Multi-Candidate Voting Model Based on Blockchain
Dongliang Xu, Wei Shi, Wensheng Zhai, and Zhihong Tian

Abstract—Electronic voting has partially solved the problems of poor anonymity and low efficiency associated with traditional voting. However, the difficulties it introduced into the supervision of the vote counting, as well as its need for a concurrent guaranteed trusted third party, should not be overlooked. With the advent of blockchain technology in recent years, its features such as decentralization, anonymity, and non-tampering have made it a good candidate in solving the problems that electronic voting faces. In this study, we propose a multi-candidate voting model based on the blockchain technology. By introducing an asymmetric encryption and an anonymity-preserving voting algorithm, votes can be counted without relying on a third party, and the voting results can be displayed in real time in a manner that satisfies various levels of voting security and privacy requirements. Experimental results show that the proposed model solves the aforementioned problems of electronic voting without significant negative impact from an increasing number of voters or candidates.

Index Terms—blockchain, voting, voting anonymity confusion algorithm.

I. INTRODUCTION

A. Motivation and Significance

Electronic voting is an online voting method based on cryptography technology. Voters can vote conveniently online through mobile phones, computers, and other terminals [1]. This method will automatically count the votes and display the voting results. However, existing electronic voting models are frequently disputed by the public due to their opacity and vulnerability to loss of voting data, breach of personal privacy, and hackers' attacks stemming from the centralization of data storage and the generated voting results.

Since the birth of Bitcoin [2] in 2009, digital currency has steadily gained attention. From 2014 onwards, researchers have intensively focused on its underlining technology, namely blockchain. Its decentralization, anonymity, and non-tampering characteristics have lead to researching its applicability to credit investigation, trade, finance, internet of vehicles [3], smart campus virtualization platform [4] and other fields.

The application of blockchain technology to electronic voting can significantly improve its anonymity, eligibility, and impartiality through data decentralization, anonymity, non-tampering, and other features, as well as its functions such as automatic vote counting and examination with a smart contract. Szabo [5] defined smart contracts as a set of commitments in a digital form, including agreements on which contract participants can enforce these commitments. An agreement is a set of rules that both parties must abide by, and commitments include contract terms for implementing business logic and rules-based operations that define the nature and purpose of the contract. The digital form indicates that the contract will be composed of codes whose output can be predicted and executed automatically. Developing an efficient and effective smart contract that adapts to necessary security level remains a problem to be tackled. Based on blockchain and smart contracts, this study uses an elliptic curve encryption algorithm for signature and verification. This study proposes a general electronic voting model based on blockchain, which is implemented and tested in Hyperledger Fabric. Furthermore, this study compares the security objectives and application scope with existing models. It leaves the investigation of a new secure blockchain-based solution for electronic voting to the future work.

B. State of the Art

With the development of Internet technology [6] and cryptography, online electronic voting has become a new voting solution, which can effectively solve the disadvantages of traditional paper voting such as high cost, tedious steps, and numerous errors. Existing electronic voting models mainly include ring and blind signature-, anonymous channel- [7], homomorphic encryption-, and hybrid network-based electronic voting models. Generally, ring and blind signatures require anonymous channel and a trusted third party (TTP) as signatories, whereas homomorphic encryption and hybrid network have high computational complexity, making them difficult be put into practice.
Chaum [8] proposed the first electronic voting model, which adopted the public key cryptosystem, and used digital false name voting to hide the identity of the voter; thus, any party concerned could verify whether the ballot was fully counted by using digital false name voting, thereby realizing the integrity of votes counting. In 1985, Cohen and Fisher [9] suggested an electronic voting model based on homomorphic encryption, which required the entire voting process to be kept synchronous. Benaloh et al. [10], Sako et al. [11], and Iverson [12] presented their own electronic voting models based on homomorphic encryption, but each model has advantages and disadvantages and is impractical with large-scale transmission and high-intensity calculation, thereby making them unsuitable for large-scale voting.

