DBE-voting: A Privacy-preserving and Auditable Blockchain-based E-voting System

Zhonghao Liu*, Xinwei Zhang†, Laphou Lao*, Guyue Li†‡, Bin Xiao*
*Department of Computing, The Hong Kong Polytechnic University, Hong Kong
†School of Cyber Science and Engineering, Southeast University, Nanjing, China
‡Purple Mountain Laboratories for Network and Communication Security, Nanjing, China
Email: zhonghao.liu@connect.polyu.hk, xwzhang1998@gmail.com, {cslhlao, csbxiao}@comp.polyu.edu.hk, guyuelee@seu.edu.cn

***Abstract*—Blockchain technology can construct a distributed and trusted ledger, which can be used for electronic voting (E-voting) systems to ensure the security of voting data and improve government credibility. However, existing blockchain-based solutions cannot fully fulfill five core requirements in E-voting, i.e., auditability, privacy, authentication, correctness, and unreusability, which make them unpractical in the reality. In this paper, we propose a Double Blockchain-based E-voting (DBE-voting) system, which consists of a private blockchain and a public blockchain. In the proposed system, the voter information is only recorded in the private blockchain for further auditing and the voting results are recorded in both blockchains. The voter's privacy can be protected in the private blockchain while the voting results can be queried in the public blockchain for verifying the correctness of the election process. Moreover, the ballot recorded in both blockchains is signed with a valid linkable ring signature to ensure authentication and unreusability. We propose an on-chain and off-chain hybrid storage mechanism to ensure the consistency and correctness of voting data in two blockchains. Experimental results demonstrate that the throughput of our system can reach 29 transactions per second when the block size is 512 KB. The security analysis shows that the DBE-voting is the first blockchain-based system that can meet all five requirements simultaneously.**

***Index Terms*—E-voting, blockchain, blockchain-based E-voting, Hyperledger Fabric, data consistency**

I. Introduction

Since blockchain technology can provide a distributed and immutable ledger to store data, this technology has been widely used in many fields, e.g., Internet of Things [1], [2]. Recently, the blockchain has also been introduced to *Electronic voting (E-voting)* system for addressing the security problems in traditional E-voting systems, which called *Blockchain-based E-voting (BE-voting)* [3].

In the BE-voting systems, the electronic ballots are recorded in the immutable ledger and distributed across the blockchain network to improve data authentication, and voters' identities are encrypted by cryptography when they make a transaction with the BE-voting system to protect voters' privacy. Without loss of generality, a reliable BE-voting system should satisfy the following core requirements: (1) Auditability: Only registered and verified voters can vote, and the records should keep who are involved in the voting; (2) Privacy: The voter's personal information and the voting process will be kept private; (3) Authentication: The ballot must be generated by valid voters and nobody can change them; (4) Correctness: The recording and counting processes should execute correctly; (5) Unreusability: A valid voter can only vote once. Unfortunately, current E-voting systems can not meet all these requirements [3]–[12].

Meeting all five requirements in a BE-voting system is still a huge challenge. For example, if a ballot recorded in a single blockchain contains voter personal information for auditing purposes and is kept confidential for privacy protection, voters and the public cannot verify the correctness of the election process since the data are encrypted and they are not allowed to access the system. Similarly, if the system satisfies privacy and correctness, the auditability may not be satisfied. The single blockchain, which records voting data without voter personal information, can be accessed by anyone to verify the correctness. However, without the related voter personal information, the system cannot prevent the impersonating attack [13], audit the election result, and investigate the legal issues.

In this paper, we propose a *Double Blockchain-based E-voting (DBE-voting)* system, including a private blockchain and a public blockchain to meet all five requirements. Both two blockchains record the same voting data. The private blockchain also records voters' personal information for further auditing, and no one will be able to access the private blockchain. The public blockchain allows voters to query their ballots and supervise the election tallying process to verify the correctness of the election. Besides, the linkable ring signature [14] is used to authenticate and encrypt the voters' identification and realize unreusability. Meanwhile, maintaining the data consistency between two blockchains becomes another challenge, because of the low scalability and lack of interoperability in the blockchain network [15]. We propose an on-chain and off-chain hybrid storage mechanism. The off-chain database is set up to store the public blockchain transaction data and the server can retrieve it to remake the transaction only with the public blockchain if the transaction fails in it. The main contributions of this paper are summarized as follows:

(1) **Double Blockchain-based E-voting System.**We propose a novel DBE-voting system, which consists of a private blockchain and a public blockchain, to meet all

the requirements for a reliable E-voting system.

