An improved secure and efficient e-voting scheme based on blockchain systems

Article

Accepted Version

Zhang, J., Wu, C., Sherratt, R. S. ORCID: https://orcid.org/0000-0001-7899-4445 and Wang, J. (2024) An improved secure and efficient e-voting scheme based on blockchain systems. IEEE Internet of Things. ISSN 2327-4662 doi: https://doi.org/10.1109/JIOT.2024.3507366 Available at https://centaur.reading.ac.uk/121258/

To link to this article DOI: http://dx.doi.org/10.1109/JIOT.2024.3507366

Publisher: IEEE

An Improved Secure and Efficient E-Voting Scheme Based on Blockchain Systems

Jingyu Zhang, *Member, IEEE*, Chenghao Wu, *Member, IEEE*, R. Simon Sherratt, *Fellow, IEEE*,
and Jin Wang, *Senior Member, IEEE*

***Abstract*—With the rapid development of the Internet of Things (IoT) and blockchain technology, e-voting has been widely used in all aspects of people's lives. However, there is a common problem in the vast majority of e-voting solutions: the inability to complete vote counting without a trusted third-party organization, which may lead to security risks. When designing an e-voting system, ensuring the trustworthiness of the voting results as well as protecting the privacy of the voters are always the most important issues. To address this challenge, we propose improved secure and efficient (ISE)-Voting, an ISE e-voting scheme for blockchain-assisted IoT devices. Our proposed ISE-Voting achieves voter privacy anonymity, distributed vote counting, and public verifiability of counting results in e-voting systems by using secret-sharing and identity-based ring signatures in the blockchain system. In addition, we introduce a cloud service provider (CSP), which is used to share the computational pressure of the system and assist ISE-Voting to complete the final counting. According to the experimental analysis and results, our scheme is not only able to meet the basic security goals of satisfying correctness, anonymity, unforgeability and verifiability, and provide 128-bit identity security for the voters in the post-quantum environment. Moreover, it can complete the distributed counting of voters' ballots within an effective time, which provides a feasible solution for future e-voting systems.**

***Index Terms*—Anonymity, e-voting, e-voting privacy, identity-based ring signature, secret sharing.**

I. INTRODUCTION

In Recent years, electronic voting has been a research hotspot in both academia and industry, and voting activities are often found in our lives, such as student elections and corporate board elections. The development of online e-voting shows the digitization and modernization of the voting process, bringing more efficiency, transparency and inclusiveness to the election process, and a typical framework of online e-voting in a blockchain system is shown in Fig. 1. The introduction of e-voting systems aims to address many of the challenges associated with traditional paper-based voting, including the time-consuming nature of the voting process, wasted resources, ballot counting errors, and difficulties in managing and analyzing voting data. The advent of e-voting systems not only simplifies the voting process for voters, but also enhances the credibility and fairness of elections. It enables voters to participate in elections over a wider geographical area and to exercise their electoral rights conveniently wherever they are. In addition, e-voting systems can provide real-time election results, providing governments, candidates and voters with more rapid feedback and data analysis, which helps better understand voter needs and political trends. The first e-voting scheme was proposed by Chaum [1] in the 1980s. However, the introduction of e-voting systems also comes with a new set of challenges and risks. For example, they all lack traceability and transparency, rely on a centralized authority, and require a trusted third party to collect ballots, verify and tally the results. The emergence of blockchain technology [2] has solved the above problems very well. As an innovative technology, blockchain is widely used in the field of the Internet of Things (IoT) [3], [4], [5]. Through the

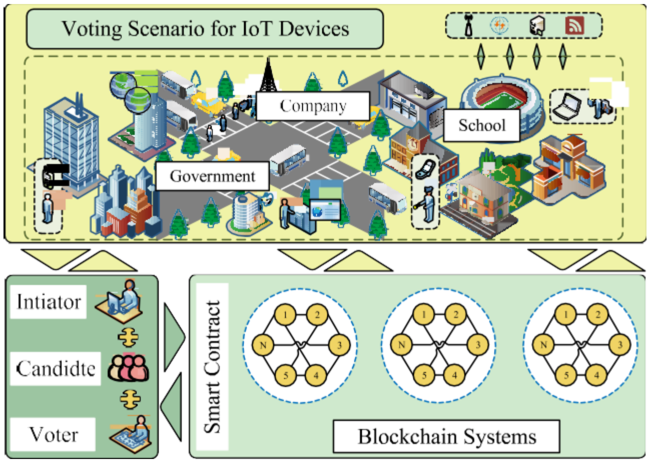


Fig. 1. Typical framework of e-voting in blockchain systems.

Received 24 May 2024; revised 16 October 2024; accepted 14 November 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62072056, Grant 62473146, and Grant U23BB2004; in part by the Hunan Provincial Key Research and Development Program under Grant 2022GK2019, Grant 2024JK2005, and Grant 2022SK2107; in part by the Natural Science Foundation of Hunan Province, China, under Grant 2023JJ30054; in part by the Hunan Provincial Natural Science Foundation Key Program under Grant 2024JJ3017; and in part by the Research Foundation of Education Bureau of Hunan Province, China, under Grant 23B0303. *(Corresponding author: Jin Wang.)*

Jingyu Zhang is with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410004, China, and also with the National Key Laboratory of Information Systems Engineering, National University of Defense Technology, Changsha 410004, China (e-mail: zhangzhang@csust.edu.cn).

Chenghao Wu is with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410004, China (e-mail: wch@stu.csust.edu.cn).

R. Simon Sherratt is with the School of Biomedical Engineering, University of Reading, RG6 6AY Reading, U.K. (e-mail: sherratt@ieee.org).

Jin Wang is with the Sanya Institute of Hunan University of Science and Technology, Sanya 572024, China (e-mail: jinwang@hnust.edu.cn).

Digital Object Identifier 10.1109/JIOT.2024.3507366

immutability of blockchain, distributed ledgers, and smart contracts, voting data can be securely stored and verified, which can ensure that each ballot is unforgeable, and all participants in the system can track and verify the results of the voting in real time, thus increasing the trustworthiness and transparency of the election, and decreasing the potential risks and errors. In traditional blockchain authentication mechanisms, public key cryptosystems are usually employed to verify user identities [6]. However, this approach carries inherent security risks, particularly concerning privacy protection. Moreover, if the device is intruded, malicious users may illegally access the private information. In this case, the e-voting system will still face the problems of authentication, data privacy protection, and trustworthiness of the voting results, which will result in serious security problems [7].

In order to ensure the security and efficiency of the e-voting scheme in the current blockchain systems, this article deeply researches the advantages and disadvantages of online e-voting schemes based on blockchain and various cryptographic security techniques. Based on this, our paper proposes an online e-voting scheme that integrates blockchain and cryptographic technologies with high security and efficiency. Our main contributions are summarized as follows.

1) We propose improved secure and efficient (ISE)-Voting, a blockchain-based e-voting solution, and it is highly secure. In addition, to fulfill the essential security properties of e-voting systems, we employ an algorithm of identity-based ring signature based on symmetric primitives.
2) To ensure the public verifiability and credibility of the counting results, we innovatively design a verifiable e-counting solution based on secret sharing, combined with a cloud server provider (CSP) to effectively share the computational pressure.
3) We perform a thorough security analysis on ISE-Voting. Additionally, we design experiments to assess the proposed scheme. The results of these experiments indicate the better performance on online e-voting, in terms of system security.

This remaining paper is organized as follows. Section II describes the related work. In Section III, the system roles and entities, symbolic descriptions, and framework and goals of our proposed scheme are presented. The implementation of our proposed scheme ISE-Voting is described in detail in Section IV. Section V provides the security as well as performance analysis and experimental evaluation of our scheme. Finally, the summary is given in Section VI.

II. RELATED WORK

An e-voting system is a comprehensive cryptography-based system. The cryptographic security techniques it relies on can be generally categorized into four categories: 1) homomorphic encryption [8]; 2) digital signatures [9], [10]; 3) hybrid networks [1]; and 4) secret sharing [11], [12], and these cryptographic security techniques provide a solid foundation for the continued development of e-voting systems.

Research on e-voting systems generally involves two aspects: 1) safeguarding user privacy and 2) optimizing ballot format (BF). First, for user privacy protection, [13] proposed a verifiable online e-voting system via mix-net protocol [1], which randomizes the ciphertext through a chain of hybrid servers and recovers the plaintext ballots in an unlinkable manner. Clarkson et al. [14] proposed an e-voting scheme based on ring signatures and clash attack protection, which adds a new security model called “RE-NOTE,” and this model allows a group of users to vote without providing related information. In addition, this approach improves the security of the e-voting system using the new model. Ge et al. [15] proposed the Koinonia voting system where any user can verify that each ballot is formatted and counted correctly. Revathy et al. [17] proposed an e-voting scheme using deep learning techniques. Specifically, the scheme uses convolutional neural network (CNN) for face recognition. The voting process combines blockchain technology with a blind signature scheme, and its main goal is to evaluate the ability of online e-voting systems in guaranteeing security. Chaudhary et al. [16] proposed a voting mechanism that utilizes blockchain. The mechanism utilizes IPFS and 5G technologies to ensure that voters are able to participate in candidate elections in a cost-effective, reliable, and secure manner.