The first practical large-scale scenario of an electronic voting model (i.e., FOO) was proposed by Fujioka, Okamoto, and Ohta in 1992 [13]. In this model, the roles of voting participants are voting initiator, voter, and vote-counting persons, who use blind signature and bit-promise technology to encrypt the voting information and send it to a TTP to count the votes. Evidently, the model can neither abandon the vote nor completely guarantee whether the TTP is credible. However, the model still attracts considerable attention, and many universities and research institutions have improved and developed the corresponding electronic voting software system. However, any of the above-mentioned electronic voting models, including the FOO, must be based on a TTP, a credible counting body, to count the votes. Such centralization is a huge risk to voters' privacy and will lead to data tampering and loss. Given that these protocols have a TTP and require voters to trust the TTP, users are unable to detect and defend against collusion attacks by counting agencies and regulators. The existence of these problems will lead to the distrust of users and thus restrict the promotion of electronic voting.

In 2002, Kiayias and Yung [14] proposed the first electronic voting model that was capable of self-counting, which used open third-party voting procedures to count votes; moreover, these researchers introduced a fault-tolerant mechanism, which is applicable to small-scale scene voting, to correct “faults” in the voting process. Hao, Ryan, and Zielinski [15] presented a two-round anonymous voting model with a self-counting function in 2010, thereby significantly improving the efficiency, calculation cost, and number of rounds because this model did not require anonymous channels. Such electronic voting models with self-counting capability weaken the TTP, convert the counting process into a publicly verifiable process, and allow any party concerned to perform and verify the counting process after all votes have been cast, thus weakening the unique status of counting institutions and enhancing the credibility of electronic voting.

In recent years, with the popularity of Bitcoin, the underlying blockchain technology has also gained the attention of many researchers. As shown in Fig. 1, blockchain [16] is involved in any number of nodes in the system. All information is stored over a period of time through the calculation of cryptography algorithm and records in data blocks. In addition, the generated data “password” is used to verify the validity of the information and links to the next block and jointly by all participating node systems to decide whether the record is authentic. In the blockchain, in order to maintain the same data in the whole chain and ensure the fairness of each participant, all participants must have a unified protocol; that is, the consensus mechanism solves the problem of unifying data [17] in the blockchain and the problem of trust caused by proceeding to the center [18].

Blockchain is essentially an open, transparent, non-tamperable distributed database ledger technology [19] that records all transaction information. In the absence of third-party intermediaries, blockchain has the characteristics of decentralization and non-tampering, which is conducive to solving problems such as difficult verification, privacy disclosure, and tampering with votes. Moreover, a smart contract can complete the self-counting function and realize the automatic vote counting process.

Recently, researchers have developed many applications in accordance with these characteristics. James et al. [20] proposed a blockchain-based voting platform in 2016 to conduct national level elections. The paper only gives a description of the process of the model and requires a trusted third party to conduct hidden user votes and vote counting.Yan et al. [21] used the distributed ElGamal encryption system and zero-knowledge proof protocol; he also designed and developed a blockchain voting platform based on e-therem and adopted the digital signature identity access mechanism to ensure the identity legitimacy of voters. In this paper, we propose a secure electronic voting model for additional candidates, and a smart contract, which was implemented in the TTP. The proposed model is designed to replace the traditional voting protocol in order to decrease the cost of the trust system. Using a digital signature (i.e., the
identity of the access mechanism to ensure the legitimacy of voters’ identity), a voting blockchain platform is designed and developed on the basis of the Etheric Fang platform.

Cramer et al. [22] first proposed the problem of multi-candidate voting in 1996 and suggested a “1-out-of-m” multi-candidate voting model (i.e. selecting one candidate among multiple candidates). However, considering that the model uses the ElGamal homomorphic encryption system, the calculation is complex. Thus, the model cannot be extended from 1-out-of-m to k-out-of-m (i.e. multiple candidates are selected from multiple candidates) type voting. In 2006, Zhong [23] proposed a k-out-of-m multi-candidate voting protocol that combined secure multi-party and multi-precision computation without central trusted institutions. This model satisfies the characteristics of anonymity, fairness, and non-repeatable voting of electronic voting and is a favorable solution.

C. Elliptic Curve Cryptography (ECC) Algorithm

The encryption algorithm and key distribution [24] and management mechanisms [25] used in this study are mainly based on ECC. It is used to establish public key encryption [26] based on elliptic curve mathematics. The use of elliptic curve mathematics in cryptography was introduced in 1987 by Koblitz [27] and Miller [28].

