J Inf Process Syst, Vol.16, No.2, pp.421~434, April 2020
https://doi.org/10.3745/JIPS.03.0135

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

A Study on Electronic Voting System Using Private Blockchain

Chang-Hyun Roh* and Im-Yeong Lee**

Abstract

The development of digital technology has changed the lives of many people in terms of the velocity and convenience of completing tasks. This technology has also been applied to the process of voting, yet electronic voting is seldom used. The existing electronic voting scheme operates by applying various encryption algorithms. This type of electronic voting can be problematic since the administrator is given full authority. The administrator cannot always be trusted, and the contents of the ballot could be forged or tampered by a single point of failure. To resolve these problems, researchers continue to apply blockchain technology to electronic voting. Blockchain technology provides reliability and data integrity because all untrusted network participants have the same data. In this paper, we propose an electronic voting system that secures reliability by applying blockchain technology to electronic voting and ensures secret voting.

Keywords

Blockchain, Distributed Ledger System, Electronic Voting

1. Introduction

Voting is often referred to as a flower of democracy, as it is a process in which voters can express their opinions directly. The current paper voting system is quite costly and time consuming, and because of this, electronic voting has been proposed as an alternative. Conventional electronic voting has attempted to provide convenient and secure voting using cryptographic algorithms. However, in a centralized structure, it is possible that administrators may take advantage of their privileges and forge or tamper with the voting data. Additionally, since it enables the possibility of a single point of failure, it is only utilized with small numbers of people so far. Recently, various research has been conducted on applying the blockchain technique to solve the issue of trust and the threat of a single point of failure. Blockchain refers to the fact that all participants in a peer-to-peer (P2P) network are assured of the reliability of the same data by applying various consensus algorithms. The application of this feature to electronic voting ensures that all voting data is processed as a transaction and stored in a block so that malicious network participants cannot easily tamper with the data.

In this paper, we propose increased integrity in voting by encrypting the contents of the ballot using blockchain electronic voting and ensuring confidentiality to voters so that they cannot be identified by

Manuscript received October 1, 2018; first revision April 1, 2019; second revision August 19, 2019; third revision November 4, 2019; accepted December 9, 2019.

Corresponding Author: Im-Yeong Lee (imylee@sch.ac.kr)
* Dept. of Computer Engineering, SoonChunHyang University, Asan, Korea (rohch@sch.ac.kr)
** Dept. of Computer Software Engineering, SoonChunHyang University, Asan, Korea (imylee@sch.ac.kr)

the content. To this end, Section 2 describes related research on electronic voting and blockchain, and Section 3 proposes a way to meet all the security requirements of electronic voting. In Section 4, we analyze the proposed security requirements and analyze the voting processing speed.

2. Related Research

2.1 Electronic Voting

Electronic voting means electronicizing all the procedures for voting. The voting process consists of setting up candidates, voting, identification, voting, counting, and scoring. All processes must be performed reliably to have a legitimate effect as an election [1]. A variety of techniques have been proposed, such as short message service (SMS) electronic voting using a smartphone or a telephone, voting using a general server, and tablet PC [2-4].

Characteristics of electronic voting

Electronic voting is implemented by applying appropriate cryptographic algorithms. This scheme must satisfy the following voting requirements as proposed in [1]:

- Completeness: All information compiled on the ballot should be handled correctly.
- Soundness: Votes should not be interrupted by unjust voters.
- Privacy: No one else should know what the voter voted for.
- Unreusability: Voters must be able to vote only once.
- Eligibility: Voters can only vote if they have the power to vote.
- Fairness: No matter what happens, you should not influence your vote.
- Verifiability: Anyone should be able to verify the results of the voting.

Problems of electronic voting

Electronic voting has the advantage of fewer constraints of time, space, cost less to proceed with the vote counting that is fast and convenient. However, many countries are not currently using it. This is due to the lack of reliability of the system and the problems of data forgery and modulation. In recent years, increasingly there are many studies that suggest the electronic voting system by applying the blockchain to solve these problems [5].

2.2 Blockchain

The blockchain was first proposed by Satoshi Nakamoto [6] in 2009 and the most important feature of blockchain technology is that all participants have the same data in the form of the distributed database described above. Periodically generated blocks are connected in order of block thus, if the data is falsified or modulated, the attacker cannot easily modulate the data because the subsequent blocks must be changed from the previous block (Fig. 1). Blockchain has various classifications and characteristics, as indicated in Table 1. The blockchain consists of competitive and non-competitive types of consensus algorithms depending on how the blockchain is maintained [7]. The description is in the Table 2.