For the design and optimization of BF, [18] proposed a protocol based on ElGamal and specified verifier proofs. In this scheme, the teller proves to the voter that the submitted information about the reordering is correct by using a specified verifier proof. And each valid ballot is encrypted using a deterministic cryptographic function. Li et al. [19] proposed a blockchain-based self-recording ballot e-voting system. The scheme utilizes linkable group signatures and homomorphic time-locking puzzles to maintain anonymity, accountability, and a balance between vote size and efficiency in the e-voting system. Shahandashti and Hao [20] designed a privacy-enhancing DRE-ip thus encrypting ballots in real time. This scheme can publicly verify the results of vote counting in the voting system without decrypting the private ballots. Liu and Zhao [21] proposed a vote counting scheme based on secret sharing as well as K-anonymity, in which the votes consist of 0 and 1. It not only satisfies the basic security goals of noncheating, universal verifiability and anonymity, but also the security does not depend on any computational hardness assumptions. Huber et al. [22] designed an electronic voting system with provable security. The system is particularly suitable for election scenarios in which ballots are publicly counted but remain anonymous. By designing a completely new protocol, this scheme realizes a practical e-voting mechanism.

Taken together, the related work described above, although they all provide valuable solutions and approaches for building more reliable e-voting systems for blockchain-assisted IoT devices. However, there are still many problems in protecting user privacy in e-voting systems as well as the trustworthiness of ballot counting results. To this end, we design and implement ISE-Voting by using identity-based ring signature based on symmetric primitives and secret sharing techniques.

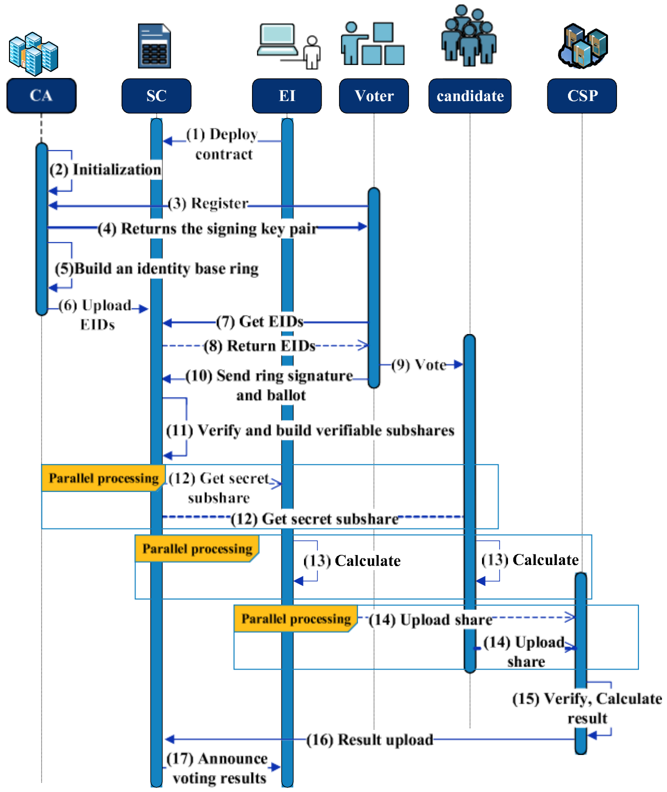


Fig. 2. Timing process for ISE-Voting.

TABLE I
DEFINITION OF SYMBOLS

Symbols	**Description**
k	Security parameter
pp	Public parameter
(Mpk, Msk)	Master public key, master private key
ID_i	V_i's identity ID, where $ID_i \epsilon \{0, 1\}^*$
S_{ID_i}	Private key for signature of V_i
c_{ID_i}	V_i's ballot
$c^j_{ID_i}$	V_i's ballot for C_j
σ_{ID_i}	Signature of V_i
$EIDs$	List of qualified voters
x_i	Statement, public Information of V_i
w_i	Witness, V_i's private information
$path_{ID_i}$	Path direction of V_i

In our scheme, identity-based ring signatures utilize symmetric primitives to streamline key management and boost data processing efficiency. This approach not only facilitates symmetric key operations but also ensures robustness against quantum attacks. Conversely, the secret sharing technique secures sensitive data by distributing it across multiple shares, thereby preserving the overall system's security even if some data is compromised. Additionally, this method promotes decentralized storage, increasing the system's fault tolerance and transparency. To sum up, it provides a viable solution for the secure implementation of modern e-voting systems, ensuring the fairness and transparency.

III. FRAMEWORK OF ISE-VOTING SYSTEM

In this section, we provide a relevant introduction to ISE-Voting's system roles and entities, the symbols in the proposed framework.

A. System Roles and Entities

In our designed scheme, which contains six main types of roles, the timing process of ISE-Voting is shown in Fig. 2.

DCA (Decentralized Registration Center): It is responsible for auditing the voter's identity information (e.g., ID, email address, etc.). If the audit passes, the *DCA* sends the corresponding signature key pairs to voters. The list of voters is publicly stored on the blockchain and can be monitored and verified by anyone.

SC (A Smart Contract on the Blockchain): It is used to assist the overall process of voting, thus automating the control and managing the execution of the voting scheme without human intervention.

Election Initiator (EI): It is responsible for creating the voting contract, setting the information, such as the topic of the vote, the list of candidates, the BF, etc. and making it public. Among them, the *BF* utilizes the Borda counting method [23] in order to realize the implementation.

V_i (The i*th Voter):* It has an identity *ID* derived from personal identity information and a unique signature key derived from the *ID*. We assume that there are a total of *n* voters in the system (where $i = 1, \dots, n$).

C_j (The j*th Candidate):* It assists the EI in the computation of the eligible ballot information and its final ballot result is c_j. We suppose there are a total of *m* candidates in the system (where $j = 1, \dots, m$).

CSP: It is used to share part of the computational tasks in the ballot counting process, thus reducing the computational burden on the candidates and the EI.

Our proposed ISE-Voting achieves decentralized role management through clear role definitions and the modular design. The EI is responsible for deploying smart contracts and managing participant registrations. Smart contracts are used to automatically execute interactions and task assignments between roles, ensuring that each participant understands the permissions and responsibilities, while also reducing the complexity of manual interventions. Additionally, the blockchain system facilitates transparent communication between roles, ensuring smooth information flow among voters, candidates, and CSPs.

B. Description of Symbols

In this section, we give the necessary description of the main notations in our proposed scheme as shown in Table I. ISE-Voting uses the security parameter *k*, the public parameter *pp*, and the master key pair (MPK, MSK) to generate the key pair (ID_i, S_{ID_i}) used for voting for the eligible voters (in fact, it is generated by a private key generator (PKG)). The voter V_i can vote for *m* candidates to generate the ballot message $c_{ID} = \{c^1_{ID_i}, \dots, c^m_{ID_i}\}$, and then sign the ballot to generate σ_{ID_i}, which is essentially constructed as a noninteractive zero-knowledge proof system, where the voter V_i utilizes the public

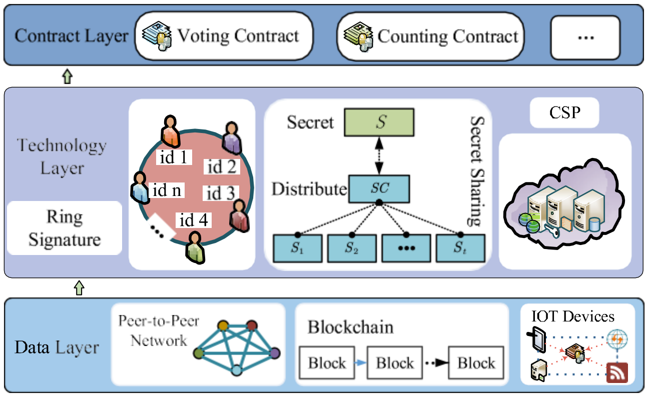


Fig. 3. Main framework of ISE-Voting.

information x_i and the private information w_i in order to prove his knowledge of the circuit C. $path_{ID_i}$ is ultimately used to achieve voter's anonymity.

C. Main Framework

The designed ISE-Voting contains a total of three layers of main framework, as shown in Fig. 3.

1) *Contract Layer:* The top layer is the contract layer, which is responsible for managing all relevant data in ISE-Voting. Voting and counting processes are conducted through smart contracts. Different types of contracts, such as voting contracts and counting contracts, can be clearly defined and managed to ensure the transparency and traceability of data processing.
2) *Technology Layer:* The middle layer is the technology layer, which includes the specific necessary cryptographic techniques to implement ISE-Voting, including ring signature, secret sharing, and CSP technologies. The ring signature ensures voter anonymity while allowing for effective identity verification. Meanwhile, the secret sharing technique divides each voter's ballot into multiple subshares, enhancing the system's security and fault tolerance.
3) *Data Layer:* The bottom layer is the data layer. As an infrastructure for data storage, IoT devices collect and process voting-related data, and some public voting information is distributed via the blockchain, allowing eligible participants to access the desired information in real time and ensuring data transparency and verifiability. The blockchain's tamper-proof nature further guarantees the security of the voting data.