The mathematical basis of ECC is the insolubility of elliptic curve discrete logarithm problem (ECDLP). Discrete logarithm problem on elliptic curve: known elliptic curve $E_p(a,b)$ on finite field $F_p$, and the point $G(x,y)$, $X = xG$, $x < n$, $nG = O^n$, $n$ is the order of $G$, solving the $x$. The difficulty of this problem guarantees the security of ECC. In the elliptic curve encryption algorithm, $X$ is the public key and $x$ is the private key. From the property of the elliptic curve, the public key is easy to find with the known private key, whereas the private key is nearly infeasible to find with the known public key. Therefore, we use ECC for encryption and digital signature.

D. Security Requirements for Blockchain-Based Electronic Voting Models

After studying the existing electronic voting models and the characteristics of blockchain, we introduce the following requirements that a blockchain-based secure [29] electronic voting model must satisfy:

1) Legitimacy: The voting activity initiated by the poll sponsor shall be reviewed to ensure that the content of the poll is presented in accordance with the requirements of the law. Moreover, voters must satisfy the voting requirements and thus unqualified voters are prohibited from voting.

2) Anonymity: The identity of the voter is hidden. The content of the ballot will be encrypted to prevent the ballot information from being stolen maliciously and then leaked to affect the voting process.

3) “Unforgeability”: The attacker cannot forge the voting information, and the recorded legal voting information cannot be tampered with. Voters can verify whether their votes are correct and valid through the model to ensure that the voting results cannot be forged.

4) Self-counting: A third party or a manual vote counting is no longer necessary when the final result of a voting is counted by the model and the vote results are automatically updated. The voting process shows when a vote is cast, and the voting result changes accordingly without revealing the voter's specific voting content.

II. K-out-of-M Electronic Voting Model

In this paper, we propose an electronic voting model of k-out-of-m (i.e. voters can select $k, k \in [0, m]$ from $m$ candidates) based on the blockchain technology. The model uses the characteristics of blockchain to decrypt every ballot under the premise of guaranteeing the anonymity of voters. Namely, the model can perform complex transmission of voting information without compromising voters’ identities. Most importantly, the model is not limited to a specific implementation of a certain blockchain. The voting model can be realized provided that the blockchain platform uses ECC asymmetric encryption and has access to mechanism and a smart contract.

A. Model Establishment

In this model, the main participants are the counting node $V_0$, the verified voters $(V_1, V_2, ..., V_a)$, the candidates $(C_0, C_1, ..., C_{m-1})$, voting initiator $I$, and $K$ candidates ($K$ is a variable). Among them, the voting initiator $I$ must ensure that the content of the voting is in compliance with relevant laws and regulations, and voters must be verified to have the right to vote to enter the blockchain network (such as uploading some certification materials). Given that the two requirements mentioned above have different verification standards in various situations, this model assumes that the two requirements have been established.

1) The Counting Node and Voters: The counting node $V_0$ in the model is automatically generated by the smart contract after $I$ successfully initiates the voting and is deployed on the node provided by the initiator. This node becomes the counting node $V_0$ and cannot be logged into by any user. It only serves as the communication object of each voter and is responsible for recording and displaying the voting situation. Counting nodes can be considered as special contract accounts in the existing ethereum and Hyperledger Fabric platforms.

\[ X_{V_0} = x_{V_0}G(x_{V_0} < n, nG = O^n) \]

where $G(x,y)$ is a point in the elliptic curve $E_p(a,b)$, in which, in the given finite field $F_p$, $n$ is the order of $G$; $X_{V_0}$ is the public key of $V_0$; and $x_{V_0}$ is the private key of $V_0$. The vote-counting node $V_0$ generates $X_{V_0}$ and $x_{V_0}$ using Formula (1) and broadcasts its $X_{V_0}$ to all nodes before the voting start time $T_{start}$.

Voters in the model $(V_1, V_2, ..., V_a)$ are the user nodes that can be logged into and constitute the main body of the voting activity. After legitimacy verification, the user can access the blockchain network and become the new voter. Voters select candidates and vote. Voters as nodes are part of the blockchain and keep accounts in accordance with the consensus mechanism to ensure the authenticity and credibility of information on the chain.
where \( G(x,y) \) is a point in the elliptic curve \( E_p(a,b) \), in which, in the given finite field \( \mathbb{F}_p \), \( n \) is the order of \( G \); \( x_{V_1 \cdots V_n} \) are the public keys of \( (V_1 \cdots V_n) \); and \( x_{V_1 \cdots V_n} \) are the private keys of \( (V_1 \cdots V_n) \). The vote-counting nodes \( (V_1 \cdots V_n) \) generate \( x_{V_1 \cdots V_n} \) and \( x_{V_1 \cdots V_n} \) using Formula (2) and broadcast their \( x_{V_1 \cdots V_n} \) to all nodes before the voting start time \( T_{\text{start}} \).