(2) **Cross-chain Data Consistency.** We propose an on-chain and off-chain hybrid storage mechanism, which ensures data consistency between the private blockchain and public blockchain.

(3) **System Implementation and Security Analysis.** We develop a prototype based on Hyperledger Fabric and evaluate its performance. The results demonstrate that our system has a good performance when the block size is 512 KB. The security analysis proves that our system can satisfy all the requirements.

The remainder of this paper is structured as follows. Section II provides a literature review of the existing BE-voting research. The system model and the data consistency method is described in section III and section IV. Section V reports the evaluation results and analyses the requirements. Finally, we conclude this paper in Section VI.

II. Related Work

There exist some researches that try to realize a reliable E-voting system based on blockchain technology. As shown in Table I, we conclude that the existing researches do not provide a comprehensive BE-voting system to satisfy all the above-mentioned requirements.

Previous approaches can meet some requirements. For the privacy, most existing work encrypts the voters' identity [3], [5], [6], [8], [11] or customizes the system framework [7], [12]. For the correctness of the recording and counting process, some approaches program these processes in the smart contract [3], [5], [6], [9], [12], and others design the voting protocol [4], [7]. For the unreusability, researchers apply the E-wallet [3] or ring signature [6] or design algorithms [10] or zero-knowledge authentication [5] to restrict multiple voting. For the auditability, a few studies attempt to achieve it through utilizing the smart contract [9] or customizing the system framework [10] to involve voters' identities in the records.

In the existing work, Ngyuen et al. [5] and Wang et al. [6] are the most comprehensive work since they satisfy most requirements except the auditability. The authors in [5] wrote the smart contract in the system to maintain the correctness of the voting process. They designed a registration process to verify voters' identities, and the non-interactive zero-knowledge proof was used to protect the voters' privacy. They utilized zero-knowledge authentication to restrict voters' voting multiple times. However, they did not provide the auditability that the records did not keep who were involved in them and did not mention what type of blockchain to realize and evaluate the entire system. In [6], authors proposed a blockchain-based voting protocol for large-scale voting. They designed the registration process to verify the voters' identities and applied the ring signature to ensure authentication, unreusability and to protect voters' privacy. They utilized the smart contract and public verification mechanism to ensure the correctness of the recording and tallying process. However, the system architecture in this research was unknown, and the audit process was unclear.

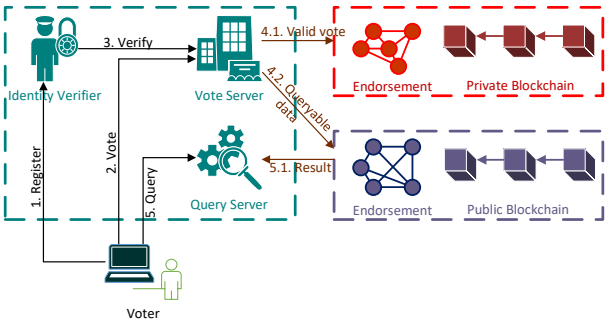


Fig. 1. The proposed DBE-voting architecture.

In summary, current E-voting systems can not meet all requirements or do not implement their proposed systems. In this paper, the proposed DBE-voting system can satisfy all the requirements mentioned above, and we implement a prototype to test its performance.

III. The Proposed DBE-voting

This section will devise a double blockchain-based electronic voting system, which satisfies and optimizes identified requirements and considerations. According to Fig.1, we introduce the main activities in the election process in this section.

A. Election creation

Election administrators design and write the election-related options in the vote server platform, for example, the candidate list and the voting district list. After that, they design the private chaincode and public chaincode with the decided option lists (In Hyperledger Fabric, the smart contract is called chaincode). Then, they install the chaincodes in the corresponding organizations, which affiliate with either private blockchain or public blockchain, and instantiate these organizations to create the first block of the election. At the same time, the administrators create the table of the off-chain database and design the query server platform, which displays the query result and data in the database. Finally, they build the connection by using fabric SDK between the central servers and blockchains (The fabric SDK supports the developers with powerful and convenient API).