In our proposed ISE-Voting, high-performance full nodes are deployed by *EIs* or blockchain service providers. These nodes are responsible for maintaining the integrity of the entire blockchain system, executing smart contracts, verifying transactions, and participating in consensus, thus ensuring the security and efficiency of the system. In contrast, general-purpose nodes can be deployed by registered voters and candidates. They primarily handle common transaction requests, store voting records, and provide data access, ensuring the transparency and verifiability of the voting process. Individual nodes in the voting system can be IoT devices (e.g., smartphones and tablets) or servers, distributed across different geographical locations. Each node transmits and interacts with secure data through encrypted communication protocols to guarantee the security and consistency of information across devices. First, the EI deploys the corresponding smart contract SC and publishes it on the blockchain, and the voters as well as the candidates obtain a corresponding permission after registering in the system. Eligible Voter V_i can vote for each candidate by using the IoT devices, depending on their personal preference, and then sign its ballot by using its own signature key through the ring signature technology in the middle layer. The EI is able to verify the validity of the signature through the smart contract SC, as well as the correctness of the BF.

If the verification is passed, the smart contract SC realizes the secret sharing of private ballots by utilizing the secret sharing technology in the middle layer, and each candidate and the EI will get a part of the secret subshare, and calculate the corresponding share, but none of them can know the real ballots or the final results of the individual candidates. Each candidate and the EI send the results of their respective calculations to the CSP for the final vote count. The CSP first verifies the correctness of the calculations of each calculation participant and informs the corresponding malicious users. If the verification is passed, then the final count is calculated and published so that everyone can verify the correctness and validity of the results.

D. Design Goals

In practical application scenarios, our proposed ISE-Voting aims to fulfill the following basic security requirements and properties.

Unforgeability: Adversary A cannot falsify an eligible ballot result. That is, no polynomial-time adversary can win the following game by a non-negligible advantage, then the ISE-Voting scheme is unforgeable. The game is played between adversary A and challenger C. We can define the wining advantage of A in the above game as: $Adv_{A}^{Forge} = Pr[A\ succeeds]$.

Anonymity: The identity of the voter and the final voting result are not available to other users in the ISE-Voting system. That is, for a given arbitrary set of identities EIDs, c_{ID}, and σ_{ID_i}, even with infinite computational capacity, no adversary can identify the true signer with a probability better than a random guess, then the scheme is unconditionally anonymous. The game is played between adversary A and challenger C. At this point, A's advantage in the above game can be defined as: $Adv_{A}^{Anon} = |Success_{A}^{Anon} - (1/n)|$.

Correctness: This property requires that the ballots of all eligible voters in ISE-Voting be counted accurately, preventing attackers from forging the process of eligible voting.

Verifiability: All users in the ISE-Voting system are able to verify the final vote results to ensure that eligible ballots have been counted correctly.

Immutability: This property is used to ensure that voting data is protected from unauthorized modification or tampering during the transmission and storage process.

Robustness: The ability of the ISE-Voting system to maintain stability and reliability despite anomalies or malicious attacks, and to ensure that the voting process runs smoothly and that the accuracy and integrity of the voting results are not compromised.

Fault Tolerance: This feature requires the system to be highly fault-tolerant to ensure that in the event of node failure or malicious attacks, the system can still maintain stable operation and ensure the accuracy and integrity of voting results.

Scalability: It implies the ability of the ISE-Voting system to handle a growing number of users and increased system load without compromising performance or risking system running. It entails maintaining efficient operation as the system expands in size, all while upholding the security and integrity of the voting data.

IV. IMPLEMENTATION OF ISE-VOTING

Our proposed ISE-Voting ensures the security of the e-voting system by applying ring signatures as well as secret sharing techniques in the blockchain systems. An identity-based ring signature based on symmetric primitives is utilized to guarantee the privacy and anonymity of the voter's identity. In addition, a new counting model based on secret sharing is designed to implement the calculation of the final ballot results.

The implementation of the ISE-Voting utilizes DS [24] algorithm and the ACC [25], [26] algorithm. Among them, the DS algorithm is a digital signature algorithm, and it generally includes three phases, *DS.KeyGen*, *DS.Sign*, and *DS.Verify*. ACC algorithm is an accumulator algorithm, it generally includes four phases, *Acc.Gen*, *Acc.Eval*, *Acc.WitGen*, and *Acc.Verify*, and the algorithm possesses correctness and collision freeness. Our scheme consists of three phases: 1) initialization and key generation phase, 2) voting phase, and 3) ballot counting and verification phase.

A. Initialization and Key Generation Phase

This phase is jointly accomplished by the EI and *DCA* through the voting contract SC. The phase specifically involves four substeps.

1) *Initialization:* The EI creates the voting contract SC, sets the system-related parameters, and specifies information, such as the list of candidates, the BF, etc., and then deploys it to the blockchain.
2) *System Parameters and Key Generation:* The *DCA* first generates the system's master public key *Mpk* and master private key *Msk* by executing the algorithm $(Mpk, Msk) \leftarrow DS.KeyGen(1^{\kappa})$. Second, it generates the public parameter *pp* by executing the algorithm $pp \leftarrow Acc.Gen(1^{\kappa})$, and then publicizes the master public key *Mpk* and the public parameter *pp*.
3) *Voter's Identity Registration:* The voter V_i adopts the ID_i derived from the personally identifiable information (PII) and then uploads it to the ISE-Voting system. *DCA* executes the algorithm $S_{ID_i} \leftarrow DS.Sign(ID_i, Msk)$ in order to generate the V_i's signature private key S_{ID_i}. The S_{ID_i} is essentially a digital signature, which is actually executed by PKG. Before the voting starts, *DCA* utilizes SC in order to form the set EIDs of qualified ID and publicize it to the blockchain, while the S_{ID_i} is kept secretly by the voter as a private key.
4) *Valid Identity Set Accumulation:* The EI executes the algorithm $(A_{\mathrm{EIDs}}, M_R) \leftarrow Acc.Eval(pp, \mathrm{EIDs})$ to accumulate the sets of identities belonging to the ring through the voting contract SC, and finally outputs the accumulator A_{EIDs} and the updated public key M_R.

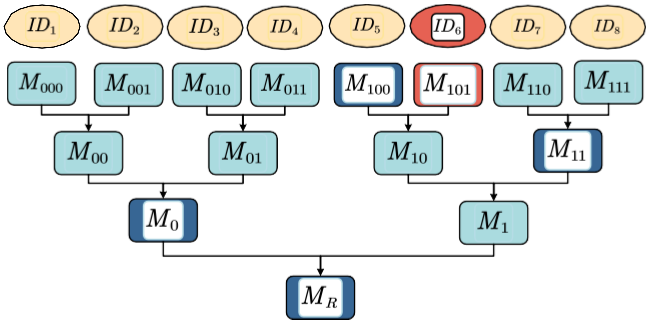


Fig. 4. Identity proof process based on Merkle Tree.

B. Voting Phase

This phase is mainly executed by the voter, specifically, the voter V_i will call the SC from the ISE-Voting system and then vote for the candidate based on the *BF* released by the EI and the individual intention. This phase contains two substeps.

1) The voter V_i executes the accumulator evaluation algorithm $Acc.Eval(pp, \mathrm{EIDs})$ by utilizing the public information to generate the parameter information: the accumulator A_{EIDs} as well as M_R.
2) The voter V_i executes the identity path generation algorithm (which is also known as the accumulator evidence generation algorithm) $path_{ID_i} \leftarrow Acc.WitGen(M_R, A_{\mathrm{EIDs}}, \mathrm{EIDs}, ID_i)$ by utilizing the public key M_R, the accumulator A_{EIDs}, the set EIDs, and an element ID_i belonging to the qualified set EIDs as inputs, and finally returns its own path direction $path_{ID_i}$ as a valid proof of identity.

Here, for the ease of description, we can assume that $n = 8$ in the ISE-Voting system, i.e., there are eight voters $V_1, \ldots, V_8$, and their respective *ID* numbers are accumulated into the Merkel accumulator as part of the identity proof through the hashing operation, and ultimately generates the root hash value M_R. For V_6, whose identity proof process is illustrated in Fig. 4, the witness w_{ID_i} (that is, $path_{ID_i}$) of the voter V_i is defined as: $w_{ID_i} = ((i_1, \ldots, i_\tau), (w_\tau, \ldots, w_1))$, where $\tau = \log n$, $i_1, \ldots, i_\tau = bin_\tau(i-1)E\{0,1\}^{\tau}$, and *bin* denotes the binary decomposition operation.

In order to further the hiding of identity of voter V_i, we use the multiplexer μ [27] in the ISE-Voting system to hide the identity of the V_i by using the path direction $path_{ID_i}$ to the root M_R. Our approach for anonymizing user identities primarily involves using disjunctive proofs to simulate the commutativity of inputs across each level of the hash function. This method allows us to obscure the precise path through the tree. Each

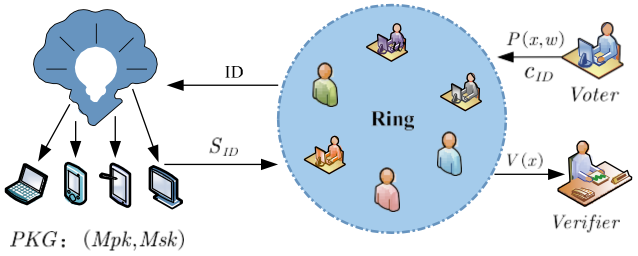


Fig. 5. ISE-Voting anonymous signature.

layer's individual statements are seamlessly integrated into an overarching junction structure. This is described in (1), where $U_\tau = H(M_{i_1,\ldots,i_f})$ and j ranges from $\tau - 1$ to 0

$$H\left(\mu\left(U_{j+1}, w_{j+1}, i_{j+1}\right)\right) = \begin{cases} H\left(U_{j+1}, w_{j+1}\right), & i_{j+1} = 0 \\ H\left(w_{j+1}, U_{j+1}\right), & i_{j+1} = 1 \end{cases} \tag{1}$$

Before the voting time deadline, the voter V_i signs the ballot c_{ID_i} by using his own signature key S_{ID_i} to generate the ring signature σ_{ID_i}, which is executed by the algorithm $sign(c_{ID_i}, \mathrm{EIDs}, ID_i, S_{ID_i}, Mpk, pp)$, and then uploads the signature data (sig_{ID_i}, c_{ID_i}) to the SC. It is worth noting here that the construction of this scheme for the identity-based ring signatures is essentially a noninteractive zero-knowledge proof system [28]. That is, $\sigma_{ID_i} = NIZK.Proof(x_i, w_i)$.