2) Voting Information: The model supports \( k \) votes out of \( m \) type voting, and the voting information \( \text{mes} \) encoding mode is a \( m \)-bit binary number converted into a decimal number.

3) Voting Anonymity Confusion Algorithm: The use of blockchain for electronic voting also raises new questions. In the case of voting, for example, the encrypted information transmitted during a vote and the time will be recorded on the blockchain. Among them, \( \text{mes} \) is invisible through encryption. However, in accordance with the voting time of different voters recorded on the chain and the number of votes changing in real time displayed by \( V_0 \), the voter's \( \text{mes} \) is completely infeasible to infer. Given this situation, we propose an Anonymity Preserving Voting (APV) algorithm for voting to ensure the anonymity of the model.

**Algorithm 1 Anonymity Preserving Voting (APV) algorithm**

**Require:** \( P \) //Maximum number of mixed voters

**Ensure:** \( R(r_1,r_2,\cdots,r_m) \) //The number of votes

\[ \text{start} = 1 \]

\[ \text{rest} \_c = C \] //The number of remaining voters

**while** \( \text{rest} \_c > P + 1 \) **do**

//The number of remaining voters is greater than maximum number of mixed voters plus one

\[ p = \text{Rand}(2,P) \] //Generating a random number \( p \in [2,P] \)

\[ g = (\text{mes}_{\text{start}_{p-1}} \cdots \text{mes}_{\text{start}_{p+1}}) \] //\( V_0 \) received \( p \) of \( \text{mes} \) and divided them into group \( g \)

**for** \( \forall \_r \in g \) **do**

Wait(\( p \)) //Wait for a multiple of \( p \)

Update(\( R_r \)) //Update the total number of votes \( R \) based on \( r \)

\[ \text{rest} \_c = \text{rest} \_c - p \]

\[ \text{start}_{p} = \text{start}_{p} + p \]

\[ g = (\text{mes}_{p-1} \cdots \text{mes}_{p+1}) \] //\( V_0 \) received the last \( p \) of \( \text{mes} \) and divided them into group \( g \)

**for** \( \forall \_r \in g \) **do**

Wait(\( p \)) //Wait for a multiple of \( p \)

Update(\( R_r \)) //Update the total number of votes \( R \) based on \( r \)

\[ X_{V_1 \cdots V_n} = x_{V_1 \cdots V_n} G(x_{V_1 \cdots V_n} < n; nG = O^m) \] (2)

**B. Model Specification**

1) Voting Information Transmission: As shown in Fig. 2, the voting information \( \text{mes} \) is transmitted after encryption in the blockchain. The following information is an example of the voting data transmission of the voting node \( V_1 \) to illustrate the information encryption transmission and verification method of the model.

a) \( V_1 \) generates \( \text{mes}_{V_1} \): \( V_1 \) selects \( k \) candidates to vote for. The selected candidates code is 10..01, and voting information is \( \text{mes}_{V_1} = 2^{m-1} + 1 \).

b) \( V_1 \) generates \( D_{V_1} \):

- \( R(x,y)_{V_1} = r_{V_1} G(x,y)_{V_1} \)
- \( c_{V_1} = x(R_{V_1}) \mod n_{V_1} \)
- \( s_{V_1} = k^{-1}(\text{Hash}(\text{mes}_{V_1}) + x_{V_1} c_{V_1}) \mod n_{V_1} \)
- \( D_{V_1} = (c_{V_1}, s_{V_1}, \text{mes}_{V_1}) \)

where \( r_{V_1} \) is a random integer selected by \( V_1 \) (\( r \) is not necessarily equal to \( r' \)), \( x(R_{V_1}) \) is the abscissa of \( R_{V_1} \). The encrypted \( El_{V_1} \) is generated by \( V_1 \) using \( X_{V_0} \) in accordance with the Formula (1).

c) \( V_0 \) decrpts \( CI_{V_1} \): \( V_0 \) received \( CI_{V_1} = (c_1, c_2) \) from \( V_1 \).

