International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Privacy-Preserving and Verifiable Electronic Voting via Blockchain

Mr. N. S C Mohana Rao[1], Mr. Pothula Nani Babu[2], I Srinivasu[3]

[1]Associate Professor, M Tech, VSM College of Engineering(A), Ramachandrapuram.
[2,3]Assistant Professor, M Tech (PhD), VSM College of Engineering(A), Ramachandrapuram.

Abstract:
While the ballot's confidentiality is guaranteed by the existing electronic voting method, voter anonymity is not specifically addressed. Since smart contracts control every aspect of the voting process, large-scale voting is difficult to accomplish and contract execution efficiency is reduced. Comprehensive explanations of the voting procedure and the validity of the material on the ballot are lacking. This article presents a blockchain-based multiparty safe verifiable electronic voting system.
The IPFS distributed file system is used in the solution to address the issue of block storage limitations. To increase efficiency, management and computation contracts are used to handle the task distribution and vote counting procedures independently. Voter anonymity is maintained using ring signature technology. The computation results are encrypted and posted to the blockchain, allowing for verification, and the vote counting procedure employs a key-sharing mechanism to protect the privacy of the votes. Lastly, a performance analysis and security proof of the scheme were carried out.

Keywords: Ring signature, IPFS, blockchain, electronic voting, and verifiable.

Introduction:

As information technology continues to advance, electronic voting has emerged as a crucial element of e-government. The efficiency of electronic voting has steadily increased due to the broad use of Internet and cryptographic technology, which has reduced numerous expenses and made the procedure more convenient. Building on this framework, a large number of academics have put forth several electronic voting schemes that use methods like blind signatures, ring signatures, hybrid networks, and homomorphic encryption.
By using a hybrid network protocol to link voters and ballots, Chaum et al. were able to accomplish anonymous voting [1], however this method is vulnerable to ciphertext assaults. By applying probability theory to statistically confirm the likelihood of successfully searching for cheating servers, Boneh and Golle et al. [2] enhanced the voting protocol based on hybrid network technology to stop ciphertext leakage; however, this added to the protocol's computational complexity. The well-known FOO voting protocol was proposed by Fujioka, Okamoto, and others [3]. This method, which is based on bit commitment and blind signature technology, is effective for large-scale voting but ineffective and does not permit vote abandonment. The FOO voting protocol was further enhanced by Roffen et al.'s proposal of an electronic voting system without receipts [4], which allowed for waivers and guaranteed anonymity, verifiability, and non-receipt of votes. Citation [5] created a homomorphic encryption-based

application-oriented network electronic voting system. An electronic voting system based on fully homomorphic encryption was proposed by Chandra Priya, J. et al.
[6]. The use of blockchain technology in electronic voting has made voting easier in recent years and reduced the need for third-party organizations. has been diminished. A Bitcoin-based electronic voting protocol is put forth for the first time [7]. This plan uses an incentive system to ensure voting fairness. Even if they make use of important blockchain technology, several blockchain-based electronic voting methods [8, 9] still need outside help to cast ballots. Only a limited number of votes are permitted by the voting procedure suggested in reference [10], and abstention is not permitted. References [11,12] achieve a privacy-preserving, self-counting electronic voting system by using smart contracts in place of third-party institutions. To address the requirements of large-scale voting, Chondros et al. [13] designed a distributed electronic voting mechanism. A practical Byzantine fault-tolerant (PBFT) consensus process is employed to guarantee that votes are neither altered or tainted in order to produce correct voting results. There are various problems with the voting systems covered in the literature mentioned above: First of all, some schemes employ third-party counting techniques, which raise security concerns in addition to decreasing vote efficiency. Second, although some methods use self-counting votes, they rely on smart contracts for coordinated processing and keep all voting data on the blockchain. Without a doubt, this strategy raises the price of contracts and block storage, which restricts the scope of voting. Finally, the anonymity of voters' identities is not given much thought by most systems, which mainly concentrate on guaranteeing the validity and security of voting data. Given the possible issues raised above, this paper develops a voting model, creates a voting algorithm, and suggests a safe, multi-party verifiable electronic voting system based on blockchain technology. Third-party institutions are replaced with smart contracts, which include management and computing contracts. To support a specific scale of electronic voting, the management contract distributes the vote counting task among the computing network. During the registration phase, the management contract confirms the voters' identity and authenticity. To make it easier to find and verify votes, the computing contract computes, sorts, and saves the vote results for every candidate on the blockchain. This plan works well for elections where there are several candidates. Lastly, security proofs and performance studies are carried out using the random oracle model and the DBDH assumption, and the algorithmic complexity of this method is examined and contrasted.

