting contract): Accept the voter's ballot information and verify the legitimacy of the ballot. The deployment of multi-party summation and multi-party sorting calculation tasks on the votes.

Peer (Blockchain node): Contracts (management contracts and calculation contracts) are deployed in nodes to provide corresponding services. At the same time, the index information and the results of the node's votes for the candidates are encrypted and stored.

MPC_n (MPC node): The ticket counting contract assigns the summation and sorting calculation tasks to the MPC node, and the node calculates and returns the result to the contract for processing.

ID_i (Unique ID): The identity information of users (voters and candidates) is identified in the agreement, and each user only has its own unique and valid ID.

Voting steps

The main steps of the electronic voting scheme are divided into four stages: registration, voting, counting, and publishing. The sequence diagram of the scheme is presented in Fig. 3.

The specific process is as follows:

1. Registration stage

   Voter V_i and candidate C_i must register within the specified registration period. The ring signature algorithm within management contract SC_m signs the identity information, generates a unique identity ID_i related to the identity details, and verifies the legality of the signature. The identity's format is verified for consistency. If it is legal and the voter or candidate is qualified to participate in the voting process, identity ID_i is communicated back to them.
2. Voting stage
   Voter V_i confirms eligibility to vote (having received unique identity ID_i) and then casts a vote. Unique identity ID_i can vote only once and cannot make repeated votes. Utilize ring signature algorithm $\sigma \leftarrow \mathrm{sig}(\mathrm{sk}, \mathrm{m}, \mathrm{PK})$ for signature calculation; E is the information to be signed, including the identity information of candidate C_i and voter V_i, while PK is a set of public keys. Signature verification $\mathrm{verify}(\sigma, \mathrm{m}, PK)$ ensures the anonymity of identities. Concurrently, the voter uses unique identity ID_i to encrypt the voting information and stores it in the IPFS distributed file system. Each file block storing information is managed as key-value pairs through a distributed hash table. Voters store the returned storage index information on the blockchain, and the authorizer obtains the decryption key from the server to verify the voting information.
3. Counting stage
   After the management contract validates the identities and votes of candidate C_i and voter V_i, the process is handed over to the calculation contract, which allocates calculation tasks to the MPC_n node. Upon receiving a calculation request, the MPC_n node performs the relevant tasks, primarily divided into two parts: multi-party summation and multi-party sorting. Initially, the votes are summed based on candidate identity, and then sorted based on the total number of votes each candidate has received. The calculation results from the management contract's pairing and sorting are linked through the encryption