\[ R'_{V_0} = x_{V_0} c_{1} \]

\[ D_{V_1} = c_{2} x(R'_{V_0})^{-1} \]

where \( x(R'_{V_0}) \) is the abscissa of \( R'_{V_0} \), \( V_0 \) decrypts \( El_{V_1} \) to determine \( D_{V_1} \) in accordance with the Formula (12) \( D_{V_1} \) already contains \( \text{mes}_{V_1} \), but \( V_0 \) must also verify the signature to ensure that the vote is sent from \( V_1 \).

d) \( V_0 \) verifies the signature of \( D_{V_1} \):

\[ b_1 = \text{Hash}(\text{mes}_{V_1}) x_{V_1}^{-1} \mod n_{V_1} \]

\[ b_2 = c_{V_1} s_{V_1}^{-1} \mod n_{V_1} \]

\[ R''_{V_0} = b_1 G_{V_1} + b_2 X_{V_1} \]

\[ c_{V_1} = x(R''_{V_0}) \mod n_{V_1} \]

If this is true, then \( \text{mes}_{V_1} \) is confirmed to be sent by \( V_1 \) and is thus not a forgery.

2) Voting process: After the voting is successfully initiated, \( I \) sets the voting start time \( T_{\text{start}} \) and the voting end time \( T_{\text{end}} \). Before \( T_{\text{start}} \), \( V_0 \) generates \( X_{V_0} \) and \( x_{V_0} \), and broadcasts \( X_{V_0} \) \( n \) qualified voters \( (V_1, V_2, \ldots, V_n) \) access the blockchain network and generate \( \{X_{V_1}, X_{V_2}, \ldots, X_{V_n}\} \), \( \{x_{V_1}, x_{V_2}, \ldots, x_{V_n}\} \), and broadcast \( \{X_{V_1}, X_{V_2}, \ldots, X_{V_n}\} \). Between \( T_{\text{start}} \) and \( T_{\text{end}} \), each voter selects \( k \) candidates and sends the encrypted information to the counting node \( V_0 \), \( V_0 \) is responsible for identifying the voter, counting the qualified votes, and displaying the current voting result. As shown in Fig. 3, the following is an example of the voting process of the voting node \( V_1 \) to illustrate the voting process of this model:

\( V_1 \) selects \( k \) candidates, so the selected candidate code is 10..01, form information \( \text{mes}_{V_1} = 2^{m-1} + 1 \). \( D_{V_1} \) is signed.
using $mes_{V1}$ and $x_{V1}$ in accordance with the above-mentioned signature method. $EI_{V1}$ gets encrypted using $DV_{V1}$ and $X_{V1}$ in accordance with the above-mentioned encryption method, and $EI_{V1}$ is sent to $V_0$.

After receiving $EI_{V1}$ from $V_1$, $V_0$ decrypts $EI_{V1}$ to determine $DV_{V1}$ under the control of the smart contract and then verifies the signature in $DV_{V1}$ using $X_{V1}$; if successful, then $mes_{V1}$ is sent by $V_1$ and is not faked. Then, $X_{V1}$ is used as a mark to verify that this voter is a first-time voter. Afterward, $mes_{V1}$ is counted, and $V_1$ is marked as having voted. The voting information $EI_{V1}$ is written into the blockchain.

$V_0$ automatically counts the qualified votes received and shows the voting result using Algorithm 1 under the control of the smart contract.

When the time reaches $T_{end}$, all users who have not voted will automatically generate voting information $mes = 00...00$. Then, it will be sent to $V_0$. After $V_0$ has received the information of all voters, the final voting result is calculated.

3) Decentralized: In the previous sections, we use the concept of a counting node in the proposed model. It generates voting information and stores it in the blockchain network during interactions with the voters. However, the counting node is not a centralized database or a TTP in the traditional sense. Instead, it is essentially a node deployed by one or a group of smart contracts in the blockchain. The difference between the counting node and a voting node is: certain smart contracts must be executed on the counting node in order to complete the counting function. All the encrypted information generated by a voting activity will be recorded in the block, and the block is confirmed to be valid through the consensus mechanism, then stored in each participating node. Furthermore, the ballot can be modified solely by the counting function, which avoids the centralization risk of traditional electronic voting strategies.

### III. EXPERIMENT AND RESULT ANALYSIS

We have proposed a $k-out-of-m$ voting model based on blockchain. In this section, the Hyperledger Fabric implementation model is adopted, and Hyperledger Caliper is used for the performance test to evaluate the accuracy and security properties of the model and compare it with other models. The Hyperledger Fabric [30] is a blockchain framework implementation by The Linux Foundation, which leverages container technology to host smart contracts called “chaincode” that comprise the application logic of the system. The node deployed by chaincode can be considered our voting node.