Related work

Ring signature:

One particular kind of group signature is the ring signature [19]. There is no mechanism for establishing a group and no trusted hub. The signer is totally anonymous to the verifier. Its distinctive ring structure is reflected in the name. The genuine signer creates a ring with a break using the public keys produced by other potential signers, then joins the break with their own private key to establish a whole ring. Any verifier can confirm who created the ring signature by using the public keys of the ring members. They confirm the validity of the signature based on the computation's outcome, after which they either accept or reject it. The following three algorithms are typically seen in ring signatures:

1. Key generation algorithm R − Gen: The signer in the ring creates a set of public keys PK = (pk1, pk2,..., pkn) and a public–private key pair (pk, sk) using the key generation algorithm.

2. Algorithm for Ring Signatures The algorithm used to create ring signatures is called R − Sig. The inputs consist of the private key sk of the real signer in the ring, the public key set PK of the n signers in the ring, and the message m to be signed. Message m is subsequently subjected to the algorithm's ↞ sig(sk,m, PK) ring signature computation.
3. Ring signature verification procedure R − Ver: Enter the signed message m and the outcome of the message's ring signature computation. The output is the verification result after the signature has been checked and calculated using verify(m,PK). One verification was successful, and zero was unsuccessful.

Premises for challenging issues and public key encryption:

First definition:

With groups G1 and G2 assumed, and a mapping e: G1 × G2 → G2, challenger B chooses a, b, c, and z ∈ Zq at random to produce two quintuples, specifically: Choose a byte { at random from {0, 1}; output the quintuple T1 if { = 0; output T2 otherwise.

T1 = (p,A = pa, B = pb, C = pc, Z = e(p, p)z)

T2 = (p,A = pa, B = pb, C = pc, Z = e(p, p)abc)

Adversary A uses the obtained quintuples T1 and T2 to generate {∗ as a hypothesis about }. Adversary A's advantage is AdvA = | Pr[{∗ = }]−1 2| ≥ } (an undeniable advantage }), where the probability is dependent on using random bits and selecting a, b, c, and z ∈ Zq at random. Second definition: The probabilistic polynomial-time method PPT serves as the foundation for a public key encryption scheme [20], which is made up of the following four parts:

Algorithm for system generation Setup(1): This algorithm outputs the system parameter and accepts the safety parameter as input.

Key Generation method Gen(w): This method generates a public–private key pair (pk, sk) from the system-generated parameter (pk,sk)as input.

Cryptography Algorithm En(pk,M,r): This algorithm generates the ciphertext C from the inputs of the public key pk, the plaintext message M, and the system parameter `.

Decryption Method De(sk, C): When given the ciphertext C and the private key sk, this algorithm produces the plaintext M; if not, it fails.

The previously mentioned public key encryption system, which is based on number-theoretic assumptions, was shown to be accurate by Reference [21], which also conducted a security study of the scheme using the framework of adversary attack games in a random oracle model. The scheme's advantage function was established, and it was demonstrated that adversaries using the random oracle model have very little advantage over the course of two interrogation stages, meeting the requirements of indistinguishability under chosen-ciphertext attacks (IND-CCA) security, also known as INDCCA2 security.

Requirements for electronic voting security;

The electronic voting system replaces techniques like dictatorship and drawing lots, provides fairness and transparency, saves a lot of resources, and has emerged as the most important decision-making tool in e-government. It converts individual ideals into community ones and embodies democracy. A computer network is used for electronic voting. Since the network environment is inherently risky,

security is an essential component of electronic voting. The following essential conditions should be met by a secure electronic voting protocol:

1. Privacy: To safeguard voter privacy, including voter and candidate identities and ballot data, by making sure that only the voter has access to this information.
2. Legality: Voters are only permitted to cast ballots if their identities have been verified via authentication and registration.
3. Correctness: To guarantee precise counting, ballots must follow a standard format.
4. Completeness: Before moving on to the vote counting phase, all legitimate votes cast by voters must satisfy the accuracy requirements.
5. Uniqueness: Every voter is only allowed to cast one ballot; they cannot do it more than once.
6. Verifiability: In compliance with regulations, voting results and voter IDs must be verifiable to stop unauthorized individuals from manipulating or fabricating the results.
7. No Receipts: Candidates should not be given evidence of their votes, and voters should not be able to produce receipts that differ from their real votes in order to avoid changes to the voting results and guarantee the validity of the vote.
8. Anti-Attack: This involves stopping candidates from planning to canvass with voters, stopping voter cooperation and attacks to examine ballot information, and destroying ballot information to make voting impossible. The attacker is regarded as a malevolent actor in various attack scenarios.
9. Fairness: None of the above stated criteria may influence the vote result.