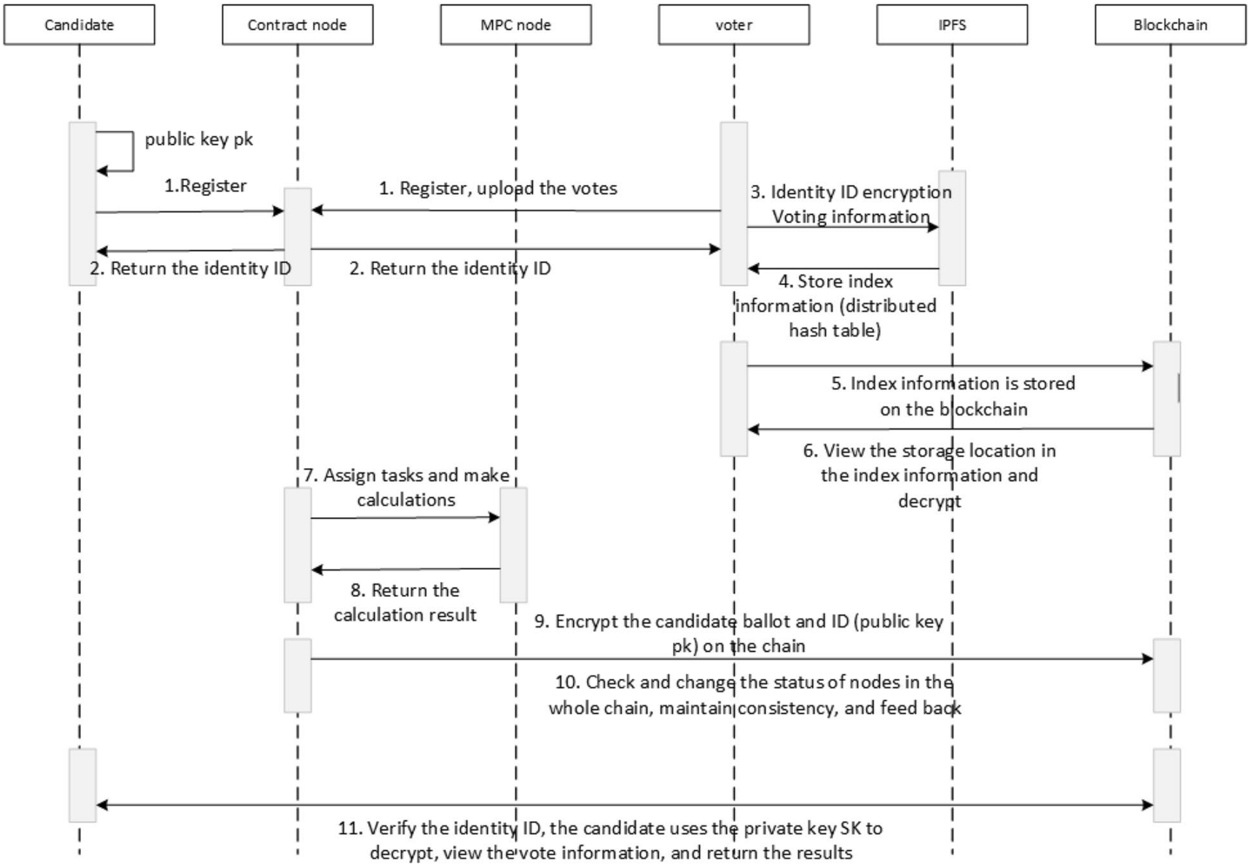
Candidate
Contract node
MPC node
voter
IPFS
Blockchain
public key pk
1.Register
1. Register, upload the votes
3. Identity ID encryption Voting information
2. Return the identity ID
2. Return the identity ID
4. Store index information (distributed hash table)
5. Index information is stored on the blockchain
6. View the storage location in the index information and decrypt
7. Assign tasks and make calculations
8. Return the calculation result
9. Encrypt the candidate ballot and ID (public key pk) on the chain
10. Check and change the status of nodes in the whole chain, maintain consistency, and feed back
11. Verify the identity ID, the candidate uses the private key SK to decrypt, view the vote information, and return the results

Fig. 3 Timing diagram of the scheme

processing with candidate's public key, ensuring the verifiability of the key calculation steps.

4. Verification stage
The candidate's vote results and corresponding identity ID_i are encrypted with the corresponding public key via the management contract and published on blockchain node *Peer*. The candidate uses their private key to decrypt the information and verifies identity ID_i to view their own vote.

Voting algorithm

Multi party data summation algorithm

The secure multi-party summation algorithm is an important component of secure multi-party computing [23, 24]. The scheme employs the secure multi-party summation algorithm to tally votes for each candidate and utilizes the secure multi-party summation protocol from the threshold key sharing system to prevent voter collusion and ensure the fairness of the voting process. For the specific steps of the algorithm, see Table 2 below:

In the algorithm, participant $R_i(1 \le i \le n)$ possesses secret data S_i and a verifiable public identity $x_j(1 \le j \le n)$, and randomly selects a polynomial $f_i(x) = S_i + \sum_{j=1}^{t-1} a_{ij}x^j$ of degree $(\mathrm{t}-1)$ to encrypt S_i. Substitute x_j into the random polynomial $f_i(x)$ for calculation, where $(\mathrm{x}_1 = 1, \mathrm{x}_2 = 2, ..., \mathrm{x}_\mathrm{n} = \mathrm{n})$, and send the calculation result to the corresponding participant R_j. Sum the data using equation $F(x_i) = \sum_{j=1}^{n} f_j(x_i)$. $(\mathrm{x}_1 = 1, \mathrm{x}_2 = 2, ..., \mathrm{x}_\mathrm{n} = \mathrm{n})$ resembles the value of the n-dimensional Vandermonde matrix $Q = (\lambda_1, \lambda_2, .., \lambda_\mathrm{n})^\mathrm{T}$; there exists an invertible matrix such that equation $(1,2, ..., \mathrm{n})(\lambda_1, \lambda_2, .., \lambda_\mathrm{n})^\mathrm{T} = (1,0, ..., 0)$ is satisfied, thereby eliminating the random value in $f_i(x)$. All participants $R_i(1 \le i \le n)$ calculate S collectively.