#### A. Experimental Environment and System Architecture for Model Validation

Docker Compose [31] is a tool for defining and running multi-container Docker applications. We can use a YAMl file to configure the application's services with it. Then, with a single command, we implement and start all the services from our configuration in the YAMl file. Therefore, we use Docker Compose to start multiple node containers on a single machine with the Hyperledger Fabric, rather than multiple physical machines, for experiments. Fig. 4 illustrates the system architecture for model validation based on the Hyperledger Fabric.

In Fig. 4, voters use voting peers, which can be seen as voting nodes by the voting system, to access the blockchain network. The Certificate Authority (CA) node guarantees the access of voting nodes is legitimate; the orderer node guarantees the consistency of the voting information of each voter; the endorser node takes charge of verifying signatures and the smart contract node takes charge of implementing smart contracts. The endorser node and the smart contract node can together be regarded as the counting node.

#### B. Model Performance Test

We used Hyperledger Caliper [32] for performance testing [33]. This tool is used for a blockchain performance benchmark for Hyperledger Burrow, Fabric, Iroha, and Sawtooth. Currently, the supported performance indicators are success rate, transaction, read throughput transaction, read latency, and resource consumption. Moreover, the system uses a consensus mechanism based on Kafka [34].

1) Relationship Between Number of Voters and Model Performance: The relationship between the number of voters...
and the performance of the model was tested. In the fabric network we implemented, the number of voters is the same as that of peer nodes, and the number of candidates is 10. When testing the performance of the model, the limit on the number of votes is temporarily removed, that is, each node can vote without limit. This is because if the number of votes is limited, the number of pieces of valid voting information will never exceed the number of nodes, and the performance cannot be tested in the case of a small number of nodes. In our experiment, each node votes or queries 1000 times at a specific frequency, referred to as the Send Rate. The experimental results are presented as follows:

In Figs. 5 and 6, vote refers to the voting operation, query refers to the operation of searching the voting result, Send Rate refers to the number of operation requests per second, Avg Latency refers to the average time delay of each operation, and Throughput refers to the number of completed operation requests per second. The delay of query illustrated in Fig. 5 is approximately 0.02s, and the throughput rate depicted in Fig. 6 is close to the Send Rate. That is, every operation request can be processed in time, and the number of voters has no impact on the performance of query operation. In Fig. 5, a vote's delay increases rapidly with the Send Rate and slightly with the number of nodes. In Fig. 6, the increase in the Send Rate of a vote's time-dependent throughput rate rises slightly. Furthermore, the throughput rate decreases slightly with the increase in the number of nodes. Moreover, in actual conditions, each node will only conduct a qualified vote once, and the delay and throughput rate of the voting operation are acceptable. As illustrated by the experimental results, the throughput rate and time delay of the voting operation as well as the viewing of voting results in the small-scale voting scenario satisfy the requirements of the model.

2) Relationship Between the Number of Candidates and Performance: The model we designed is suitable for the “k-out-of-m”-type voting because the size of the voting information depends on m. Thus, we must test the impact of different numbers of candidates on the performance of the model. The following experiment uses $4peers - 1order$ (i.e. the number of voters is 4), and the voting quantity limit is temporarily removed. Each node votes or queries by 1000 times in accordance with the different frequencies. The experimental results are presented as follows:

In Figs. 7 and 8, vote refers to the voting operation, query refers to the operation of searching the voting result, Send Rate refers to the number of operation requests per second, Avg Latency refers to the average time delay of each operation, and Throughput refers to the number of operation requests that can be completed per second. In Fig. 7, when the number of candidates is not too large, it slightly influences the voting delay. Only when the number of candidates is very

---

Fig. 4. System architecture diagram for model validation.

Fig. 5. Avg. Latency with different numbers of voters.

Fig. 6. Throughput with different numbers of voters.
large, the delay is too increased. The number of candidates has a very minor influence on the delay of the query. In Fig. 8, when the number of candidates is relatively small, it slightly influences the throughput of the voting. Only when the number of candidates is significantly large, the throughput declines significantly, whereas the number of candidates only slightly influences the throughput of a query. The number of candidates affects only the binary digits of the voting information. Thus, the influence the number of candidates has on the vote count has \( O(n^2) \) complexity. However, given that the counting process is simple, it only slightly influences the performance. The effect of the number of candidates on Algorithm 1 is \( O(n) \) complexity, with minimal impact on the performance.