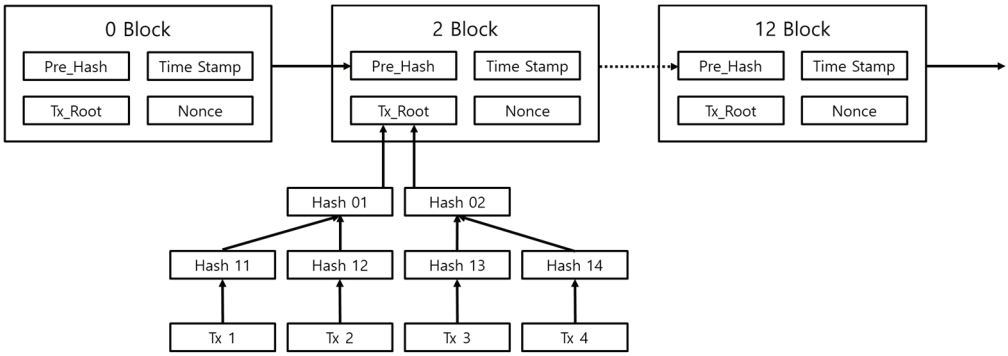

Fig. 1. Configuration of blockchain.

Table 1. Classification and characteristics of blockchain

	Public	**Private**	**Consortium**
Definition	Everyone joins the network	Independent use of an institution	Several institutions join
Authority	Anyone can read	Only authorized organizations can view	
Validate and approve	Anyone can verify and approve	Approved authorities and supervisors	
Creating a transaction	Anyone can create transactions	Only those responsible are responsible	
Governance	Difficult to change the first rule	Can be changed according to the decision of participating organization	
Characteristic	Public, distributions	Private, centralize	
Speed	Slow	Very fast compared to public blockchain	
Case	Bitcoin, Ethereum, Ripple	Hyperledger Project, EEA, R3CEV	

Table 2. Comparison existing consensus algorithms

	PoW	**PoS**	**DPos**	**PBFT**	**Raft**
Participation	Open	Open	Permissioned	Permissioned	Permissioned
Energy saving	No	Partial	Partial	Yes	Yes
Tolerated power of adversary (%)	<25	<51	<51	<33.3	<33.3
Fork occurrence	Occur	Occur	Occur	Not occur	Not occur
Example	Bitcoin	Ethereum	EOS	Hyperledger Fabric	Quorum

2.2.1 Competitive consensus algorithm

The competing consensus algorithm ensures unity in the blockchain network by accepting only one consensus that satisfies the condition through proceeding to different consensuses at several places in the untrusted system at the same time. In the case of the competition algorithm, all the nodes participating in the network can be a part of the negotiation even if they are not directly participating. However, there is a possibility of double payment or double voting because the branch occurs when competing nodes generate blocks at the same time. Competitive algorithms include proof-of-work (PoW), proof-of-stake (PoS), and delegated proof-of-stake (DPoS).

Proof-of-work

PoW is a structure in which nodes participating in a block chain work and receive compensation. For example, it can be said to draw the 1st place in a running race. PoW finds the data and nonce values smaller than the size set using the hash algorithm. Because PoW is the key to fast hashing, hardware with high computational power is needed.

Proof-of-stake

PoS is a structure that receives compensation for ownership (assets, additions, etc.). PoS is proposed to solve the problem of excessive energy consumption of PoW. The initial algorithm of the PoS was inefficient because it is advantageous to those who have a lot of stakes. In order to solve this problem, we have improved based on the amount of possessed property and the date of possession.

2.2.2 Non-competitive consensus algorithm

Non-competitive consensus algorithms are a way of proceeding only one agreement at a time in a trusted system. And all nodes participating in the network check and reply to agreements. However, there is a disadvantage that if the malicious node participates more than a certain number, the system cannot be maintained. Practical Byzantine fault tolerant (PBFT) algorithm, Paxos algorithm, and Raft algorithm are examples of non-competitive algorithms.

Paxos

Paxos is a consensus algorithm that tolerates failures. Paxos consists of two stages: advance confirmation and acceptance processing. The pre-confirmation is carried out by the node that proposes the consensus, delivering the message for the consensus to another node. Then, if more than half of the other nodes agree with the message, it is a structure that generates a block.