B. Voter registration

The identity verifier processes the registration of a voter. It currently uses offline registration and online verification to authenticate the voters' identities. Offline registration requires the voter to send their personal information, including the ID number and phone number, to create an account on the identity verifier platform. The online verification requires the voter to send their real-time face ID and fingerprints to the identity verifier. The staff behind the identity verifier verify the voter's personal information and real-time biological information. If the voters pass the verification, they can generate the private key and the public key ring through the API supplied by the identity verifier on their local devices. The voting server

TABLE I
COMPARISON OF CURRENT E-VOTING SYSTEMS.

Relevant Literature	Auditability	Privacy	Authentication	Correctness	Unreusability	Implementation
Ethereum-based E-voting system [3]	×	√	√	√	√	×
Permissioned BE-voting system [4]	×	×	×	√	×	√
zVote [5]	×	√	√	√	√	√
Large-scale BE-voting protocol [6]	×	√	√	√	√	√
Prêt à Voter E-voting method [7]	×	√	√	√	×	√
Permissionless BE-voting protocol [8]	×	√	√	×	×	×
BroncoVote [9]	√	×	√	√	×	√
BE-voting system in P2P network [10]	√	×	√	×	√	×
Privacy-preserving BE-voting protocol [11]	×	√	√	×	×	×
Decentralized BE-voting system [12]	×	√	√	√	×	×
Our System (DBE-voting)	√	√	√	√	√	√

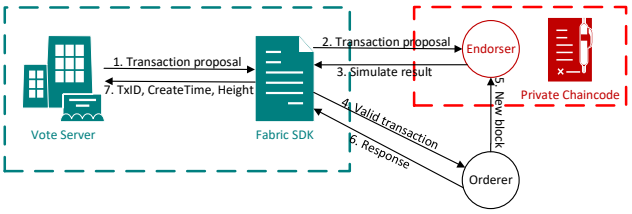


Fig. 2. Transaction flow of signed ballot in the private blockchain.

checks for the validity and linkability of the signature when voters send their signed vote to the vote server.

C. Vote verification

After the valid voter fulfills the vote and signs it, the vote server verifies the validity of the signature. Then, it checks for linkability with signatures that have been recorded in the private blockchain. The truth of signature validity represents the voter has a valid identity, and the truth of signature linkability represents the voter has voted. The vote server first checks the signature validity. Then, it checks the signature linkability if the validity is true. In the final, only one condition can make the system move to the following process: The validity is true, and the linkability is false.

D. Vote transaction in private blockchain

When the signed vote passes the verification, the vote server makes a transaction with the private blockchain through the fabric SDK (shown in Fig. 2). Since the blockchain network is isolated and its process is independent of the vote server, the fabric SDK can be treated as a "gateway" to build a connection between the vote server and private blockchain network. After the connection is built, it sends the transaction proposal to the endorsers by invoking the functions in fabric SDK (The peers will call the endorsers if they can execute the transaction proposal). The transaction proposal includes the operation type, the function name, voting data, and other parameters. The voting data contains the public key ring, candidate name, voting district, the linkable ring signature, and other customized personal information. The consensus mechanism in Hyperledger Fabric initiates when

TABLE II
EXAMPLE OF TRANSACTION IN PUBLIC BLOCKCHAIN AND RECORD IN OFF-CHAIN DATABASE.

TxID	Height	Candidate	Location	Signature	Create time
8dfa76...	4	Jason	Haidian	616242...	16:07:18

the endorser peer executes the transaction proposal. Supposing the result of the transaction proposal reaches a consensus in the private blockchain. In that case, the orderer returns the transaction's status, including transaction ID, block height and create time, back to the vote server (The orderer only focuses on receiving the transaction, generating block, and sending it to peer nodes.). Table II shows the combination of proposed parameters and the response, which is the new transaction format in the following system processes. In this table, the data of TxID, Height and Create time come from the private blockchain.

E. Data stored in off-chain database

The off-chain database stores the combined data (shown in Table II) in the off-chain database before the vote server makes a transaction with the public blockchain. Ensuring that the data queried by voters from the public blockchain have corresponding records in the private blockchain, the system must maintain data consistency between the private blockchain and the public blockchain.

F. Vote transaction in public blockchain

After the system stores the cross-chain data in the off-chain database, the vote server transacts with the public blockchain through the fabric SDK. The process details are similar to the vote transaction in the private blockchain. When the vote server receives the result returned from the public blockchain, it generates the unique receipt and sends it back to the voter. Voters can use the receipt generated from the vote server to their votes recorded in the blockchain.