C. Self-counting Accuracy Testing

We tested the accuracy of self-counting by comparing the real results with the results of self-counting. The actual voting time and the voting results were counted simultaneously to test the validity of Algorithm 1. In this experiment, the maximum number of mixed votes per node is 3. In addition, 10peers – 1order (i.e., the number of voters is 10) is adopted, and the experiment is conducted in the case of 4 candidates. The experimental results are illustrated in Table II.

The information presented in Table II is the voting information of the current voter, and the self-counting result is the real-time counting result displayed by the system when the current voter votes. Considering that the value of \( p \) cannot be determined each time, the display delay of the voting results is completely random. In Table II, the voting information of \( V_3 \) is 0010, and the difference between the self-counting result when \( V_3 \) votes and the self-counting result when \( V_9 \) votes are 0,1,1,1. Thus, the voting information of \( V_3 \) cannot be inferred. Therefore, when the voting time and the self-counting result time are known, the specific conditions of voters’ votes cannot be deduced regularly, but the current voting results can be displayed with a short delay, and the votes can be accurately counted after the voting ends. In addition, the entire vote-counting process is conducted using a smart contract, and no artificial vote-counting drawbacks are observed. Given that the smart contracts are fully public, the voters can trust the results. Therefore, under the premise of guaranteeing anonymity, the model has the property of accurate self-counting.

D. Model Security Testing

The legitimacy of voting is guaranteed by the access audit of blockchain. Given the different legitimacy standards of voting, legitimacy is not discussed in this study. However, some attackers with the intention of falsifying the voting results can still be qualified to access the blockchain network. For example, in scenarios with loose standards of legitimacy review, such as a referendum, the identity of attackers and eligible voters can be overlapped; that is, attackers may be part of the eligible voters. On this basis, we tested the security properties of the model.

To facilitate testing the performance of the model, we must remove the limit on the number of votes, but the security test must limit each voter to one eligible vote. Subsequently, we used the node configuration of 10peers – 1order, set the maximum number of mixed votes to 3, and conducted multiple experiments in the case of 4 candidates to test the security of the proposed voting model.

The experimental results are provided in Table III, assuming the following scenario: \( V_A \) is a qualified voter and an attacker in the voting link, and \( V_1 \) is a non-attacker voter, and decryption is a necessary step of the query. Results show that \( V_1 \) decrypting \( EI_{V_1} \) successfully, but \( V_1 \) voting with \( X_{V_1} \) and \( V_A \) decrypting \( EI_{V_A} \) both fail. \( V_1 \) voting with \( X_{V_1} \) succeeded only once. The above-mentioned experiments confirm the two
The experimental results and arguments summarized in Table III confirm that the specific voting information of voters cannot be inferred with the public information provided when Algorithm 1 is used in the voting model. Since voters cannot determine the specific voting information of other voters, we conclude that the proposed model has achieved anonymity.

### E. Comparison with Existing Models

We compared all the voting models discussed in this paper in terms of their voting methods, voting types, TTPs, counting methods, and the presence or absence of the mixed voting mechanism. The models involved in the comparison are all relatively mature voting models, which basically satisfy the anonymity and non-forgery requirements.

The comparison results show (see Table III) that in terms of voting methods, comparing with the FOO model, our proposed model uses the blockchain technology, therefore, problems caused by traditional electronic voting methods such as the requirement for a TTP and the inability to unify votes in real time are avoided. However, given the limitation of the blockchain technology, the throughput rate is lower in our model than in the traditional electronic voting when the number of candidates scales up. This means that our model is suitable for small-scale voting scenarios. In terms of voting types, our proposed model supports “k-out-of-m” voting, which includes the “1-out-of-m” voting type of Yan's model [21] with strong voting applicability. This model does not require a TTP to guarantee the voting process and results. Comparing with the model in literature [20], third-party vote-counting fraud is not an issue for our proposed model. Furthermore, our proposed model uses smart contract automatic vote counting, which is simpler and more efficient than the whole vote counting proposed in the model in literature [23]. Because the smart contract code is open, the vote counting results are trustworthy. The vote counting result accuracy obtained by our proposed algorithm is much higher than that of the automatic vote counting proposed in the model in literature [15]. In particular, Algorithm 1 proposed in this study solves the problem of inferring the specific voting information of a voter based on the available voting time and real-time voting results in the blockchain voting model. Thus, the model can display the voting results in real time without compromising the voters’ anonymity. In general, this model has great advantages in terms of anonymity, non-forgery, and self-counting and has disadvantages in its throughput rate in large-sized voting scenarios. Our proposed model is more suitable for small- and medium-sized voting scenarios.