Table 2 Multi party data summation algorithm

```
Algorithm: multi party data summation algorithm
Input: participant$R_i$'s secret data$S_i$ and corresponding public identity$x_j$,
where $(1 \le \mathrm{i}, \mathrm{j} \le \mathrm{n})$
Output:$S$
Step:
1 Compute $f_i(x) = S_i + \sum_{j=1}^{t-1} a_{ij}x^j$
2 Compute $f_i(x_j)(x_1 = 1, x_2 = 2, ..., x_n = n)$
3 $f_i(x_j) \rightarrow R_j$
4 Compute $F(x_i) = \sum_{j=1}^{n} f_j(x_i)$
5 Compute $(1,2, ..., \mathrm{n})(\lambda_1, \lambda_2, .., \lambda_\mathrm{n})^\mathrm{T} = (1,0, ..., 0)$
6 Compute $S = \sum_{i=1}^{n}\sum_{j=1}^{n} f_j(x_i) = \sum_{i=1}^{n} F(x_i)\lambda_i$
```

Multi party data sorting algorithm

Due to the multi-party summation of candidates' votes, to facilitate the authorization for viewing the ranking of candidates' votes, the multi-party sorting algorithm is employed [25–29]. The secure multi-party sorting algorithm, based on the Shamir threshold key system, is enhanced to sort the data. The specific steps of the algorithm are presented in Table 3 below:

In the algorithm, participant $R_i(1 \le i \le n)$ has secret data S_i and a verifiable public identity $x_j(1 \le j \le n)$. Randomly select $(\mathrm{t}-1)$-degree polynomials $f_{in}(x) = (S_i + n) + \sum_{j=1}^{t-1} a_{in}x^j$ and $g_i(x) = g_i + \sum_{j=1}^{t-1} b_{ij}x^j$ (with random numbers $g_i \in \mathbb{Q}^+$) to encrypt S_i. Substitute x_j into the random polynomials $f_{in}(x)$ and $g_i(x)$ to obtain n groups of calculation results, with $(\mathrm{n}+1)$ data points in each group. Calculate the values of $f_{in}(x)$ and $g_i\left(x_j\right)$, and send them to the corresponding participant R_j. This is analogous to the fifth step of the multi-party summation algorithm, where formula $(1,2, ..., \mathrm{n})(\lambda_1, \lambda_2, .., \lambda_\mathrm{n})^\mathrm{T} = (1,0, ..., 0)$ eliminates the value of random terms in polynomials $f_{in}(x)$ and $g_i(x)$. Participants calculate their vote data T_i and publish it for comparison and sorting.

Computational complexity

The above algorithm is analyzed. In step 2 of the summation algorithm, participants performed a total of n encryption operations on the secret data. In step 4, participants performed a total of n operations when summing data using expression $F(x_i)$. In step 5, a total of n^2 operations were performed to eliminate random item values. In step 6, a total of n summation operations are performed. In step 2 of the sorting algorithm, the polynomial $f_{ij}\left(x_j\right)$ and $g_i(x)$ are encrypted $n(n+1)$ times. In step 4, a total of n^2 calculations were performed for calculation $F_i(x_j)$. A total of n^2 calculations were performed when eliminating random items. In step 6, participants performed n^2 operations by calculating T_i. Compared with reference [30], the calculation time of our proposed scheme is reduced. With the increasing value of n, the change of calculation time is shown in Fig. 4.

Compared with the electronic voting protocols proposed in references [30, 31], this paper has reduced

Table 3 Multi-party data sorting algorithm

```
Algorithm: multi-party data sorting algorithm
Input: participant$R_i$' s secret data $S_i$ and corresponding public identity, where $x_j$
$(1 \le \mathrm{i}, \mathrm{j} \le \mathrm{n})$
Output:$T_i$
Step:
1 Compute
$f_{in}(x) = (S_i + n) + \sum_{j=1}^{t-1} a_{in}x^j, g_i(x) = g_i + \sum_{j=1}^{t-1} b_{ij}x^j (g_i \in \mathbb{Q}^+)$
2 Compute $f_{ij}\left(x_j\right), g_i(x)(x_1 = 1, x_2 = 2, ..., x_n = n)$
3 $f_i\left(x_j\right), g_i\left(x_j\right) \rightarrow R_j$
4 Compute $F_i(x_j) = \sum_{i=1}^{n} f_{ij}(x_j)g_i\left(x_j\right)$
5 Compute $(1,2, ..., \mathrm{n})(\lambda_1, \lambda_2, .., \lambda_\mathrm{n})^\mathrm{T} = (1,0, ..., 0)$
6 Compute $T_i = \sum_{j=1}^{n}\sum_{i=1}^{n} f_{in}(x)g_i(x)\lambda_j = \sum_{j=1}^{n} F_i(x_j)\lambda_j$
```