G. Ballot query

Voters who receive the receipt from the vote server successfully can access the query server to search their ballots recorded in the blockchain. The voter needs to input the public

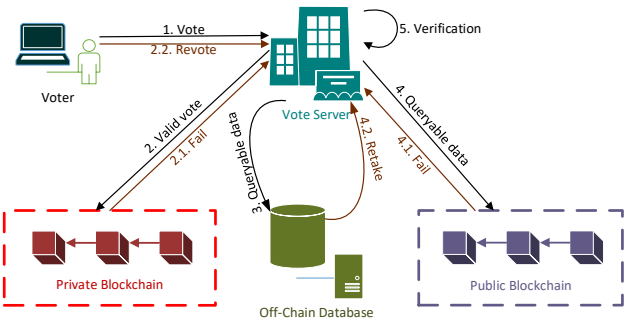


Fig. 3. The data consistency process in the proposed system.

transaction ID, as the keyword, to the query server platform. The query server makes a query operation with the public blockchain through the fabric SDK, which process is similar to the vote transaction in the private blockchain. Table II shows the search result returned from the query server, and the transaction ID represents the transaction made in the private blockchain. Voters can compare the unique private transaction ID recorder in the receipt with the result shown in the query platform and verify the validity of the linkable ring signature in their local device.

H. Tallying result

When the election is over, the vote server starts tallying the results based on the ballots recorded in the public blockchain. The vote server invokes the tallying function programmed in the chaincode, and endorsers of the public blockchain traverse the entire public blockchain and select the vote transaction recorded in it. Then, they count the ballot number for each candidate, compare them to get the result, and make a consensus. Eventually, the vote server gets a consistent result and publishes the final result to the citizen. During this process, the citizen can supervise the entire tallying process. If the final result is correct and the audit process is complete, the records in each peer's local storage need to be eliminated before the next election to ensure voter privacy.

IV. Data Consistency in DBE-voting

To ensure the ballot information searched by voters and counted in the tallying process does have the corresponding valid ballots recorded in the private blockchain. As shown in Fig. 3, Our proposed system returns the errors to voters' local devices if the transaction fails in the private or public blockchain. Then, the vote server remakes the transaction again. After the ballot is recorded in the public blockchain, the voter server invokes the specific verify function to check the consistency of the transaction above.

A. Ensuring the transactions are recorded in private blockchain

According to the consensus mechanism in Hyperledger Fabric [16] (The Hyperledger Fabric uses execute-order-validate architecture to help the system reach the consensus), if the transaction fails in executing phase, it will not be packed in the new block; If the transaction fails in validation, the packed transaction will be marked as invalid.

The vote server sends the transaction proposal to different endorsers through the fabric SDK and waits for simulated results. If the simulated results that the vote server received are different or the parameters in the transaction proposal do not satisfy the conditions set in the private chaincode. The endorsers return an error to fabric SDK even it satisfies the endorsement policy (Endorsement policy is a boolean expression to guide peers on how to determine whether the transaction is approved or not). Supposing the transaction proposal passes the conditions specified in the private chaincode and the simulated results are same. In that case, this transaction will be sent to the orderer to be packaged into a new block.

After the peer nodes receive the block sent from the orderer, they evaluate these transactions based on the default system chaincodes and the endorsement policy. If it does not pass the evaluation, the transaction will be marked as an invalid transaction. Finally, the peer nodes update this new block to their local ledger and return the result to the vote server. If the transaction success, the result contains the transaction ID and other related data stored in the peer's local ledger. Otherwise, the vote server notifies the voters and asks them to vote again since their ballots are marked as invalid transactions in the block.

This process ensures the ballot is recorded in the private blockchain when the vote server receives the transaction ID, create time, and block height from Hyperledger Fabric.

B. Ensuring the transactions are recorded in public blockchain

The second step to realize the data consistency is guaranteeing the queryable data (shown in Table II) are recorded in the public blockchain. The vote server stores the queryable data in an off-chain database before it transactions with the public blockchain. The transaction flow in the public blockchain is similar to the previous step, but it is different when they handle invalid transactions. Since the valid vote has been recorded in the private blockchain and the voter cannot vote again the vote server can only return the error to the voter's local device and solve the error by itself.