System model:

Conventional model of voting:

Traditional voting systems have several drawbacks when the voting scale grows, including the high expense and low efficiency of manual vote counting and the inability to guarantee result correctness. The integrity of the voter (or a reliable third party) is essential to both the confidentiality of traditional voting and the precision of vote counting. Because traditional voting systems give voters a great deal of power, voter identity and ballot information may be compromised, which could influence their willingness to cast a ballot and give candidates canvassing opportunities, potentially compromising the election's fairness. Fig. 1 below depicts the conventional election model:

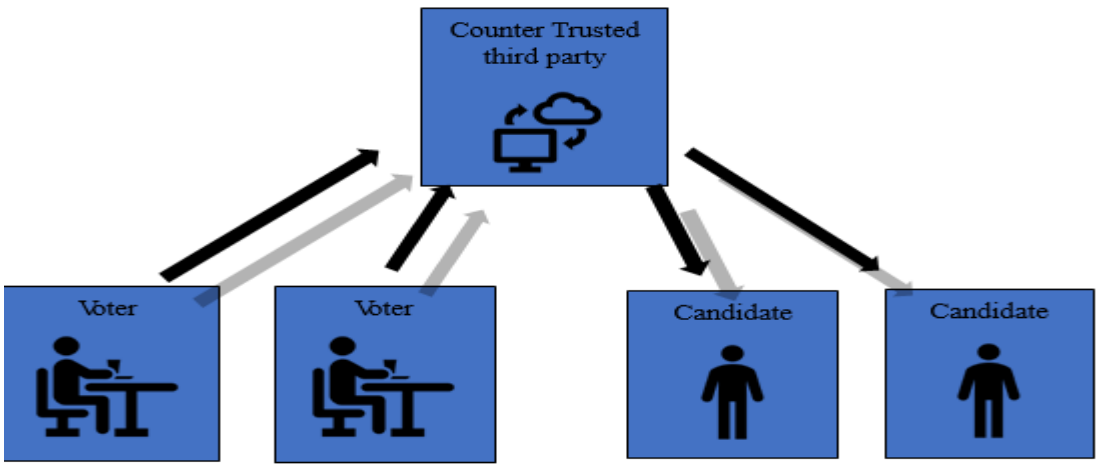

Fig. Model of traditional electronic voting scheme

The BC-MPC voting model:

An electronic voting system based on the LUC secret system and secret sharing is suggested to stop voting fraud and guarantee the impartiality and fairness of electronic voting [22]. This system improves voting efficiency and satisfies the security requirements of anonymity, no reception, verifiability, and fairness. Blockchain technology can solve present problems with anonymity, transparency, and public verification in electronic voting because of its decentralization, openness, tamper resistance, anonymity, and traceability features. The blockchain, candidates, MPC contracts (including management and computing contracts), computing networks, the blockchain, and the IPFS distributed file system are the six main entities that make up our suggested blockchain-based secure multi-party voting calculation model, which is presented in this section. The management contract and the counting contract are the two sections that make up the MPC contract. Four steps are usually included in an electronic voting scheme: registration, voting, counting, and verification. The program's precise structure is shown in Fig. 2 below.

In order to make sure that every registered voter and candidate is authentic, the management contract first confirms the validity of their identities using the format and length of the input characters. The contract creates a ring signature, computes the signature, and returns a distinct identity after voter registration using the voter's registered account number and password as a public–private key pair. Voters encrypt and store vote data in the IPFS distributed file system using their identities. The index data acquired through the distributed hash table is the storage location that is returned. The blockchain is used by voters to store the index data, and a consensus process is used to get nodes to agree. The decryption key is acquired by the authorizer to Examine the voting data and confirm its legitimacy and legality. Second, the synchronization mechanism ensures that, after the registration period concludes, the management contract executes task allocation, generates the calculation script based on the data summation and sorting algorithm, processes the voting information based on the MPC parameters (including participants' public identities, secret data, etc.), and transmits calculation requests to the MPC nodes within the computing network. After processing the votes in a sequential manner using multi-party computation for sorting and summarization, each computing network node sends the computation results back to the management contract. Using a public key, the management contract encrypts the candidate's identity and the associated ballot data before storing them on the blockchain.

To ensure consistency, the consensus method updates and validates the block information, making search and verification simple. Lastly, the candidate or other authorized personnel can view the vote results by using the private key to decrypt and confirm their identity.