computational complexity and improved computational efficiency, although the round complexity is higher than that in reference [31]. The three schemes are discussed and analyzed under the semi-honest model, and the complexity comparison is presented in Table 4 below.

Safety proof and performance analysis

In this section, we conducted games between the adversary (attacker) and the challenger (participant) based on encryption algorithm schemes under the DBDH assumption and within the random oracle model, thereby proving that the scheme is secure against selected ciphertext attacks within the random oracle model. We also analyzed various performance aspects of the scheme, including privacy, legality, correctness, integrity, uniqueness, verifiability, non-repudiation, resistance to attacks, fairness, and other relevant criteria.

Safety proof

Lemma 1: Under the DBDH assumption and within the random oracle model, the candidate public key pk is employed to encrypt vote-related information in the scheme. The encryption algorithm of the scheme exhibits indistinguishability under adaptive chosen-ciphertext attacks, thus ensuring the scheme is IND-CCA2 secure.

Proof: Assuming the existence of a probabilistic polynomial-time algorithm PPT, given a public key encryption scheme $II = (Gen, En, De)$, the indistinguishability game between Adversary A and Challenger B under an adaptive chosen-ciphertext attack within the scope of PPT is as follows:

Table 4 Complexity comparison

Program	Computation model	Computational complexity	Wheel complexity
This paper	Semi honest model	$O(n^2)$	$O(n)$
[30]	Semi honest model	$O(n^2 + n + 1)$	–
[31]	Semi honest model	$O(n^3)$	$O(1)$

(1) Initialization: B generates encryption algorithm $II = (Gen, En, De)$, and A obtains public key pk.
(2) Inquiry 1: A asks the decryption oracle $OracleDe()$ owned by B, and B runs the decryption oracle $OracleDe()$ to decrypt the ciphertext C to obtain the corresponding plaintext, and send the plaintext to A
(3) Challenge phase: A selects two messages M_0, M_1 of the same length and sends them to B, B randomly selects a byte $\sigma \in \{0, 1\}$, and accepts B's ciphertext $C^* = E_n(pk, M_\sigma)$.
(4) Inquiry 2: A asks the decryption oracle $OracleDe()$ owned by B, A takes the ciphertext C^* $(C \neq C^*)$ and sends it to B, and B decrypts the ciphertext C^* and sends the plaintext to A.
(5) Guess: Adversary A submits σ's guess σ^*. When $\sigma^* = \sigma$, A wins the game, otherwise, the game fails.

The functional expression $Adv_{II,A}^{CCA2}(\lambda) = |Pr(\sigma = \sigma^*)$ $-1/2|$ of the advantage of the opponent A to win the game, where the safety parameter of $Adv_{II,A}^{CCA2}(\lambda) = |Pr(\sigma = \sigma^*) - 1/2|$ is λ,for the adversary A in any polynomial time, there is a function $\theta(\lambda)$ that can be ignored and contains the safety parameter λ, Let