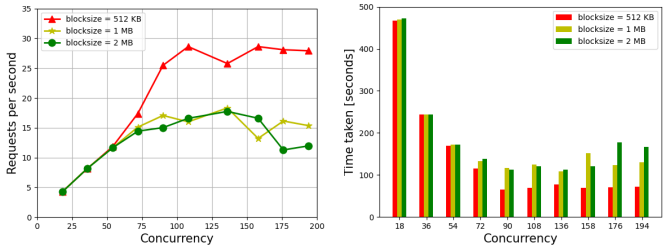
We designed the method to set the off-chain database to store the queryable data before the transaction starts. If the vote server fails to make the transaction in the public blockchain, it can retake the queryable data from the off-chain database and make a transaction again.

C. Verifying the data consistency before generating the receipt.

The last step to data consistency is verifying the correlation of properties that the vote server gets from the private and public blockchain.

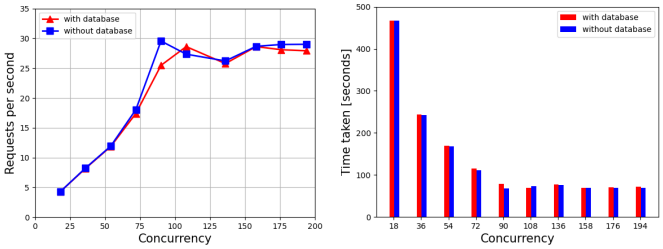
$$(T_{pub} - T_{pri}) <= (H_{pub} - H_{pri} + 1) * BatchTimeout \quad (1)$$

As shown in Equation 1, the *BatchTimeout* means the duration of generating one block, the T indicates the create



(a) (b)

Fig. 4. The impact of block size parameter on system performance. (a) The impact of block size on throughput. (b) The impact of block size on time taken for test.



(a) (b)

Fig. 5. The impact of data consistency process on system performance. (a) The impact of data consistency process on throughput. (b) The impact of consistency process on time taken for test.

time, and the *H* means the block height. The *BatchTimeout* is set at the system design and fixed when the system is launched. The left side represents the actual duration of recording the queryable data into the public blockchain. The right side represents the maximum theoretical duration of recording the queryable data into the public blockchain.

This logic is adaptive for verifying the data consistency in the situation that the vote server meets the error when making the transaction with the public blockchain. For example, a particular transaction fails to store in the public blockchain, and the vote server has to spend extra time to make the transaction again until the transaction is recorded in the public blockchain successfully. The public blockchain might generate new blocks in this period if the error does not affect the system's operation. So, this transaction's block height in the public blockchain may be larger than that in the private blockchain.

In this verification process, we assume the public blockchain records other transactions successfully, even the error happened in a particular transaction. This condition ensures the Equation 1 working correctly.

V. Evaluation

This section first evaluates the performance of our system. Then, we analyze whether the system meets the requirements.

A. Experiments

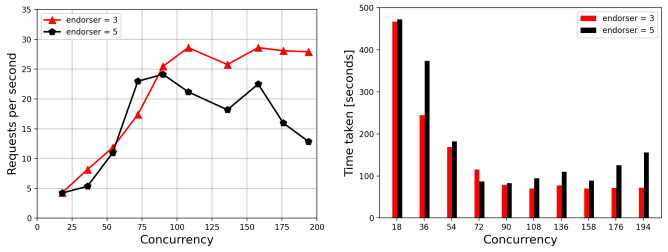
1) ***Setup****:* We implement our system based on the Hyperledger Fabric. In our experiment: (1) the prototype is realized on Fabric v1.4.2-preview and the system performance is monitored by a local logging process, (2) peers are hosted separately in Docker containers as dedicated VMs, (3) all clients are hosted in one node, (4) a single Fabric orderer node offers ordering service, (5) each blockchain has three endorsers and each endorser represents one organization (Org), (6) the endorsement policy is designed to satisfy the simplest BFT requirement, which is implemented as *OR(AND(PriOrgOne, PriOrgTwo), AND(PriOrgOne, PriOrgThree), AND(PriOrgTwo, PriOrgThree))*, and (7) the sqlite3 is applied as the off-chain database to store the cross-chain data.