Design of an electronic voting protocol:

Definition of protocol parameters:

Table 1 lists the voting algorithm's role descriptions and associated symbols. The following is a description of the roles and their duties:

vi (Vote): Candidates are linked to the voting options, and voters are linked to the voting data. Vi (Voters): Following identification verification, voters register to cast their ballots. The symbol of the ring

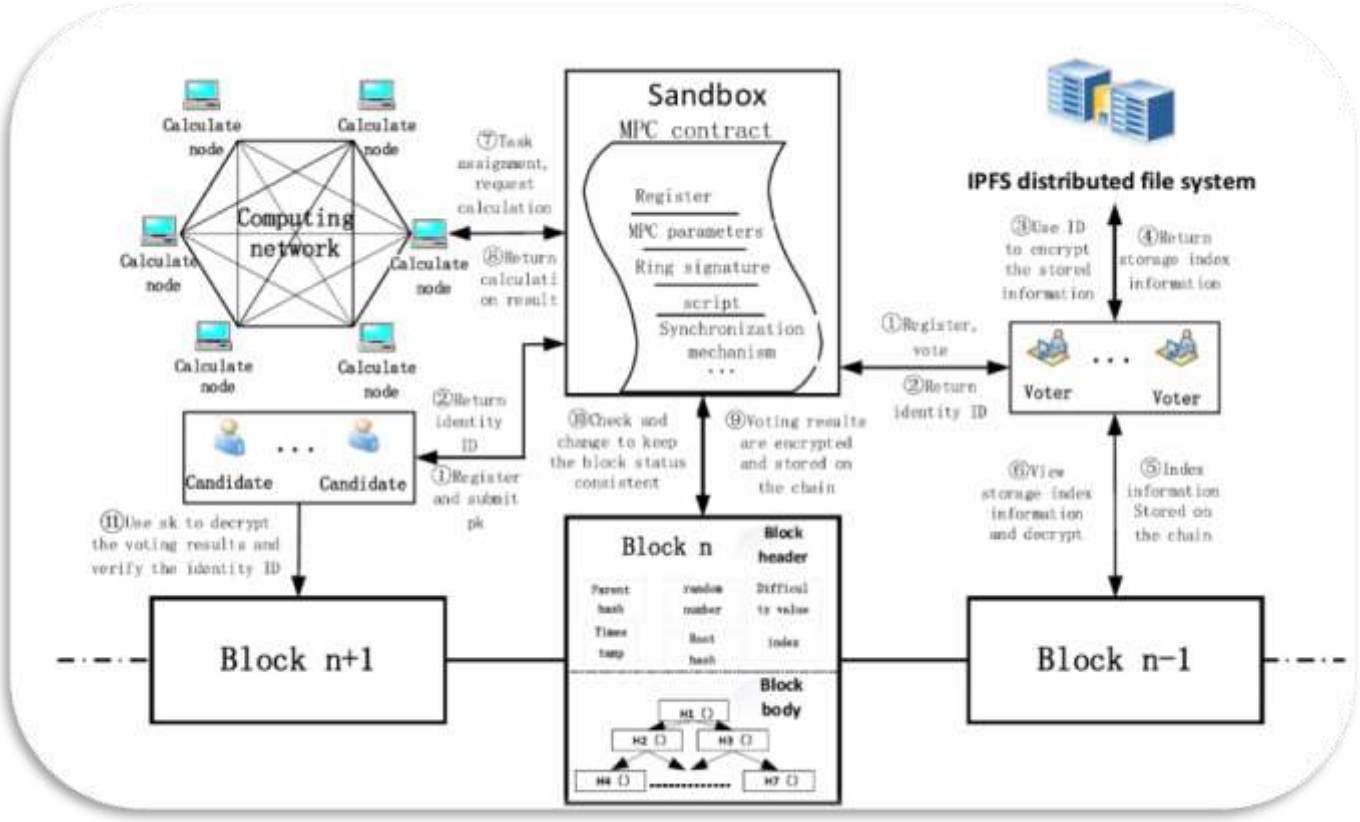

Fig. 2 BC-MPC program model

Table 1 Symbols and definitions of voting roles:

Symbol	Role definition
vi	vote
Vi	Voter
Ci	Candidate
SCm	Management contract
SCt	Counting Contract
Peer	blockchain node
MPCn	MPC node
IDi	Unique identification

Without requiring human involvement, ture anonymizes the identity and communicates with the computer protocol via the contract. Ci (Candidate): The candidate can use the appropriate public key pk that the management contract uses to encrypt the voting result on the chain after the voter's identity is confirmed by IDi and the public key pk is stored in the contract. To see one's own votes, the private key SK must be decrypted.

SCm (contract for management): Verify the legality of the voter's identification information registration, sign the voting documents, and compute the contract management. To encrypt the ballot outcome and post it on the chain, use the candidate public key PK. Vote counting contract (SCt): Acknowledge the voter's ballot details and confirm the ballot's authenticity. Multi-party sorting and multi-party summation computation activities are applied to the votes.