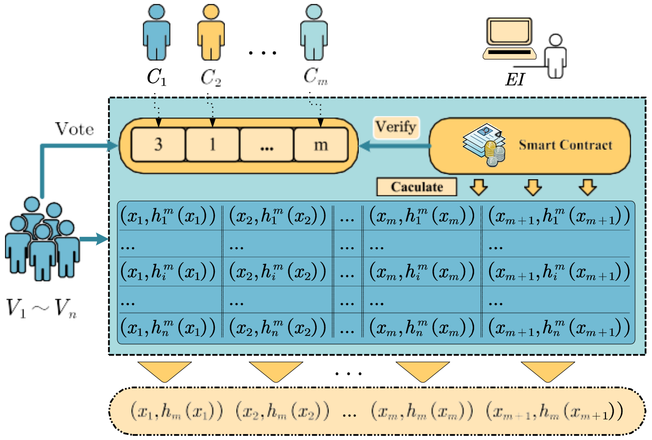
An NIZK argument generally consists of three probabilistic polynomial time (PPT) algorithms, *NIZK.Setup*, *NIZK.Prove*, and *NIZK.Verify*. In ISE-Voting, for voter V_i, it takes the statement $x_i = (c_{ID_i}, M_R, A_{\mathrm{EIDs}}, Mpk)$, and the witness $w_i = (S_{ID_i}, ID_i, path_{ID_i})$ as inputs, and outputs the argument σ_{ID_i} to prove how well V_i knows the inputs w_i of the circuit C such that the circuit satisfies $C(x_i, w_i) = y_i$, which means that the final result of $C(x_i, w_i)$ is 1. In this algorithm, using the Fiat–Shamir transform, c_{ID_i} can be embedded to generate the challenge $c_i = H(r_i, c_{ID_i})$, where r_i is a random value. The details of the process are shown in Fig. 5. For an adversary experiment $Adv^{zk}_{\mathcal{A},\mathrm{NIZK}}(k)$, it has negligible advantage

$$\begin{aligned} Adv^{zk}_{\mathcal{A},\mathrm{NIZK}}(k) = \Big| &\Pr\left[crs \leftarrow \mathrm{NIZK.Setup}\left(1^k\right) : \mathcal{A}^{\mathrm{NIZK.Prove}} = 1 \right] \\ &- \Pr\left[\left(crs^*, \sigma^*\right) \leftarrow \mathrm{NIZK.Sim}\left(1^k, x_i\right) : \mathcal{A}^{(x_i, crs^*, \sigma^*)} = 1 \right] \Big| \\ &\le \mathrm{negl}(k) \end{aligned}$$

where $(crs^*, \sigma^*_{ID_i}) \leftarrow \mathrm{NIZK.Sim}(1^k, x_i)$ is a simulator that takes the security parameter k and statement x as input, and outputs the common reference string $crs*$ and the simulation proof $\sigma^*_{ID_i}$. Then, it means that the NIZK argument possesses zero-knowledge. If there exist algorithms S, *NIZK.Sim* and extractors E that satisfy the definition of zero-knowledge, then the proof system NIZK satisfies simulation extractability such that

$$\begin{aligned} Adv^{SimE}_{\mathcal{A},NIZK}(k) = \Pr\Big[&\left(x_i, \sigma_{ID_i}\right) \leftarrow \mathcal{A}^{S,\mathrm{NIZK.Sim}}\left(1^k\right) \\ &w_i \leftarrow \mathrm{NIZK.Ext}\left(crs, t, x_i, \sigma_{ID_i}\right) : \mathrm{NIZK.Verify}\left(x_i, \sigma_{ID_i}\right) \\ &= 1 \wedge \left(x_i, \sigma_{ID_i}\right) \notin M \wedge (x_i, w_i) \notin R \Big] \le negl(k) \end{aligned}$$

where $E = ((crs, t) \leftarrow \mathrm{NIZK.ExtGen}(1^k, t), w_i \leftarrow$ $NIZK.Ext(crs, t, x_i, \sigma_{ID_i}))$ is the extractor, $\sigma_{ID_i} \leftarrow S(t, x_i)$, t is a state, and M is the list of queries made by $\mathcal{A}$ to *NIZK.Sim.*



$(x_1, h_1^m(x_1))$	$(x_2, h_1^m(x_2))$	...	$(x_m, h_1^m(x_m))$	$(x_{m+1}, h_1^m(x_{m+1}))$
...	...	...	...	...
$(x_1, h_i^m(x_1))$	$(x_2, h_i^m(x_2))$	...	$(x_m, h_i^m(x_m))$	$(x_{m+1}, h_i^m(x_{m+1}))$
...	...	...	...	...
$(x_1, h_n^m(x_1))$	$(x_2, h_n^m(x_2))$	...	$(x_m, h_n^m(x_m))$	$(x_{m+1}, h_n^m(x_{m+1}))$

Fig. 6. Proposed ballot counting scheme.

Algorithm 1 Ballot Cutting Algorithm

```
Input: $(m, n), \{c^j_{ID_i}\}_{i\in[n], j\in[m]}$;
Output: $(x^j_i, h^m_i(x^j_i))_{i\in[n], j\in[m+2]}$;
for $i \leftarrow 1$ to $n$ do
    $a^i_{1,1}$ = random value in $Z_q$;
    for $j \leftarrow 1$ to $m$ do
        $h^j_i(x) = \sum_{t=1}^{j} a^i_{j,t} \cdot x^t + c^j_{ID_i}$;
        if $(j == 1)$ then
            $x_j$ = random value();
        end if
        if $(j == m)$ then
            $x_{j+2}$ = random value();
        end if
        $x_{j+1}$ = random value();
        for $t \leftarrow 1$ to $j + 1$ do
            $a^i_{j+1,t} = h^j_i(x_t)$;
            if $(j == m)$ then
                $a^i_{j+1,t+1} = h^j_i(x_{t+1})$;
            end if
        end for
    end for
end for
```

For a given binary relation $R : \{0, 1\}^* \times \{0, 1\}^* \longrightarrow \{0, 1\}$, V_i needs to satisfy two conditions in order to make it establish that $(x_i, w_i)ER$ as follows.

1) *Proof the Identity Belongs to the Set (ID_iEEIDs):* That is, $Acc.Verify(M_R, A_{\mathrm{EIDs}}, path_{ID_i}, ID_i) = 1$. The algorithm takes the public key M_R, the accumulator A_{EIDs}, the witness w_{ID_i}, and the voter's identity ID_i as inputs and finally outputs the verification result.
2) *Proof of the Validity of S_{ID_i} :* That is, $DS.Verify(ID_i, S_{ID_i}, Mpk) = 1$. The algorithm takes the message ID_i which has been signed by the voter, the master public key Mpk, and the signature private key S_{ID_i} as input and finally outputs the verification result.

C. Ballot Counting and Verification Phase

This phase is a common phase for all users in the system, and it contains two subphases: 1) the ballot counting subphase and 2) the verification subphase. For the former subphase, when the voting time ended, the voter will no longer be able to vote through the system. The ISE-Voting system will verify the validity of the uploaded signatures as well as the legitimacy of the ballots through the contract SC. Further, the subphase includes two steps as follows.

1) The contract SC obtains (A_{EIDs}, M_R) through the accumulator algorithm $Acc.Eval(pp, \text{EIDs})$.
2) The contract SC verifies the validity of the signature by using the information obtained above and the returned value of the ring signature verification algorithm $NIZK.verify((c_{ID_i}, M_R, A_{\text{EIDs}}, Mpk), \sigma_{ID_i})$.

If the verification fails (returns 0), the system will report the possible dishonest behavior of the corresponding malicious voter. If the verification passes (returns 1), the contract SC will collect the qualified ballots for the next computation, and the secret subshares will be distributed by the SC to each candidate C_j (where $j = 1, \ldots, m$) and the EI, and then C_j (where $j = 1, \ldots, m$) and EI, respectively, sum up the secret subshares. For the latter subphase, when the CSP calculates the final ballot result based on the secret summation value, all users in the system can verify the validity of the result.

The Ballot Counting Subphase In the ballot counting subphase, the contract SC in the ISE-Voting system will use qualified ballots for secret sharing, which can be divided into five substages: 1) ballot cutting stage; 2) ballot subshare sharing stage; 3) verification message broadcasting stage; 4) ballot share verification stage; and 5) ballot reconstruction stage. The proposed ballot counting scheme is shown in Fig. 6.