2) ***Methodology****:* In the experiment, we assume all voters have passed the identity verification and ignore the time that voters fill the ballot form. They send the request to the vote server and wait to receive the receipt. We use the Apache Benchmark as the benchmarking tool to test the performance of our E-voting system. The total number of requests is not changed. We regularly increase the concurrency number in each test to see system performance changes. In addition, the *BatchTimeout*, as a system parameter in Hyperledger Fabric, can be treated as the response time when the voters get their receipt, and we set this parameter as 2 seconds for a good user experience.

3) ***Impact of block size****:* Fig. 4(a) shows that when the concurrency number is larger than 72, the system performance of 512 KB is better than those of 1 MB or 2 MB. Moreover, the 512 KB block size system is more stable than others when the concurrency number is larger than 90. The performances of 1 MB and 2 MB are not stable and stop increasing because the nodes crash when the concurrency number is larger than 136. Fig. 4(b) shows the time consumed in each test. We can observe that as the number of concurrencies increases and the total number of requests stays the same, the time taken for the test decreases. Since the system performance of 512 KB is better than those of 1 MB or 2 MB, its time taken is generally less than others.

Although the system throughput in this experiment is not sufficient for real-world government elections, we demonstrate the system's feasibility. To satisfy all the above-mentioned requirements, we implement two relatively isolated blockchains in one system, and our experiment shows that the design is achievable. Due to equipment limitations, the system prototype is implemented on a personal computer on which all peer nodes are hosted. We believe our system can achieve better performance when each node can be implemented on a single server.

4) ***Impact of data consistency process****:* The designed on-chain and off-chain hybrid storage scheme might become the bottleneck of the system's performance. We run an experiment by comparing the performance of adding this scheme into the system with the performance of removing this scheme to evaluate the impact of this design scheme. Results are depicted in Fig. 5(a), with the increasing number of concurrency, the



(a) (b)

Fig. 6. The impact of endorsement policy on system performance. (a) The impact of endorsement policy on throughput. (b) The impact of endorsement policy on time taken for test.

throughput reaches its peak and fluctuates between 25 and 30 requests per second. The time taken in Fig. 5(b) is not significantly different. It indicates that the data consistency process does not consume plenty of time to manipulate and verify the data consistency between the private blockchain and the public blockchain.

*5) **Impact of endorsement policy**:* The number of endorsers and the design of the endorsement policy are likely to change based on the actual government situation. To evaluate the impact of the endorsement policy, we increase the endorser number to five in each blockchain and update the policy that three of five endorsers should simulate the same result when reaching a consensus. Results are depicted in Fig. 6(a) and Fig. 6(b), the system that each blockchain with five endorsers needs more time to finish the requests than the system with three endorsers. It indicates that the increasing number of endorsers has a significant impact on the system performance since we set all peer nodes in one computer and the computing resources are not enough for the system. We believe this problem can be solved when each endorser is hosted on a single server, and their performances are not affected by each other.

Even though the performance of the system with five endorsers is not significant, it can complete 24 transactions per second. It still has a better performance compared to the work in [6], whose system with five nodes needs almost 3 seconds to verify a ballot.

B. Security Analysis

This subsection analyzes how our designed system satisfies the requirements. And the analysis is stated as follows:

1) Auditability: Impersonating attack [13] can disrupt the election by impersonating unqualified citizens as legitimate voters to cast the ballots on the government server. The E-voting system should ensure the identity of each voter is authentic and legitimate. The records should keep who is involved in the transactions to audit the election result and investigate the legal issues.

In our system, the voters are instructed to complete offline registration and online verification before joining the election. Offline registration requires the voters to submit valid personal information to the official website, including the ID number and phone number. The system produces a list of eligible voters privately for further verification. The online verification requires the voter, who has registered on the list, to provide a face ID and fingerprint to the official website. The online verifiers check the list and audit the validity of the voters' information submitted. This offline-online design leads the technologies to form an auditable verification system. Meanwhile, the records in the private blockchain contain the voters' public key ring and other personal information that can be traced back to voters when auditing elections or investigating legal issues.

2) Privacy: Privacy protection in E-voting is the biggest concern of voters because the leakage of privacy may threaten the personal and property safety of voters. When voters' privacy is leaked, the E-voting can be threatened by voter coercion attack [17]. The coercers can oblige the voter to vote as they wish.