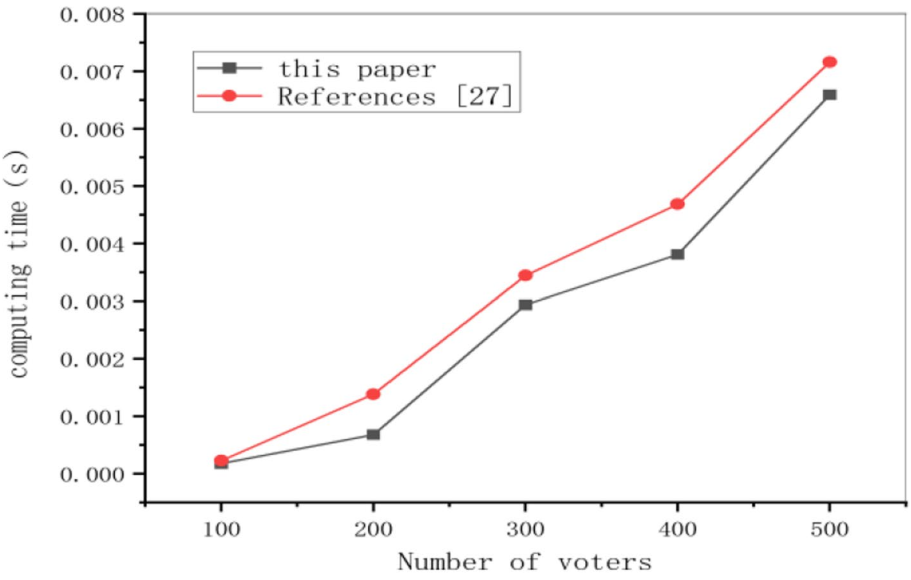

Fig. 4 Calculation overhead

$Adv_{II,A}^{CCA2}(\lambda) \leq \theta(\lambda)$,then the encryption algorithm is indistinguishable under adaptive ciphertext attacks.

Therefore, as per the security analysis outlined in Definition 2 and under the random oracle model, in the two stages of inquiry within the game between Adversary A and Challenger B, Adversary A's advantage in winning the game is only negligible, implying that no adversary can successfully break the encryption algorithm. This scenario corresponds to IND-CCA2 security. Conversely, if Adversary A only has a negligible advantage in the first query stage under the random oracle model, indicating that no adversary can penetrate the encryption algorithm, the scheme is said to have IND-CCA1 security.

Lemma 2: Based on the DBDH assumption and within the random oracle model, the voting information of voters in the scheme is encrypted and stored using the corresponding identity ID_i. The encryption algorithm of the scheme exhibits indistinguishability under adaptive chosen-ciphertext attacks, thereby ensuring the scheme is IND-ID-CCA secure.

Proof: Suppose there is a probabilistic polynomial time PPT, the same as Lemma 1. Given an encryption scheme $II = (Gen, En, De)$, the indistinguishability game between the adversary A and the challenger B under the adaptive choice ciphertext attack in polynomial time is as follows:

(1) Initialization: B enters the security parameter λ and generates the public system parameter ω and the secret master key SK_m. Denoted as $(\omega, SK_m) \leftarrow Init(\lambda)$.

(2) Inquiry 1: The execution of the inquiry depends on the result of the previous inquiry.

Perform $OracleSK()$ query on ID's secret key generation. B uses the secret key generation algorithm to generate a secret key SK corresponding to ID, and sends SK to the opponent.

Perform $OracleSK()$ query on (ID, C)'s decryption. B uses the secret key generation algorithm to generate the secret key SK corresponding to ID, then uses SK to decrypt the cipher text C, and the resulting plain text is sent to A.

(3) Challenge phase: A selects two messages of the same length M_0, M_1 and a challenged public key ID^* (restrict ID^* is not in the query phase 1). B randomly selects a byte $\sigma \in \{0, 1\}$, calculates the ciphertext $C^* = \varepsilon_{ID^*}(M)$ and sends it to A.

(4) Inquiry 2: A generates more inquiries, one of the following inquiries.

Perform $OracleSK()$ query on ID's secret key generation $(ID \neq ID^*)$. B responds like query 1.

Perform $OracleSK()$ query on (ID, C)'s decryption $((ID , C) \neq (ID^*, C^*))$. B responds like query 1.

(5) Guess: Adversary A submits σ's guess σ^*. When $\sigma^* = \sigma$,A wins the game, otherwise, the game fails.

The functional expression $Adv_{II,A}^{IND-ID\_CCA}(\lambda) = |Pr(\sigma = \sigma^*) - 1/2|$ of the advantage of the opponent A to win the game, where the safety parameter of $Adv_{II,A}^{IND-ID-CCA}(\lambda) = |Pr(\sigma = \sigma^*) - 1/2|$ is λ,for the adversary A in any polynomial time, there is a function $\theta(\lambda)$ that can be ignored and contains the safety parameter λ, Let $Adv_{II,A}^{IND-ID-CCA}(\lambda) \leq \theta(\lambda)$, then the encryption algorithm is indistinguishable under adaptive ciphertext attacks. Therefore, A's advantage in winning the game is negligible. No adversary can break the encryption algorithm. Therefore, our solution is IND-ID- CCA security.