1) The ballot cutting stage is executed by the contract SC in an automated mode. When the BF of voter V_i (where $i = 1, \ldots, n$) is reviewed and approved, the contract SC will secretly cut the ballot $c_{ID_i} = \{c^1_{ID_i}, c^2_{ID_i}, \ldots, c^m_{ID_i}\}$ of each voter. First, the large prime numbers p and q are selected such that $q|(p-1)$, and the function $h : Z_q \to Z_p$ is selected. The execution process contains a total of m rounds, and j is the current execution round. The algorithm is described as shown in Algorithm 1, and the specific execution flow is as follows.
   a) When $j = 1$, the contract SC randomly selects an element $a^i_{1,1}$ in the region Z_q and then utilizes this element to construct the polynomial $h^1_i(x) = a^i_{1,1} \cdot x + c^1_{ID_i}$. Then, two points x_1 and x_2 are randomly selected and substituted to get: $(x_1, h^1_i(x_1))$, $(x_2, h^1_i(x_2))$. The result is then submitted to the next round of coefficient assignment: $a^i_{2,1} = h^1_i(x_1)$ and $a^i_{2,2} = h^1_i(x_2)$, and the constructed polynomial is destroyed.
   b) When $j = 2, \ldots, m-1$, the polynomial coefficients generated in the previous round by the contract SC computation are utilized in order to construct the polynomial $h^j_i(x) = a^i_{j,1} \cdot x + \cdots + a^i_{j,j} \cdot x^j + c^j_{ID_i}$. Then, combine $x_1, \ldots, x_j$ and randomly select one point x_{j+1} in the region to substitute into $h^j_i(x)$ to obtain: $(x_1, h^j_i(x_1)), \ldots, (x_{j+1}, h^j_i(x_{j+1}))$. The result is then submitted to the next round of coefficient assignment: $a^i_{j+1,1} = h^j_i(x_1), \ldots, a^i_{j+1,j+1} = h^j_i(x_{j+1})$, and the constructed polynomial is destroyed.
   c) When $j = m$, which is the final round of ballot cutting, the polynomial coefficients generated from the contract SC computation in round $m-1$ are used to construct the polynomial $h^m_i(x) = a^i_{m,1} \cdot x + a^i_{m,2} \cdot x^2 + \cdots + a^i_{m,m} \cdot x^m + c^m_{ID_i}$. Then, combine $x_1, \ldots, x_m$ and randomly select two points x_{m+1} and x_{m+2} in the region to substitute into $h^m_i(x)$ to obtain the final secret subshares: $(x_1, h^m_i(x_1)), \ldots, (x_{m+2}, h^m_i(x_{m+2}))$ and destroy the constructed polynomial.
2) In the ballot subshare sharing stage, the SC will share the subshares of the subballots, and each candidate C_j (where $j = 1, \ldots, m$) as well as EI will receive the secret shared subshares individually, without knowing the real ballot information. In particular, C_j will receive the ballot subshare $(x_j, h^m_1(x_j)), \ldots, (x_j, h^m_n(x_j))$, EI will obtain ballot subshares $(x_{m+1}, h^m_1(x_{m+1})), \ldots, (x_{m+1}, h^m_n(x_{m+1}))$, and V_i (where $i = 1, \ldots, n$) will receive the secret information $(a^i_{1,1}, x_{m+2}, h^m_i(x_{m+2}))$. In addition, after obtaining the ballot subshares, C_j (where $j = 1, \ldots, m$) and EI will separately calculate the summation of the ballot subshares. Particularly, they will calculate: $h_m(x_j) = \sum_{i=1}^{n} h^m_i(x_j)$ (where $j = 1, \ldots, m, m+1$) individually. The summation results will then be sent to the CSP separately.
3) In the verification information broadcasting stage, the SC will broadcast and announce some information which will be used for users to perform verification at a later stage. Specifically, the SC will use the value point set $x_1, \ldots, x_m$ and the validation information ξ_j (where $j = 0, \ldots, m$) to broadcast and publish to the blockchain. Here, $\xi_0 = g^{\sum_{i=1}^{n} c^m_{ID_i}} \bmod p$, and $\xi_j = g^{\sum_{i=1}^{n} a^i_{m,j}} \bmod p$ for $j = 1, \ldots, m$.
4) The ballot share verification stage is performed by the CSP, it verifies the validity of the received $m+1$ ballot shares by (2), where r ranges from 1 to $m+1$. If the verification passes, it goes to the next stage of the vote counting

$$g^{h_m(x_r) \bmod q} \bmod p == \prod_{j=0}^{m} \left(\xi_j\right)^{(x_r)^j} \bmod p. \qquad (2)$$

5) The ballot reconstruction stage is also performed by the CSP, which reconstructs the ballot shares via SC. Specifically, for a given $m+1$ secret shares, the CSP reconstructs the results by using the Lagrange interpolation algorithm as shown in (3) and (4), where $j = m, m-1, \ldots, 1$

$$h_j(x) = \sum_{k=1}^{j+1} h(x_k) \prod_{t=1, t \neq k}^{j+1} \left(\frac{x - x_t}{x_k - x_t} \right) \qquad (3)$$

$$C_j = h_j(0) = \sum_{k=1}^{j+1} h(x_k) \prod_{t=1, t \neq k}^{j+1} \left(\frac{-x_t}{x_k - x_t} \right). \qquad (4)$$

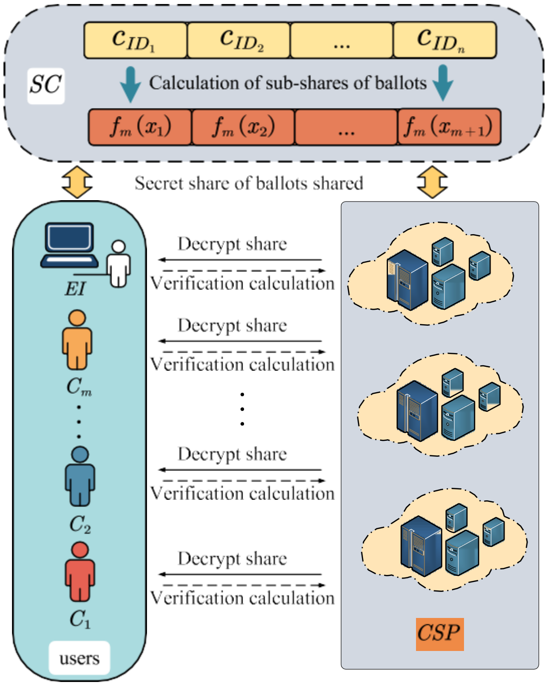


Fig. 7. Procedure of the user verification subphase.

When $j = m$, we can recover the polynomial $h_m(x)$ from $m + 1$ ballot shares, where the value of $h_m(0)$ is the final C_m's ballot result c_m, when x is 0. At this point, the coefficients of the polynomials $\{a_{m,1}, a_{m,2}, \ldots, a_{m,m}\}$ are re-executed as the output values of $h_{m-1}(x)$ with the Lagrange interpolation algorithm, and finally recover $h_{m-1}(x)$, the obtained coefficients $\{a_{m-1,1}, a_{m-1,2}, \ldots, a_{m-1,m-1}\}$ and the ballot result value c_{m-1} of C_{m-1}. Repeat the above operation until $j = 1$. CSP can finally recover the polynomial $h_1(x)$, and when x is 0, the value of $h_1(0)$ is the final ballot result value c_1 of C_1. Through m rounds of iterative execution, CSP can obtain the final ballot result of C_1 C_m: $c_m, c_{m-1}, \ldots, c_1$, and the calculated final ballot result is uploaded to SC, which publishes the final ballot result to the blockchain.

User Verification Subphase When the CSP publishes the calculated final ballot results to the blockchain via SC, all users in the system can see the final ballot results, and procedure of the user verification subphase is shown in Fig. 7. All users in the system can verify the correctness of the ballot results.

First, V_i needs to publish his qualification proof $a^i_{1,1}$ to the blockchain, and then work with the remaining voters V_j (where $j = 1, 2, \ldots, n$ and $j \,/\!= i$) to jointly compute the value of the polynomial $\hbar(x_{m+2})$. V_i constructs $\dot{h}(x_{m+2})$ by using the ballot results $c_m, c_{m-1}, \ldots, c_1$ published by SC, executing steps referenced to Algorithm 3, where $c^j_{ID_i}$ is replaced by c_j and $a^i_{1,1}$ is replaced by $\sum_{i=1}^{n} a^i_{1,1}$. The broadcast verification message is then used in conjunction with (5) to prove that the CSP computes the final ballot results correctly

$$g^{h_m(x_{m+2}) \bmod q} \bmod p \overset{?}{=} \prod_{j=0}^{m} \left(a_j^{(x_{m+2})^j} \right) \bmod p. \tag{5}$$

If the equation holds after verification through (5), it means that the CSP has truthfully carried out the calculation of the final result. Otherwise, it will be notified of the existence of malicious behavior and punished accordingly.

For the EI, it can see the real-time information of the voting and can verify the final ballot results of each candidate to determine whether the CSP has conducted the calculation truthfully. For each candidate $C_j(jE[1,m])$, they can share the calculated ballot shares $(x_1, h_m(x_1)), (x_2, h_m(x_2)), \ldots, (x_{m+1}, h_m(x_{m+1}))$ to work out the final ballot results in collaboration with other candidates, and the algorithm is executed as shown in (3) and (4). If the result calculated by candidate C_j is inconsistent with the announced result, candidate C_j first verifies the authenticity of the ballot shares shared by each other candidate C_1 C_m (excluding C_j) through the verification information broadcast on the blockchain. The specific verification can be executed through (2), and then the C_j informs the corresponding dishonest behaviors and imposes the corresponding penalties. If all the verifications are correct, then the malicious behavior of corresponding CSP node is notified to the whole system.