The voter's identity will not be revealed, nor will the relationship between the identity and the vote be discovered in our system. The double blockchains system structure allows us to store the voting results and voter information separately. After the signed ballots have been recorded in the private blockchain, the system strips the personal private information and the public key ring to form the queryable data. Then, it is appended to a public blockchain for public supervision. The original ballots recorded in private blockchain only be used for legal purposes, and the private information will never be leaked to the public. This design keeps the voters' information confidential, and the public key ring brings anonymity to voters when they vote.

3) Authentication: Suffrage or enfranchisement, the right to vote, are important but seldom mentioned in the E-voting paper until the authentication is defined. Satisfying the authentication is essential to solving legal issues in E-voting [18].

The vote is generated by the valid voter, and no one can change it when it is added to the blockchain. The system designs an identity verifier to verify the validity of voters and authorize them the right to vote. The voters must register and verify themselves by sending personal information, real-time face ID, and fingerprints to the identity verifier. After the voters pass this process, the unique linkable ring signature is generated for each valid voter, representing they have the right to vote in this election. The valid voters sign the filled ballot with their unique linkable ring signature before submitting their ballots. Moreover, the distributed, append-only, and immutable properties in blockchain technology ensure that valid signed ballots cannot be changed when recorded. These technologies provide authentication in the system.

4) Correctness: The E-voting system must record and count valid ballots correctly to prevent server attacks such as data tampering [19].

In our design, the recording and counting processes are executed in a secured, distributed blockchain-based system, preventing hacker attacks from the server side. The peer nodes are launched in the Docker container, which supports a secure execution environment for the endorsers to execute the chaincode safely. As well as, the endorsers must comply with

the specific endorsement policy when running the chaincode. The endorsement policy requires the majority of the execution results generated by different endorsers must be consistent before updating the blockchain. It prevents internal corruption from disrupting the regular operation of the system. The system provides the query server, allowing voters to search the ballot data recorded in the public blockchain. The queried data contains the ring signature, and voters can verify the validity to prove the integrity and correctness of recorded ballots. The citizens can supervise the voting results, which have been recorded in the public blockchain, through the website supported by the system. This design prevents the government server from tampering with the ballots before being recorded in the blockchain. The trusted executing environment, PBFT scheme endorsement policy, and public supervision bring the recording and counting process correctness into our system.

5) Unreusability: Valid voters can only vote once and the E-voting systems need to satisfy this requirement to prevent the double-voting attack [18].

Multiple votes from the same voter can be detected and discarded. In our system, this requirement is supported by the linkable ring signature. When the voters submit the signed ballots to the system, the system checks the linkability of each signature besides the validity. The linkability of the ring signature is represented by the public key image stored in the signature data structure. The public key image is generated by a hash function whose input is the real public key. According to the principle of hash collision, if two hash values are equal, the input value of the hash function is also the same. The hash function used for generating the public key image is SHA3-256 and it is secure enough to satisfy the principle and prevent the hash collision attack. Therefore, if the signature of the submitted ballot is linkable with the signature recorded in the private blockchain, it means that the public key image of the two signed ballots is the same, thus inferring that the voter has already voted and the submitted ballot is invalid.

VI. Conclusion

This paper presents a DBE-voting system, which is the first BE-voting that can cover all the core requirements of a reliable BE-voting system, i.e., auditability, privacy, authentication, correctness, and unreusability. DBE-voting is powered by two innovative designs: 1) double blockchain architecture, consisting of a private blockchain and a public blockchain; and 2) an on-chain and off-chain hybrid storage mechanism that combines the off-chain database with the blockchain. Besides, we implement a prototype of our system and analyze whether our system can meet all requirements. The experimental results show that the system has high performance when the block size is only 512 KB while our data consistency method is not the bottleneck. The security analysis shows that the DBE-voting can satisfy all the core requirements simultaneously.

Acknowledgment

This work was supported in part by Grants from the HK RGC GRF (Project No. PolyU 15217321), NSFC/RGC Joint Research Scheme (Project No. N_PolyU529/22), and Natural Science Foundation on Frontier Leading Technology Basic Research Project of Jiangsu under Grant BK20222001.

References

[1] C. Boudagdigue, A. Benslimane, A. Kobbane, and J. Liu, "Trust management in industrial internet of things," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3667–3682, May 2020.