Performance analysis

Privacy

The privacy of the scheme primarily encompasses two aspects: the identity privacy of voters and candidates, and the privacy of ballot data. The privacy of voters' and candidates' identities is ensured through ring signatures, which are verified to confirm the validity of the signatures. The voter's identity is used to encrypt and store voting information, and to securely demonstrate the indistinguishability of the encryption algorithm against adaptive chosen ciphertext attacks. Ballot data privacy employs a threshold key-sharing system to conduct secure multi-party summation and sorting calculations on votes, with these calculations being performed in an encrypted state. The published ballot results are encrypted and linked with the candidate's public key. The candidate can use the private key to verify the identity, decrypt, and view the corresponding ballot information. The public key encryption system can withstand adaptive chosen ciphertext attacks, as demonstrated by security proofs. The privacy of voters' and candidates' identities and the privacy of vote data are guaranteed.

Legality

Once voters and candidates have been registered and their information is verified as legitimate (with the format and length of the registration details conforming to the requirements), unique identity ID_i can be issued, enabling them to vote. If not, the voting process cannot proceed.

Correctness

Correctness is reflected across the various stages of the voting plan. In the registration stage, it is possible to ensure that the identities of voters and candidates are legitimate before voting. During the voting phase, each voter can only vote once, and any repeated voting results in invalid ballots. In the vote counting stage, the vote information is processed through multi-party summation and multi-party sorting agreements, without the intervention of factors considered by the vote-counter, ensuring accurate vote counting. In the announcement stage, the candidate decrypts the ballot corresponding to identity ID_i on the blockchain and checks the status of their ballot.

Completeness

Voters and candidates must register within the designated registration period, and the contract verifies whether the number of candidates and voters fulfills the required specifications. Furthermore, using the ring signature algorithm, the voter confirms and signs the ballot and identity, ensuring that no one can forge the signature.

Uniqueness

Voters are allotted only one opportunity to vote. Upon registration, each voter receives a unique identity ID_i associated with their ballot. Once a vote is confirmed, it cannot be recast; any attempt to do so renders the vote invalid. Repeated voting must be avoided as it can compromise the accuracy of the counting results.

Verifiability

After the voter completes the voting operation, the ballot information is stored in the IPFS distributed file system, and the index information is stored on the blockchain. Because the blockchain information is immutable, anyone with access authority can compare the blockchain index information with the vote information stored in IPFS. At the conclusion of the vote counting stage, the key counting steps are recorded on the blockchain for traceability and verification. The candidate encrypts the final vote results and stores them on the blockchain. The candidate or other authorized individuals can also verify the number of votes received by each candidate.

No receipt

The ballots are encrypted and processed using a threshold key-sharing system, and they can only be decrypted after the threshold is met, preventing attackers from modifying the ballot results. Concurrently, candidates cannot provide receipts as proof of their votes, and voters cannot create receipts that contradict their actual votes.

Resistance to attack

Thanks to the synchronization mechanism, candidates and voters receive a unique identity ID_i during the specified registration period. They submit ballot information to the contract, encrypt the ballot-related data, and store it in the IPFS distributed file system, while storing the index information on the blockchain. To facilitate the verification of identities and ballot information, voters and candidates are aware only of their own identities throughout the voting process, preventing them from knowing specific identities and thus preventing candidates from colluding with voters to canvass votes, which would undermine the integrity of the entire voting process. To prevent voters from colluding to alter the voting results, we employ signatures to anonymize identities and use threshold keys to encrypt votes.

Fairness

The vote counting in the scheme is performed by the MPC nodes within the computing network. It is divided into two stages: secure multi-party summation and secure multi-party sorting. The results of each stage are recorded on the blockchain, facilitating traceability, viewing, and verification. The secret sharing system prevents voters from colluding to disclose votes, ensuring fairness.

Scheme comparison

Through performance analysis, the comparison between this paper and related work is presented in Table 5 below. Reference [10] proposed the first scheme to achieve self-counting by using a smart contract with a zero-knowledge proof encryption mechanism. This scheme can defend against replay attacks, but it has the problem of an upper limit on calculation and storage capacity. To address the issue of online voting systems having a single point of failure and being unable to guarantee vote privacy, reference [13] introduces a distributed, privacy-protected, end-to-end verifiable electronic voting system. Reference [30] designed a short linkable ring signature electronic voting scheme based on blockchain, allowing large-scale participation, supporting multiple voting options, and featuring a self-counting function. Reference [31] proposed a voting scheme with complete privacy protection, based on the general technology of distributed ElGamal encryption and hybrid matching, ensuring the privacy of voters and candidates. Reference [32] proposed an electronic voting scheme based on blockchain, utilizing homomorphic encryption and ring signature technology. This scheme is suitable for large-scale voting and addresses issues of fraud, vote verification, and low counting efficiency across all aspects of voting. Reference [33] proposes an electronic voting protocol based on