TABLE II
SECURITY COMPARISON OF BLOCKCHAIN-BASED E-VOTING SCHEMES

Properties	References				
	S-Voting	BC-Voting	D-bame	HM-Voting	**Ours**
Unforgeability	✓	✓	✓	×	✓
Anonymity	×	×	✓	×	✓
Correctness	✓	✓	✓	✓	✓
Verifiability	✓	×	×	×	✓
Immutability	✓	✓	✓	✓	✓
Robustness	×	×	×	✓	✓
Faut tolerance	✓	×	✓	✓	✓
Scalability	×	×	✓	✓	✓

V. SECURITY AND PERFORMANCE ANALYSIS

A. Security Analysis

In this section, we will analyze potential attacks and misbehavior and present how ISE-Voting fights against them in detail.

In addition, we provide a security comparison of blockchain-based e-voting schemes, as shown in Table II. The tested e-voting schemes, include S-Voting [29], BC-Voting [30], D-bame [31], and HM-Voting [32].

1) *Unforgeability:* Suppose that event T_γ means adversary $\mathcal{A}$ wins the game γ and generates forgery $(c^*_{ID}, \mathrm{EIDs}^*, \sigma^*_{ID})$. For the case where the voter's ID belongs to EIDs, there are four possible cases involved in signing the ballot c_{ID}:

Event T_1: $\mathcal{A}$'s forgery successfully passes verification, $Adv^{Forge}_{\mathcal{A}} = Pr[\mathcal{A}\ succeeds]$, i.e., $Adv^{Forge}_{\mathcal{A}}(k) = 1$.

Event T'_1: If event T_1 occurs, through the simulated extractability feature of the NIZK protocol, the statement $x = (c^*_{ID}, U_R{}^*, A^*_{\mathrm{EIDs}}, Mpk)$ will extracts the corresponding knowledge w, ensuring that $((c^*_{ID}, M_R{}^*, A^*_{\mathrm{EIDs}}, Mpk), (S_{ID}{}^*, ID^*, path^*_{ID})) \in R$ is fulfilled. We have $Pr[T'_1] = Pr[T_1] - negl(k)$. This event can be divided into two disjoint subevents $T'_{1.1}$ and $T'_{1.2}$: $T_{1.1}$: $ID^* \in EIDs^*$: Due to the fact that DS realized EU-CMA security, we can conclude that $Pr[T_{1.1}] \leq \mathrm{Adv}^{\text{EU-CMA}}_{\mathcal{A}} < \mathrm{negl}(k)$.

$T'_{1.2}$: $ID^* \notin EIDs^*$: In this case, the extractor running on the forgery of $\mathcal{A}$ generates a valid witness (w^*_{ID}) for the extracted identity (ID^*) not included in the ring. It also generates the auxiliary information (A^*_{EIDs}). That is, $(A^*_{\text{EIDs}}, M^*_R) = Acc.Eval(pp, \text{EIDs})$, but $Acc.Verify(M^*_R, A^*_{\text{EIDs}}, w^*_{ID}, ID^*) = 1$. So if this event occurs the collision freeness property of ACC is destroyed. So we can conclude that $Pr[T'_{1.2}] < negl(k)$. Therefore, $Pr[T'_1] = Pr[T'_{1.1}] + Pr[T'_{1.2}] < negl(k)$. So we have $Pr[T_1] < negl(k)$. i.e., $Pr[Adv^{Forge}_{\mathcal{A}}(k) = 1] = Pr[T_1] < negl(k)$.

2) *Anonymity:* The anonymity of ISE-Voting is achieved through the zero-knowledge property of NIZK based on MPC-in-the-Head. For the previous property, we use a game-based approach to show that ISE-Voting is capable of voting anonymity, considering the event E_τ in which adversary $\mathcal{A}$ wins in $GAME_\tau$:

$GAME_1$: Adversary $\mathcal{A}$ runs $Adv^{Anon}_{\mathcal{A}}$.

$GAME_2$: Same game as the previous one, but the proof π (note that $\pi = \sigma_{ID}$) generated using NIZK on circuit C is replaced by the output of its simulator $NIZK.Sim$. This is computationally indistinguishable from the previous game due to the zero-knowledge nature of NIZK. Therefore, we can conclude that $|Pr[T_2] - Pr[T_1]| = Adv^{zk}_{\mathcal{A},NIZK} < negl(k)$.

3) *Correctness:* In our design of ISE-Voting, the blockchain system is used as a database to store various data generated during the e-voting process. The $V_i(i \in [1, n])$ calculates the value of the polynomial $h^m_i(x_{m+2})$ by using the set of points $x_1, \ldots, x_m$ in conjunction with the system-provided privacy data in the ballot cutting stage, and then compares the result of the calculation with the system-provided polynomial value $h^m_i(x_{m+2})$. If they are consistent, then this means that the computation process was performed truthfully.

4) *Verifiability:* As we introduced in the user verification stage, all users in the system can verify the correctness of the final reconstructed ballot results. The V_i publishes proof $a^i_{1,1}$ to the blockchain and then calculates the value of the polynomial $\bar{h}(x_{m+2})$ in conjunction with the other voters, and then combines the on-chain information with (5) in order to verify the final ballot results. The EI and the individual candidates can work through the subshare of the secret ballots in order to verify the correctness of the result.

5) *Immutability:* In our scheme, data information is publicly stored on the blockchain, which makes it impossible for any malicious attacker V^* to utilize adversary information for valid signatures, thus it enables the voters to monitor the potential malicious behavior of the EV. Additionally, a complete ballot can only be restored if all "counters" are honest and cooperative. Malicious behavior by any individual "counter" will be detected and tracked.

6) *Robustness:* After voters submit their ballots through the ISE-Voting, the system filters out abnormal data through a strict identity and ballot verification process, ensuring the validity and reliability of the input data. Additionally, the ballot data is stored across multiple network nodes in the blockchain system, where each node operates independently and is unaffected by others. This reduces the impact of individual node failures or abnormal data on the overall counting results, thereby enhancing the system's healthy. Furthermore, once the ballots and recorded information are added to the blockchain, they cannot be modified or deleted. This feature prevents data tampering and improper manipulation, further strengthening the stability and reliability of ISE-Voting in uncertain environments.

7) *Fault Tolerance:* The ISE-Voting ensures fault tolerance through multinode backups and distributed storage on the blockchain. As mentioned before, the ballot shares are divided into multiple subshares [$h_m(x_j) = \sum_{i=1}^{n} h^m_i(x_j)$ (where $j = 1, \ldots, m, m+1$)] and are stored separately in different counting nodes. This way, even if some nodes fail or are attacked, the system can still recover complete information from the remaining nodes. Furthermore, even if malicious nodes obtain the secret shares, they cannot forge the ballots. This guarantees the security and integrity of the ballot's secret shares, ensuring the final results as well as the fault tolerance of the system.

8) *Scalability:* In our design of ISE-Voting, the memory usage of voter signatures grows logarithmically with the total number of voters, ensuring high efficiency and flexibility. Additionally, the dispersion of ballot subshares $[(x_j, h^m_1(x_j)), \ldots, (x_j, h^m_n(x_j))](j \in [1, m+1])$ across $m+1$ nodes effectively distributes the computational load of the system, enhancing its concurrent processing capability. As a result, the system can accommodate a large number of concurrent voters and ballots while maintaining stable and efficient operation, even as the user scale continues to grow.

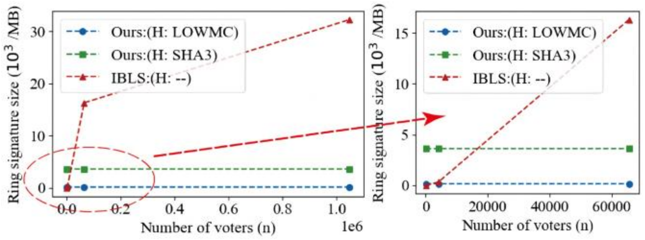


Fig. 8. Comparison of identity-based ring signature sizes.