Peer (Blockchain node): To supply matching services, contracts—both administration and calculation contracts—are installed in nodes. Simultaneously, the node's vote results for the candidates and the index data are encrypted and saved.

MPCn (MPC node): The MPC node is given the summation and sorting computation duties by the ticket counting contract, and it computes and sends the result back to the contract for processing. IDi (Unique ID): Each user (voter and candidate) has a legitimate and unique ID, and their identity information is listed in the agreement.

Steps for voting Registration, voting, counting, and publishing are the four key phases of the electronic voting system. Figure 3 displays the scheme's sequence diagram. Here is the precise procedure:

1) Registration stage:

Both candidate Ci and voter Vi must register within the allotted time. The management contract SCm's ring signature algorithm signs the identity data, creates a distinct identity IDi associated with the identity data, and confirms that the signature is legitimate. The consistency of the identity's format is checked. Identification IDi is returned to the voter or candidate if it is lawful and they are eligible to cast a ballot.

2) Voting stage:

Voter Vi casts a ballot after confirming that they are eligible to do so (after receiving their unique IDi). Voting with a unique identifier IDi is limited to one vote and cannot be repeated. Use the ring signature algorithm $\sim \rightarrow \text{sig}(sk, m, PK)$ to calculate the signature; PK is a collection of public keys, and E is the information to be signed, such as voter Vi's and candidate Ci's identity information. Verification of signatures $(\sim, m, PK)$ guarantees identity anonymity.

3) Counting stage:

The process is transferred to the calculation contract after the management contract verifies the identities and votes of voter Vi and candidate Ci. The calculation contract assigns calculation tasks to the MPCn node, which, upon receiving a calculation request, carries out the pertinent tasks, which are mainly split into two parts: multi-party sorting and multi-party summation. First, the votes are sorted according to the total number of votes each candidate has received, and then they are summed based on candidate identity.

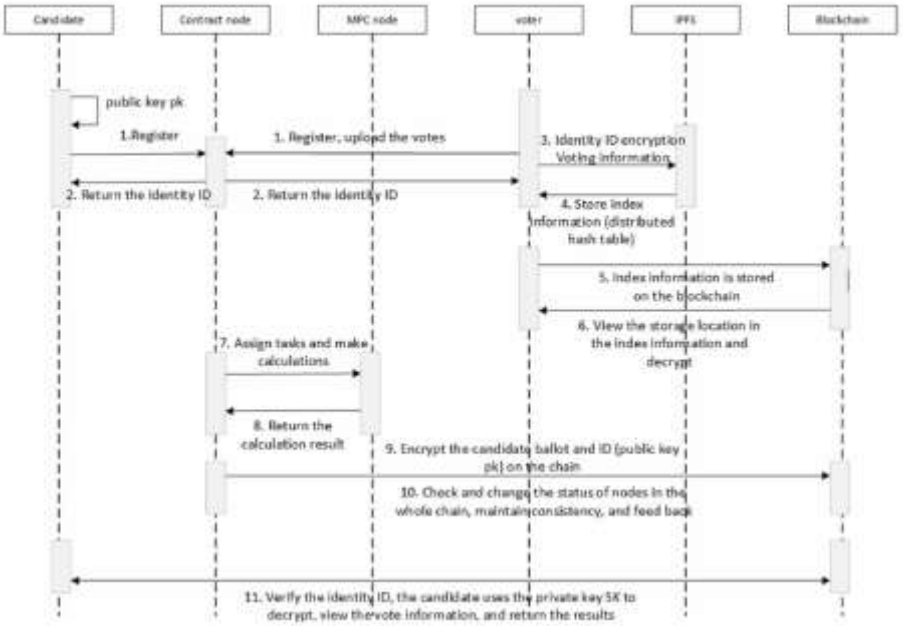

Fig. 3 Timing diagram of the scheme

4) Verification stage:

Through the management contract, the candidate's vote results and associated identity IDi are encrypted using the matching public key and posted on the blockchain node Peer. To examine their own vote, the candidate authenticates their identity and decrypts the data using their private key.