Table 5 Comparison of electronic voting schemes

Program	Encryption mechanism	Counting method	Full privacy	Non-receipt	Verifiability	Resistance to attack
[10]	Zero-knowledge proof	Self-calculation	section	—	√	√
[13]	Zero-knowledge proof, Symmetric encryption	Third-party computing	√	—	√	√
[30]	Ring signature	Self-calculation	√	—	√	√
[31]	Zero-knowledge proof, Homomorphic encryption	Third-party computing	√	×	√	√
[32]	Homomorphic encryption, Ring signature	Third-party computing	section	√	√	—
[33]	Homomorphic signcryption	Self-calculation	section	×	√	×
This paper	Asymmetric encryption, Ring signature	Self-calculation	√	√	√	√

homomorphic signcryption and blockchain, which uses the aggregation feature to count the homomorphically encrypted votes, thereby improving voting efficiency.

Conclusion

This paper introduces a secure, multi-party verifiable electronic voting scheme based on blockchain technology. The scheme employs ring signature technology to ensure the anonymity of voters' and candidates' identities, with the management contract verifying the legitimacy of voting identities. The calculation contract primarily executes assigned computational tasks, including multi-party summation and sorting protocols, thereby enhancing counting efficiency. Key calculation results are returned and stored on the blockchain for easy retrieval and verification. The scheme utilizes IPFS to address the low operational efficiency stemming from limited block storage. The candidate's public key encrypts the corresponding ballot information and identity, published on the blockchain, allowing the candidate or other authorized individuals to decrypt, verify, and view it. Finally, the security proofs and performance analysis demonstrate that the scheme is secure and reliable. This paper does not consider the security of the contract itself; future work will address the security of the contract.

Authors' contributions

W. X. S writes the entire paper. F. T provides recommendations in the selected direction and also offers fund support. L. C.Y adjusts the format of the paper. F. Optimize the charts in J.L's paper. All authors reviewed the manuscript.

Funding

This work is supported by the National Natural Science Foundation of China (Grant No.62162039,62462044), Foundation for the Key Research and Development Program of Gansu Province, China (Grant No.23YFGA0060,24YFFA016).

Data availability

No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate

No study was carried out that required ethical approval.

Consent for publication

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 2 August 2024 Accepted: 4 December 2024
Published online: 20 December 2024

References

1. Chaum DL (1981) Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM 24(2):84–90
2. Boneh D, Golle P (2002) Almost entirely correct mixing with applications to voting. In Proceedings of the 9th ACM conference on Computer and communications security 68–77
3. Fujioka A, Okamoto T, Ohta K (1992) A practical secret voting scheme for large scale elections. Proc AUSCRYPT'92, Gold Coast, Queensland, Australia 718:244–251
4. Luo F, Lin C, Zhang S, Liu Y (2015) Receipt-free electronic voting Scheme based on FOO voting Protocol. Comput Sci 42(08):180–184
5. Yi X, Okamoto E (2013) Practical Internet voting system. J Comput Appl 36(1):378–387
6. Chandra Priya J, Sathia Bhama PR, Swarnalaxmi S, Aisathul Safa A, Elakkiya I (2020) Blockchain centered homomorphic encryption: A secure solution for E-balloting. In Proceeding of the International Conference on Computer Networks, Big Data and IoT (ICCBI-2018). Springer International Publishing 811–819
7. Zhao Z, Chan THH (2016) How to vote privately using bitcoin. In Information and Communications Security: 17th International Conference, ICICS 2015, Beijing, China, December 9–11, 2015, Revised Selected Papers. Springer International Publishing 17:82–96
8. Lee K, James JI, Ejeta TG et al (2016) Electronic voting service using blockchain. J Digi Forensics, Security and Law 11(2):123–136
9. Jason PC, Yuichi K (2017) E-voting System Based on the Bitcoin Protocol and Blind Signatures. Transactions on Mathematical Modeling and Its Applications 10(01):14–22
10. McCorry P, Shahandashti SF, Hao F (2017) A smart contract for boardroom voting with maximum voter privacy. In Financial Cryptography and Data Security: 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers. Springer International Publishing 21:357–375
11. Bartolucci S, Bernat P, Joseph D (2018) SHARVOT: secret SHARe-based VOTing on the blockchain. In Proceedings of the 1st international workshop on emerging trends in software engineering for blockchain 30–34
12. Hjálmarsson FÞ, Hreiðarsson GK, Hamdaqa M, Hjálmtýsson G (2018) Blockchain-based e-voting system. In 2018 IEEE 11th international conference on cloud computing (CLOUD). IEEE 983–986
13. Chondros N, Zhang B, Zacharias T et al (2019) Distributed, end-to-end verifiable, and privacy preserving Internet voting systems. Computers&security 83:268–299