B. Performance Analysis

In our experiments, we set the number of voters n ranging from 2^6 to 2^{20}. As shown in Fig. 8, where 1e6 is 10^6. In our scheme, ISE-Voting derives its security from the collision resistance and one-way attribute of the hash function H. These hash functions have the optimized complexity and only require the assumption of the existence of an one-way function, which reduces the overall size of the proof circuits C and the signatures. Additionally, the security of the anonymous signatures in ISE-Voting is based entirely on symmetric key

TABLE III
COMPARISON OF SIGNATURE EFFICIENCY AND SECURITY

Schemes	Cryptography	$\lvert S_{ID} \rvert$	$\lvert \sigma \rvert \, (MB) \, (Asympt.)$	Assumptions	Quantum-Resistant
UIBS	Identity-Based	160 bit	$\mathcal{O}(n)$	DsjSDH	×
IBLS	Identity-Based	600 MB	$\mathcal{O}(n)$	Lattice	✓
TLIBS	Identity-Based	$n \cdot \gamma^2$ bit	$\mathcal{O}(n)$	Lattice	×
Ours	Identity-Based	167 KB	$\mathcal{O}(\log n)$	Symmetric	✓

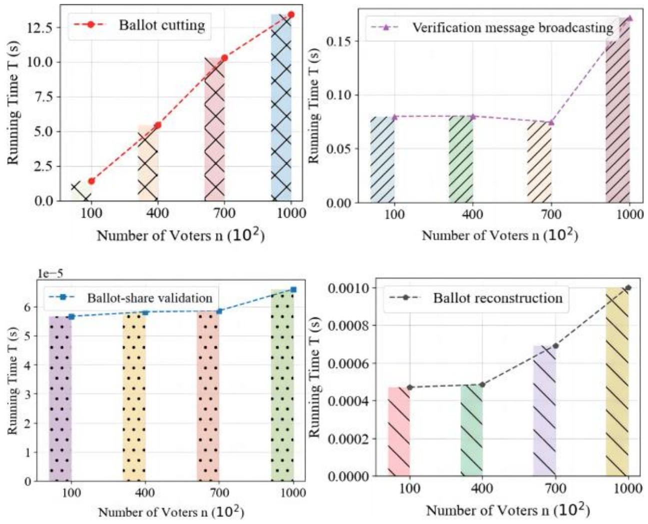


Fig. 9. When $m = 2$, the running time cost of each stage of the counting subphase.

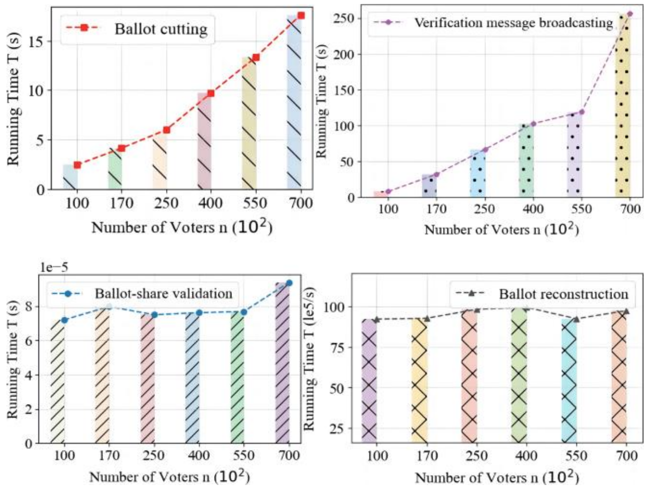


Fig. 10. When $m = 3$, the running time cost of each stage of the counting subphase.

operations, making the scheme resistant to quantum attacks. We choose two different hash functions: 1) the cryptographic hash function SHA-3 and 2) the block cipher LOWMC based on the substitution-permutation network (SPN) structure for specific analysis. Specifically, when the numbers of voters n are 2^6, 2^{12}, and 2^{20}, and the underlying hash functions is LOWMC, the sizes of the identity-based ring signatures of our scheme are 169.902, 170.145, and 170.645 MB, respectively. Meanwhile, when the underlying hash function is SHA-3, the sizes of the identity-based ring signatures of our scheme are 3618, 3622, and 3627 MB, respectively. Compared to the secure IBLS scheme [34], which has ring signature sizes of 5, 335, and 32 243 MB, respectively. Our proposed ISE-Voting shows that the cost of signatures increases in a nearly horizontal manner with the increase in the number of voters.

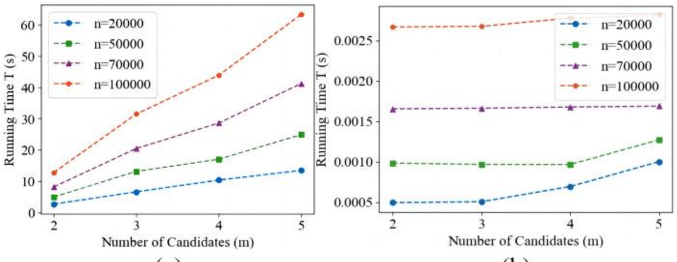
We consider aspects of signing efficiency as well as security. The evaluation is made at $k = 128$ bit post-quantum security level, and the results are shown in Table III. UIBS [33], IBLS [34], and TLIBS [35] are identity-based ring signature schemes. The signing key in the UIBS scheme is only 160 bit and it does not provide post-quantum security. The signature key in the IBLS scheme is 600 MB and it has post-quantum security. The size of the signing key in the TLIBS scheme depends on the size of the ring set n and the security parameter γ. Therefore, according to the table, ISE-Voting has better performance in terms of signature size.

In addition, in order to more comprehensively evaluate and analyze the time overhead of ISE-Voting in the stages of the ballot counting subphase, we conduct experiments on a Lenovo laptop computer by using the Python language. The laptop was configured with an Intel Core i5 CPU i5-13500 h at 2.6 GHz and 16 GB of RAM. There are five stages of the vote counting subphase (i.e., the ballot cutting stage, the ballot subshare sharing stage, the verification message broadcasting stage, the ballot share verification stage, and the ballot reconstruction stage). Among them, the four stages except the second one occupy the major time overhead of our scheme. Therefore, we analyze them in detail.

Fig. 9 shows running time of the four stages when the number of candidates and the number of voters ranges from 10 000 to 100 000. It is worth noting here that when the number of voters reaches 100 000, the ballot cutting stage takes about 13.4 s, the verification message broadcasting stage takes about 0.171 s, the ballot share verification stage takes about 0.06 ms, and the ballot reconstruction stage takes about 0.001 s.

Fig. 10 gives the running time of four stages when the number of candidates m = 3 and the number of voters ranges from 10 000 to 70 000. Note that, when the number of voters reaches 70 000, the ballot cutting stage takes about 16.98 s, the verification message broadcasting stage takes about 1.06 s, the ballot share verification stage takes about 0.093 ms, and the ballot reconstruction stage takes about 0.96 ms.

The ballot cutting stage and reconstruction stage are two of the more important stages in the vote counting subphase, and they are directly related to the runtime of the entire vote counting subphase. Fig. 11 gives a comparison of the running time when the numbers of candidates are 2–5, and the number of voters is between 20 000 and 100 000, respectively. According to Fig. 11, we can know that when the number of candidates reaches 5 and the number of voters is 20 000, the



(a) (b)

Fig. 11. Performance relationships between numbers of participant and voting time. (a) Ballot cutting. (b) Ballot reconstruction.

ballot cutting stage takes 12.68 s, and the ballot reconstruction stage takes 2.6 ms. When the number of voters is 100 000, the ballot cutting stage takes 63.32 s, and the ballot reconstruction stage takes 2.81 ms.

By comprehensively analyzing the above data, we can conclude that ISE-Voting outperforms other methods in security and shows good efficiency in both the voting phase and the counting subphase. It is proven that ISE-Voting is well-suited for a broad range of voting requirement scenarios on IoT devices and provides a reliable solution.

VI. Conclusion and Future Work

In this article, we proposed a blockchain-based e-voting system, ISE-Voting, which provides users with a more secure, transparent and efficient voting experience. ISE-Voting utilizes two algorithms, namely the zero-knowledge proof algorithm based on MPC-in-the-Head and the accumulator algorithm, to implement an identity-based ring signature. Additionally, a ballot cutting method based on secret sharing is adopted in ISE-Voting. The necessary theoretical analysis and experiments are conducted to evaluate the security and performance of ISE-Voting, and the experimental showed ISE-Voting has better performance with high security. Identity-based ring signatures with symmetric primitives simplify key management and enhances data processing performance. However, our approach still can be improved. For instance, it does not address the issue of voter authentication using strong mechanisms like biometrics. Additionally, although secret sharing technique enhances data security, it relies on the collaboration of all participants. Our implemented ballot counting algorithm is currently more suited for scenarios where voters are in the majority and candidates are in the minority. However, as the number of candidates increases, the system's efficiency may be somewhat compromised. Hence, further optimization of algorithm efficiency and rigorous management of associated security risks are needed in practical deployments. The ISE-Voting leverages existing blockchain systems and cryptographic platforms, and combines a flexible user interface design which enables various stakeholders to interact easily. This ultimately provides an efficient and secure e-voting system in real-world applications. In the future, we plan to further improve the speed of ISE-Voting's secure computation, as well as adapt ISE-Voting to real-world e-voting scenarios for IoT devices.

References

[1] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–88, 1981.

[2] G. B. Mermer, E. Zeydan, and S. S. Arslan, "An overview of blockchain technologies: Principles, opportunities, and challenges," in *Proc. 26th Signal Process. Commun. Appl. Conf. (SIU)*, 2018, pp. 1–4.

[3] D. Das, S. Banerjee, P. Chatterjee, U. Ghosh, and U. Biswas, "Blockchain for intelligent transportation systems: Applications, challenges, and opportunities," *IEEE Internet Things J.*, vol. 10, no. 21, pp. 18961–18970, Nov. 2023.

[4] C. Chi, Z. Yin, Y. Liu, and S. Chai, "A trusted cloud-edge decision architecture based on blockchain and MLP for AIoT," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 201–216, Jan. 2024.

[5] H. Bao et al., "A probabilistic and distributed validation framework based on blockchain for artificial intelligence of things," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 17–28, Jan. 2024.