Algorithm for voting A method for multi-party data summarization:

In order to prevent voter collusion and guarantee the fairness of the voting process, the method uses the threshold key sharing system's secure multi-party summation protocol in conjunction with the secure multi-party summation algorithm to count votes for each candidate. Regarding the particular algorithmic stages, In the algorithm, participant Ri($1 \leq i \leq n$) chooses at random a polynomial fi(x) = Si +~t−1 j=1 aijxj of degree (t − 1) to encrypt Si. Ri also has a verifiable public identity xj($1 \leq j \leq n$) and secret data Si. In order to calculate the random polynomial fi(x), where (x1 = 1, x2 = 2,..., xn = n), substitute xj. Then, communicate the computation result to the appropriate participant, Rj. Utilizing the formula F(xi) = j=1fj(xi), add up the data. The value of the n-dimensional Vander monde matrix Q = (λ_1, λ_2,.., λ_n)^T is similar to (x1 = 1, x2 = 2,..., xn = n); there is an invertible matrix such that equation (1,2,..., n)(λ_1, λ_2,.., λ_n)^T = (1,0,..., 0) is satisfied, thereby removing the random value in fi(x).

Data sorting method for multiple parties Because of the multi-party total of votes cast for candidates, in order to make it easier to obtain permission to display the ranking of The algorithm used for multi-party sorting of candidates' votes is

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 $(1 \le i, j \le n)$
Output: S
Step:
1 Compute $f_i(x) = S_i + \sum_{j=1}^{t-1} a_{ij} x^j$
2 Compute $f_i(x_j)(x_1 = 1, x_2 = 2, \ldots, x_n = n)$
3 $f_i(x_j) \rightarrow R_j$
4 Compute $F(x_i) = \sum_{j=1}^{n} f_j(x_i)$
5 Compute $(1,2,\ldots,n)(\lambda_1, \lambda_2, .., \lambda_n)^T = (1,0,\ldots,0)$
6 Compute $S = \sum_{i=1}^{n} \sum_{j=1}^{n} f_j(x_i) = \sum_{i=1}^{n} F(x_i)\lambda_i$

employed [25–29]. To sort the data, the Shamir threshold key system-based secure multi-party sorting method is improved. The particular actions that the algorithm takes are shown below in Table 3:
Participants Ri(1 ≤ i ≤ n) in the method have a verifiable public identity xj(1 ≤ j ≤ n) and secret data Si. To encrypt Si, choose at random the (t − 1)-degree polynomials fin(x) = (Si + n) +{t−1 j=1 ainxj and gi(x) = gi +{t−1 j=1 bijxj (using random values gi ∈ Q+). To get n sets of calculation results, with (n + 1) data points in each group, substitute xj into the random polynomials fin(x) and gi(x).

The complexity of computation:

The algorithm mentioned above is examined. Participants encrypted the secret data n times in total during step 2 of the summing process. Step 4: Participants completed n operations in total when adding up
data using the formula F(xi). In the fifth stage, n2 Procedures were carried out to get rid of odd items. principles. Step 6 involves n summation operations in total. arrived out. The sorting algorithm's second phase uses the polynomial The encryption of fij{xj{ and gi(x) occurs n(n + 1) times. In phase 4, a total of n2 computations were carried out. Fi(xj). There were n2 computations done in total. when removing arbitrary objects. Step 6: Participants calculated Ti and carried out n2 procedures. Compared

Table 3 multi-party data sorting algorithm:

Input: participant R_i's secret data S_i and corresponding public identity, where x_j $(1 \le i, j \le n)$
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 $f_{ij}(x_j), g_i(x)(x_1 = 1, x_2 = 2, \ldots, x_n = n)$
3 $f_i(x_j), g_i(x_j) \rightarrow R_j$
4 Compute $F_i(x_j) = \sum_{i=1}^{n} f_{ij}(x_j) g_i(x_j)$
5 Compute $(1,2,\ldots,n)(\lambda_1, \lambda_2, .., \lambda_n)^T = (1,0,\ldots,0)$
6 Compute $T_i = \sum_{j=1}^{n} \sum_{i=1}^{n} f_{in}(x) g_i(x) \lambda_j = \sum_{j=1}^{n} F_i(x_j) \lambda_j$

Performance analysis and safety proof:

This section demonstrated that the scheme is safe from specific ciphertext attacks within the random oracle model by conducting games between the adversary (attacker) and the challenger (participant) based on encryption algorithm schemes under the DBDH assumption.

Proof of safety:

Lemma 1: The candidate public key pk is used to encrypt vote-related data in the scheme under the DBDH assumption and in the random oracle model. The system is guaranteed to be IND-CCA2 secure since its encryption algorithm shows indistinguishability under adaptive chosen-ciphertext attacks.