14. Jayakumari B, Sheeba SL, Eapen M et al (2024) E-voting system using cloud-based hybrid blockchain technology. Journal of Safety Science and Resilience 5(1):102–109
15. Meter C (2017) Design of distributed voting systems. arxiv preprint arxiv:1702.02566
16. Xie W, Li W, Zhang H (2023) Electronic Voting Privacy Protection Scheme Based on Double Signature in Consortium Blockchain. In International Conference on Artificial Intelligence Security and Privacy. Singapore: Springer Nature Singapore 548–562
17. Taş R, Tanrıöver ÖÖ (2020) A systematic review of challenges and opportunities of blockchain for E-voting. Symmetry 12(8):1328
18. Baudier P, Kondrateva G, Ammi C et al (2021) Peace engineering: The contribution of blockchain systems to the e-voting process. Technol Forecast Soc Chang 162:120397
19. Haoyang An, Debiao He (2023) Baozi Jian, etc Ring Signature Based on SM9 Digital Signature and Its Application in Blockchain Privacy Protection. Computer Research and Development 60(11):2545–2554
20. Huang K, Xiong L, Sheng Y et al (2024) Public key cryptography in blockchain: design, analysis, security evaluation, and prospects. J Comput Sci 47(03):491–524
21. Chen Yu, Hongxu Yi (2024) Wang Yuyu Overview of Public Key Encryption. Journal of Cryptography (in Chinese and English) 11(01):191–226. https://doi.org/10.13868/j.cnki.jcr.000676
22. Pu H, Cui Z, Liu T, Wu Z, Du H (2021) An Electronic Voting Scheme Based on LUC Secret System and Secret Sharing. International Journal of Network Security 23(1):97–105
23. Wang Z, Cheung SCS, Luo Y (2016) Information-theoretic secure multiparty computation with collusion deterrence. IEEE Trans Inf Forensics Secur 12(4):980–995
24. Ji ZX, Zhang HG, Wang HZ et al (2019) Quantum protocols for secure multi-party summation. Quantum Inf Process 18(6):1–19
25. Tang C, Shi G, Yao Z (2011) Secure multi-party computation protocol for sequencing problem. Science China Information Sciences 54:1654–1662
26. Shundong Li, Jia K, Xiaoyi Y et al (2018) Secure multi-party computation for multi-character sorting [J]. J Comput Sci 41(5):206–222
27. Dery L, Tassa T, Yanai A (2021) Fear not, vote truthfully: Secure Multiparty Computation of score based rules. Expert Systems with Applications 168:114434
28. Sutradhar K, Om H (2021) An efficient simulation for quantum secure multiparty computation. Sci Rep 11(1):2206
29. Malkawi M, Yassein MB, Bataineh A (2021) Blockchain based voting system for Jordan parliament elections[J]. International Journal of Electrical and Computer Engineering 11(5):4325
30. Wu Q, Yang F, Zhou F et al (2024) A secure electronic voting scheme based on blockchain and short linkable ring signatures. Journal of Northeastern University (Natural Science Edition) 45(05):619–627
31. Lei P, Maohua S, Shoushan L, Bai W, Yang X (2012) Full privacy preserving electronic voting scheme. J Chian Univ Posts Telecommun 19(4):86–93
32. Wang B, Sun J, He Y et al (2018) Large-scale Election Based On Blockchain. Procedia Computer Science 129:234–237
33. Qu W, Wu L, Wang W et al (2022) A electronic voting protocol based on blockchain and homomorphic signcryption. Concurrency and Computation: Practice and Experience 34(16):e5817

Publisher's Note