[6] T. Saleem et al., "ProofChain: An X.509-compatible blockchain-based PKI framework with decentralized trust," *Comput. Netw.*, vol. 213, Aug. 2022, Art. no. 109069.

[7] N. Vashistha, M. M. Hossain, M. R. Shahriar, F. Farahmandi, F. Rahman, and M. M. Tehranipoor, "eChain: A blockchain-enabled ecosystem for electronic device authenticity verification," *IEEE Trans. Consum. Electron.*, vol. 68, no. 1, pp. 23–37, Feb. 2022.

[8] A. Kiayias and M. Yung, "Self-tallying elections and perfect ballot secrecy," in *Proc. Int. Conf. Theory Pract. Public Key Cryptogr. (PKC)*, 2002, pp. 141–158.

[9] S. Singh, N. K. Rajput, V. K. Rathi, H. M. Pandey, A. K. Jaiswal, and P. Tiwari, "Securing blockchain transactions using quantum teleportation and quantum digital signature," *Neural Process. Lett.*, vol. 55, pp. 3827–3842, Aug. 2023.

[10] K.-A. Shim, "On the suitability of post-quantum signature schemes for Internet of Things," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 10648–10665, Mar. 2024.

[11] J. C. Benaloh, "Secret sharing homomorphisms: Keeping shares of a secret sharing," in *Proc. Conf. Theory Appl. Cryptogr. Tech.*, 1986, pp. 251–260.

[12] E. Zhang, J. Peng, and M. Li, "Outsourcing secret sharing scheme based on homomorphism encryption," *IET Inf. Secur.*, vol. 12, no. 1, pp. 94–99, 2018.

[13] D. Chaum, "Secret-ballot receipts: True voter-verifiable elections," *IEEE Security Privacy*, vol. 2, no. 1, pp. 38–47, Jan./Feb. 2004.

[14] M. R. Clarkson, S. Chong, and A. C. Myers, "Civitas: Toward a secure voting system," in *Proc. IEEE Symp. Security Privacy (SP)*, 2008, pp. 354–368.

[15] H. Ge et al., "Koinonia: Verifiable e-voting with long-term privacy," in *Proc. Annu. Comput. Secur. Appl. Conf. (ACSAC)*, 2019, pp. 270–285.

[16] S. Chaudhary et al., "Blockchain-based secure voting mechanism underlying 5G network: A smart contract approach," *IEEE Access*, vol. 11, pp. 76537–76550, 2023.

[17] G. Revathy, K. B. Raj, A. Kumar, S. Adibatti, P. Dahiya, and T. M. Latha, "Investigation of E-voting system using face recognition using convolutional neural network (CNN)," *Theor. Comput. Sci.*, vol. 925, pp. 61–67, Aug. 2022.

[18] M. Hirt and K. Sako, "Efficient receipt-free voting based on homomorphic encryption," in *Proc. Int. Conf. Theory Appl. Cryptogr. Tech. (EUROCRYPT)*, 2000, pp. 539–556.

[19] H. Li, Y. Li, Y. Yu, B. Wang, and K. Chen, "A blockchain-based traceable self-tallying E-voting protocol in AI era," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1019–1032, Apr.–Jun. 2021.

[20] S. F. Shahandashti and F. Hao, "DRE-ip: A verifiable E-voting scheme without tallying authorities," in *Proc. 21st Eur. Symp. Res. Comput. Security (ESORICS)*, 2016, pp. 223–240.

[21] Y. Liu and Q. Zhao, "E-voting scheme using secret sharing and K-anonymity," *World Wide Web*, vol. 22, no. 4, pp. 1657–1667, 2019.

[22] N. Huber et al., "Kryvos: Publicly tally-hiding verifiable e-voting," in *Proc. 29th ACM Conf. Comput. Commun. Secur. (CCS)*, 2022, pp. 1443–1457.

[23] P. Emerson, "The original Borda count and partial voting," *Soc. Choice Welfare*, vol. 40, no. 2, pp. 353–358, 2013.

[24] J. Maire, and D. Vergnaud, "Efficient zero-knowledge arguments and digital signatures via sharing conversion in the head," in *Proc. 28th Eur. Symp. Res. Comput. Secur. (ESORICS)*, 2023, pp. 435–454.

[25] P. Camacho, A. Hevia, M. Kiwi, and R. Opazo, "Strong accumulators from collision-resistant hashing," in *Proc. 11th Inf. Secur. Conf. (ISC)*, 2008, pp. 471–486.

[26] D. Boneh, S. Eskandarian, and B. Fisch, “Post-quantum EPID signatures from symmetric primitives,” in *Proc. Cryptogr. Track RSA Conf.*, 2019, pp. 251–271.

[27] D. Derler, S. Ramacher, and D. Slamanig, “Post-quantum zero-knowledge proofs for accumulators with applications to ring signatures from symmetric-key primitives,” in *Proc. Int. Conf. Post-Quantum Cryptogr.*, 2018, pp. 419–440.

[28] J. Katz, V. Kolesnikov, and X. Wang, “Improved noninteractive zero knowledge with applications to post-quantum signatures,” in *Proc. 25th ACM Conf. Comput. Commun. Secur. (CCS)*, 2018, pp. 525–537.

[29] E. Yavuz, A. K. Koç, U. C. Çabuk, and G. Dalkılıç, “Toward secure e-voting using ethereum blockchain,” in *Proc. 6th IEEE Int. Symp. Digit. Forensic Security (ISDFS)*, 2018, pp. 1–7.

[30] K. L. S. Priya and C. Rupa, “Block chain technology-based electoral franchise,” in *Proc. 2nd IEEE Int. Conf. Innov. Mech. Ind. Appl. (ICIMIA)*, 2020, pp. 1–5.

[31] E. Zaghloul, T. Li, and J. Ren, “d-BAME: Distributed blockchain-based anonymous mobile electronic voting,” *IEEE Internet Things J.*, vol. 8, no. 22, pp. 16585–16597, Nov. 2021.

[32] H. Kim, K. E. Kim, S. Park, and J. Sohn, “E-voting system using homomorphic encryption and blockchain technology to encrypt voter data,” 2021, *arXiv:2111.05096*.

[33] M. H. Au, J. K. Liu, W. Susilo, and J. Zhou, “Realizing fully secure unrestricted ID-based ring signature in the standard model based on HIBE,” *IEEE Trans. Inf. Forensics Security*, vol. 8, pp. 1909–1922, 2013.

[34] G. Zhao and M. Tian, “A simpler construction of identity-based ring signatures from lattices,” in *Proc. Int. Conf. Provable Security (ProvSec)*, 2018, pp. 277–291.

[35] Y. Sang, Z. Li, L. Zhang, H. Jiang, and K.-C. Li, “Lattice-based identity-based ring signature without trapdoors,” *Int. J. Embed. Syst.*, vol. 11, no. 3, pp. 386–396, 2019.

Jingyu Zhang (Member, IEEE) received the Ph.D. degree in computer science and technology from Shanghai Jiao Tong University, Shanghai, China, in 2017.

He is currently a Distinguished Associate Professor with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, China. He was a Visiting Scholar with the Department of Computer Science and Engineering, Ohio State University, Columbus, OH, USA, from 2014 to 2016. He was a Postdoctoral Researcher with the National Key Laboratory of Information Systems Engineering, National University of Defense Technology, Changsha, China, from 2020 to 2023. His research interests include distributed systems, blockchain, and big data.

Chenghao Wu (Member, IEEE) is currently pursuing the M.E. degree with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, China.

His research interests include secret sharing and ring signature technology.

R. Simon Sherratt (Fellow, IEEE) was born in Heswall, U.K. He received the B.Eng. degree in electronic systems and control engineering from Sheffield City Polytechnic, Sheffield, U.K., in 1992, and the M.Sc. degree in data telecommunications and the Ph.D. degree in video signal processing from the University of Salford, Salford, U.K., in 1994 and 1996.

Since 1996, he has been a Lecturer of Electronic Engineering with the University of Reading, Reading, U.K., where he is currently a Senior Lecturer of Consumer Electronics and currently the Head of Electronic Engineering. His research topic is signal processing in consumer electronic devices concentrating on equalization, communications layer 1, DSP architectures, and adaptive signal processing.

Dr. Sherratt received the IEEE Chester Sall First Place Best Transactions Paper Award in 2004 and the Best Paper in the IEEE International Symposium on Consumer Electronics 2006. He is a member of the IEEE Consumer Electronics Society AdCom (from 2003 to 2005 and from 2006 to 2008), holding the International Symposium on Consumer Electronics Liaison Officer Post from 2005 and the Society Awards Chair Post from 2006; a member of the IEEE Transactions on Consumer Electronics publications committee from 2004; the IEEE International Conference on Consumer Electronics ExCom; and the Vice-Technical Chair in 2006; the Chair of the IEEE International Symposium on Consumer Electronics 2004; a committee member in 2002, 2003, 2005, and 2006; and the Founder and the Current Chair of the IEEE UKRI Consumer Electronics and Broadcast Technology Joint Chapter.

Jin Wang (Senior Member, IEEE) received the M.S. degree from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2005, and the Ph.D. degree from Kyung Hee University, Seoul, South Korea, in 2010.

He is currently a Full Professor with Sanya Institute of Hunan University of Science and Technology, Sanya, China. He has published more than 400 international journal and conference papers. His research interests mainly include wireless ad hoc and sensor network, and network performance analysis and optimization.

Prof. Wang is a Fellow of IET.