Proof: Given a public key encryption scheme II = (Gen, En, De) and the assumption that a probabilistic polynomial-time algorithm PPT exists, the indistinguishability game between Adversary A and Challenger B under an adaptive chosen-ciphertext attack within the parameters 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: A acquires the public key pk, and B creates the encryption algorithm II = (Gen, En, De).
2. Inquiry 1: A requests that B use the decryption oracle OracleDe(), and B uses OracleDe() to decrypt the ciphertext C and produce the associated plaintext, which is then sent to A.
3. Challenge phase: A chooses two identically lengthed messages, M0 and M1, and transmits them to B. B chooses a byte at random from {0, 1} and accepts B's ciphertext, C∗ = En(pk,M}).
4. Inquiry 2: A requests that B use the decryption oracle OracleDe(). A then delivers the ciphertext C∗ (C = C∗) to B, who decrypts it and provides A the plaintext.
5. Adversary, I suppose A offers `'s guess { σ∗. A wins the game when ` σ∗ = σA }; if not, the game is lost.

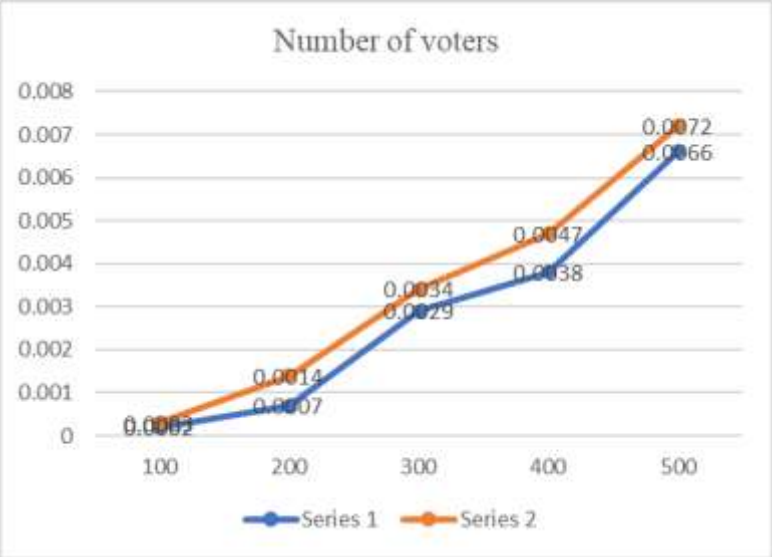

Fig. 4 Calculation overhead

Consequently, in the two stages of inquiry within the game between Adversary A and Challenger B, dversary A's advantage in winning the game is only minimal, suggesting that no adversary can successfully break the encryption algorithm, according to the security analysis described in Definition 2 and under the random oracle model. This situation is equivalent to IND-CCA2 security. On the other hand, the system is considered to have IND-CCA1 security if Adversary A only has a slight advantage in the initial query stage under the random oracle model, suggesting that no adversary can break the encryption technique. Lemma 2: The voting data of voters in the scheme is encrypted and saved using the associated identity IDi, in accordance with the DBDH assumption and the random oracle model. The system is guaranteed to be IND-ID-CCA secure since its encryption algorithm shows indistinguishability under adaptive chosen-ciphertext attacks.

Lemma 2: The voting data of voters in the scheme is encrypted and saved using the associated identity IDi, in accordance with the DBDH assumption and the random oracle model. The system is guaranteed to be IND-ID-CCA secure since its encryption algorithm shows indistinguishability under adaptive chosen-ciphertext attacks.

Proof: Assume that a probabilistic polynomial time PPT exists, which is equivalent to Lemma 1. The indistinguishability game between adversary A and challenger B under the adaptive choice ciphertext attack in polynomial time is as follows, given an encryption scheme II = (Gen, En, De):

1 Initialization: B creates the public system parameter w and the secret master key SK_m after entering the security parameter `. represented by (w, SK_m) → Init(λ).

2 Inquiry 1: The outcome of the prior inquiry determines how the current inquiry is carried out. Run an OracleSK() query to generate the secret key for ID.
B creates a secret key SK that corresponds to ID using the secret key generation technique, then transmits SK to the adversary.

Privacy and performance analysis:

Voter and candidate identification privacy and ballot data privacy are the two main components of the scheme's privacy. Ring signatures, which are validated to assure their authenticity, protect the anonymity

of voters' and candidates' identities. Voting information is encrypted and stored using the voter's identity, which also serves to safely show that the encryption algorithm is indistinguishable from adaptive selected ciphertext assaults. Secure multi-party vote summation and sorting computations are carried out in an encrypted state using a threshold key-sharing method for ballot data privacy.

Lawfulness:
Voters and candidates can be granted unique identity IDs, which will allow them to cast ballots, when their registration has been completed and their information has been confirmed to be accurate (the format and length of the registration details must meet the standards). The voting procedure cannot continue otherwise.