[2] R. Hussain, D. Kim, J. Son, J. Lee, C. A. Kerrache, A. Benslimane, and H. Oh, "Secure and privacy-aware incentives-based witness service in social internet of vehicles clouds," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2441–2448, Aug. 2018.

[3] F. Þ. Hjálmarsson, G. K. Hreiðarsson, M. Hamdaqa, and G. Hjálmtýsson, "Blockchain-based e-voting system," in *Proc. International Conference on Cloud Computing (CLOUD'18)*, San Francisco, CA, USA, Jul. 2018, pp. 983–986.

[4] R. Hanifatunnisa and B. Rahardjo, "Blockchain based e-voting recording system design," in *Proc. International Conference on Telecommunication Systems Services and Applications (TSSA'17)*, Lombok, Indonesia, Oct. 2017, pp. 1–6.

[5] T. Nguyen and M. T. Thai, "zvote: A blockchain-based privacy-preserving platform for remote e-voting," in *Proc. International Conference on Communications (ICC'22)*, Seoul, South Korea, May 2022, pp. 4745–4750.

[6] B. Wang, J. Sun, Y. He, D. Pang, and N. Lu, "Large-scale election based on blockchain," *Procedia Computer Science*, vol. 129, pp. 234–237, Mar. 2018.

[7] K. M. Khan, J. Arshad, and M. M. Khan, "Secure digital voting system based on blockchain technology," *International Journal of Electronic Government Research (IJEGR'18)*, vol. 14, no. 1, pp. 53–62, 2018.

[8] Y. Liu and Q. Wang, "An e-voting protocol based on blockchain," Cryptology ePrint Archive, Report 2017/1043, 2017, https://ia.cr/2017/1043.

[9] G. G. Dagher, P. B. Marella, M. Milojkovic, and J. Mohler, "Broncovote: Secure voting system using ethereum's blockchain," in *Proc. International Conference on Information Systems Security and Privacy (ICISSP'18)*, Funchal, Madeira, Portugal, Jan. 2018, pp. 96–107.

[10] H. Yi, "Securing e-voting based on blockchain in p2p network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1–9, May 2019.

[11] W. Zhang, Y. Yuan, Y. Hu, S. Huang, S. Cao, A. Chopra, and S. Huang, "A privacy-preserving voting protocol on blockchain," in *Proc. International Conference on Cloud Computing (CLOUD'18)*. San Francisco, CA, USA, Jul. 2018, pp. 401–408.

[12] J.-H. Hsiao, R. Tso, C.-M. Chen, and M.-E. Wu, "Decentralized e-voting systems based on the blockchain technology," in *Proc. Advances in Computer Science and Ubiquitous Computing (CUTR'17)*, vol. 474, Singapore, Dec. 2017, pp. 305–309.

[13] B. Adida, "Helios: Web-based open-audit voting," in *Proc. USENIX Security Symposium (USENIX Security'08)*, vol. 17. San Jose, CA, USA, Jul. 2008, pp. 335–348.

[14] S.-F. Sun, M. H. Au, J. K. Liu, and T. H. Yuen, "Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero," in *Proc. European Symposium on Research in Computer Security (ESORICS'17)*, Cham, 2017, pp. 456–474.

[15] Z. Liu, Y. Xiang, J. Shi, P. Gao, H. Wang, X. Xiao, B. Wen, and Y.-C. Hu, "Hyperservice: Interoperability and programmability across heterogeneous blockchains," in *Proc. ACM SIGSAC Conference on Computer and Communications Security (CCS'19)*, London, United Kingdom, Nov. 2019, pp. 549–566.

[16] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proc. the thirteenth EuroSys conference (EuroSys'18)*, no. 30, Porto, Portugal, Apr. 2018, pp. 1–15.

[17] J. P. Gibson, R. Krimmer, V. Teague, and J. Pomares, "A review of e-voting: the past, present and future," *Annals of Telecommunications*, vol. 71, no. 7, pp. 279–286, Jun. 2016.

[18] K.-H. Wang, S. K. Mondal, K. Chan, and X. Xie, "A review of contemporary e-voting: Requirements, technology, systems and usability," *Data Science and Pattern Recognition*, vol. 1, no. 1, pp. 31–47, 2017.

[19] Z. Li, B. Xiao, S. Guo, and Y. Yang, "Securing deployed smart contracts and DeFi with distributed TEE cluster," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 3, pp. 828–842, Mar. 2023.