### IV. Conclusion

In this paper, we propose a “k-out-of-m” multi-candidate voting model based on blockchain technology that ensures: 1). the preservation of anonymity during the voting process through ECC encryption and a signature mechanism; 2). the prevention of forgery on voting information by combining blockchain technology, automatic statistics and display of voting results through a smart contract. Furthermore, this study introduces an anonymity-preserving voting (APV) algorithm to mitigate the drawbacks of the blockchain voting

<table>
<thead>
<tr>
<th>Test</th>
<th>Operation times</th>
<th>Number of successes</th>
<th>Number of failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1$ votes with $X_{V_1}$</td>
<td>100</td>
<td>1</td>
<td>99</td>
</tr>
<tr>
<td>$V_A$ votes with $X_{V_1}$</td>
<td>100</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>$V_A$ decrypts $C_{IV_{V_1}}$</td>
<td>100</td>
<td>0</td>
<td>1000</td>
</tr>
<tr>
<td>$V_1$ decrypts $C_{IV_{V_1}}$</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
</tbody>
</table>
method. We use the Hyperledger Fabric framework to conduct experiments on the proposed model. Hyperledger Caliper is used to test the performance of the model. Experiments are designed and tested extensively with respect to the security requirements of the model as well as the effectiveness and efficiency of the proposed AVP algorithm. Through these experiments, we confirm that the model is particularly well-suitied for small-scale voting situations and overcomes the disadvantages of traditional electronic voting pertaining to lack of anonymity, excessive centralization and ease of forgery.

ACKNOWLEDGMENT

This work was supported in part by Shandong Provincial Natural Science Foundation under Grant ZR2019PF007, the National Key research and Development Plan of China under Grant 2018YFB0803504, Basic scientific research operating expenses of Shandong University under grant 2018ZQXM004, Guangdong Province Key research and Development Plan under Grant No. 2019B010137004.

REFERENCES


Dongliang Xu (M’16) lecturer, ShanDong University (WeiHai). He received Ph.D. degree from Harbin Institute of Technology, Harbin, China, in 2015. His research interests include Network Intrusion Detection Systems (NIDS), Multi-pattern matching, data mining, and information security.

Wei Shi Dr. Wei Shi is an Associate professor at the School of Information Technology, cross appointed to the department of Systems and Computer Engineering, in the Faculty of Engineering and Design, at Carleton University in Ottawa, Canada. She is specialized in the design and analysis of fault tolerance algorithms addressing security issues in distributed environments such as Data-Center Networks, Clouds, Mobile Agents and Actuator Systems, Wireless Sensor Networks, as well as Critical Infrastructures such as Power Grids and Smart Cities. She has also been conducting research in data privacy and Big Data analytics. Wei holds a Bachelor of Computer Engineering from Harbin Institute of Technology (HIT) in China, as well as a Master’s and a Ph.D. of Computer Science from Carleton University. She is also a Professional Engineer licensed in Canada.

Wensheng Zhai, postgraduate student, Shandong University (WeiHai). His research interests are deep learning, Artificial intelligence security and automatic speech recognition.

Zhihong Tian is currently a Professor, and Dean, with the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangdong Province, China. Guangdong Province Universities and Colleges Pearl River Scholar (Distinguished Professor). He received the B.S., M.S. and Ph.D. degree in Computer Science and Technology from Harbin Industrial University, Heilongjiang, China in 2001, 2003 and 2006, respectively. He is also a part-time Professor at Carlton University, Ottawa, Canada. Previously, he served in different academic and administrative positions at the Harbin Institute of Technology. He has authored over 200 journal and conference papers in these areas. His research interests include computer networks and cyberspace security. His research has been supported in part by the National Natural Science Foundation of China, National Key research and Development Plan of China, and National High-tech R&D Program of China (863 Program). He also served as a member, Chair, and General Chair of a number of international conferences. He is a Senior Member of the China Computer Federation, and a Member of IEEE.