Accuracy:
The voting plan's many stages all reflect correctness. It is feasible to confirm voters' and candidates' identities during the registration phase. are valid before to casting a ballot. Each voter may only cast one ballot during the voting phase, and ballots that are cast more than once are deemed invalid. Accurate vote counting is ensured at the vote counting stage by processing the vote information using multiparty summation and multi-party sorting agreements without the help of vote-counter considerations. During the announcement phase, the candidate verifies the ballot's status and decrypts the ballot that corresponds to their identity IDi on the blockchain.

Wholeness:
The contract confirms that the number of candidates and voters meets the necessary requirements, and voters and candidates must register within the allotted registration period. Additionally, the voter verifies and signs the ballot and identification using the ring signature algorithm, making sure that the signature cannot be forged.

Individuality:
Voters are only given one chance to cast their ballots. Every voter is given a distinct identifying IDi linked to their ballot at the time of registration. A vote cannot be recast once it has been certified; doing so invalidates the vote. Avoiding repeated voting is necessary since it may affect how accurately the results of the counting are determined.

Verifiability:
Following the voting process, the blockchain stores the index information and the IPFS distributed file system stores the ballot information. Anyone with access authority can compare the vote data saved in IPFS with the blockchain index data since the blockchain data is unchangeable. The key counting procedures are documented on the blockchain for traceability and verification at the end of the vote counting phase. The candidate saves the final vote results on the blockchain after encrypting them. The amount of votes each candidate has received can also be confirmed by the candidate or other authorized persons.

Not a receipt:

To prevent hackers from altering the election results, the ballots are encrypted and processed using a threshold key-sharing scheme. They can only be decrypted after the threshold is reached. At the same time, voters cannot fabricate receipts that deviate from their real votes, and candidates cannot present receipts as evidence of their votes.

Reluctance to attack:

During the designated registration time, voters and candidates are assigned a unique identification IDi by the synchronization process. While keeping the index data on the blockchain, they send ballot information to the contract, encrypt the data pertaining to the ballot, and store it in the IPFS distributed file system. Voters and candidates are only aware of their own identities during the voting process to aid in identity verification and ballot information. This keeps them from knowing specific identities and, consequently, stops candidates from banding together with voters to solicit votes, which would compromise the integrity of the entire voting process. We use threshold keys to encrypt votes and signatures to anonymize identities in order to stop voters from banding together to change the voting outcomes.

Equity:

The MPC nodes in the computing network carry out the scheme's vote counting. Secure multi-party sorting and secure multi-party summing are its two phases. Every stage's outcomes are documented on the blockchain, making traceability, viewing, and verification easier. Fairness is ensured via the secret sharing procedure, which stops voters from banding together to reveal votes.

Comparison of schemes:

Table 5 below shows the comparison of this paper with related work using performance analysis. The first plan to do selfcounting was put forth in Reference [10] and used a smart contract with a zero-knowledge proof encryption technique. Although this strategy can ward off replay attacks, it suffers from a cap on computation and storage capacity. According to reference [13], a distributed, privacy-protected, end-to-end verifiable electronic voting system is introduced to address the problem of online voting systems having a single point of failure and being unable to ensure vote privacy.

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	✓	✓	✓	✓

Table 5 Comparison of electronic voting schemes

Conclusion:

In this work, a blockchain-based, multi-party verifiable electronic voting system is presented. The management contract confirms the validity of voting identities, and the method uses ring signature technology to guarantee the secrecy of voters' and candidates' identities. In order to improve counting efficiency, the calculation contract mainly carries out prescribed computing activities, such as multi-party summation and sorting protocols. The outcomes of key calculations are returned and saved on the blockchain for convenient access and validation. In order to overcome the low operating efficiency brought on by insufficient block storage, the scheme makes use of IPFS. The candidate or other allowed individuals can decrypt, validate, and read the ballot information and identity that is encrypted by the candidate's public key and published on the blockchain. Lastly, the scheme's dependability and security are proven by the performance analysis and security proofs. The security of the contract itself is not covered in this study; that will be covered in later research.

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
7. 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. 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

9. Wang Z, Cheung SCS, Luo Y (2016) Information-theoretic secure multiparty computation with collusion deterrence. IEEE Trans Inf Forensics Secur 12(4):980–995
10. Ji ZX, Zhang HG, Wang HZ et al (2019) Quantum protocols for secure multi-party summation. Quantum Inf Process 18(6):1–19
11. Tang C, Shi G, Yao Z (2011) Secure multi-party computation protocol for sequencing problem. Science China Information Sciences 54:1654–1662 12. 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
12. Dery L, Tassa T, Yanai A (2021) Fear not, vote truthfully: Secure Multiparty Computation of score based rules. Expert Systems with Applications 168:114434
13. Sutradhar K, Om H (2021) An efficient simulation for quantum secure multiparty computation. Sci Rep 11(1):2206