Raft

Raft is an algorithm for solving the Paxos algorithm. Paxos proceeded with a verification node, but Raft selects a leader. Because the leader is elected, the processing and consensus process is accelerated (Fig. 2). In addition, the process has been simplified because it does not go through a validation node. It is being used by Quorum, which is a representative project of the Enterprise Ethereum.

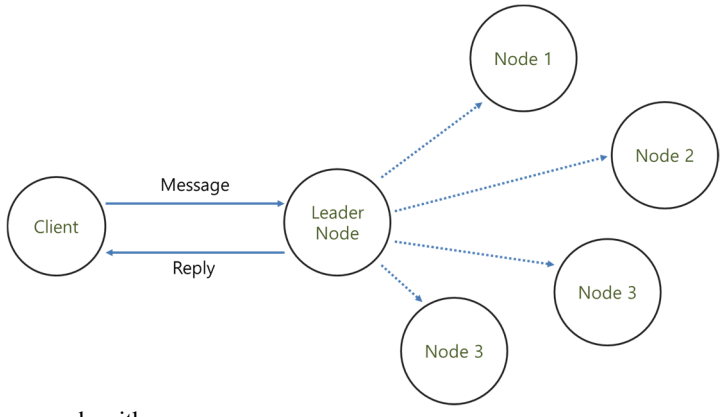

Fig. 2. Raft consensus algorithm.

Practical Byzantine fault tolerance

PBFT is an algorithm that solves the problems of uncertainty and performance which are the disadvantages of competitive algorithms. Uncertainty refers to the likelihood of a branch, and performance refers to the number of data that can be processed per second. The characteristics of the PBFT should be known to all participants in the network in advance. The disadvantage is that it can be maintained only when $n = 3f + 1$ when the number of nodes is n (Fig. 3).

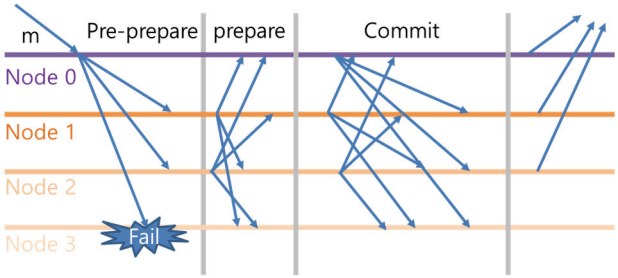

Fig. 3. PBFT consensus algorithm.

2.3 Electronic Voting Using Blockchain

Existing electronic voting has not gained confidence due to the possibility of data tampering. To solve this problem, [8] tried to conduct an electronic voting for the first time using the first blockchain, the Bitcoin. However, since the winner uses the Bitcoin by using the lottery protocol, it has difficulty in actual application. Since then, [9] has described the electronic voting structure in the Bitcoin as a process of authentication, voting, and counting. However, there is a disadvantage that the user is not provided with privacy. [10] added privacy by adding a ring signature to the Bitcoin electronic voting system. Since then, much research has been done and Ethereum based electronic voting system has been proposed. [11] is provided privacy using blind signatures, and [12] is presented electronic voting using smart contracts.

However, there is a problem in that fairness cannot be guaranteed because all researches transmit transactions in plain text. In addition, the above researches have the possibility of double voting due to the use of public blockchain. When such problems arise, it is suitable for electronic voting private blockchain to be utilized. A private blockchain can be used to restrict participants and prevent the possibility of data tampering. In addition, the branch generated in the public blockchain does not occur in the PBFT consensus algorithm, thereby preventing double voting.

For this reason, this proposed method proposes an electronic voting system that operates in private blockchain. We propose an electronic voting system that provides privacy and fairness through data encryption by preventing the double voting by using PBFT, which is the algorithm of consensus of private blockchain.

3. Proposed Scheme

In this section, the authors propose a method to satisfy the various security requirements of electronic voting using private blockchain. The proposed method consists of the voting manager, the voting server, and the counting server, and assumes that all participating objects are in secure communication (Fig. 4).

Additionally, it includes the election commission, the polling location, and the ticket office. In this proposed scheme, after the election committee establishes a vote, the public key is transmitted to the voting servers by generating public/private key pair at the counting server. Subsequently, the election committee registers candidates and voters after they authenticate, and at the time of voting, the voters connect to the voting server and insert data into the transaction and vote. Finally, when the ballot is complete, the election committee sends the counting message and whole block to the counting server, decrypts it with the private key of the counting server, and broadcasts it after the counting.

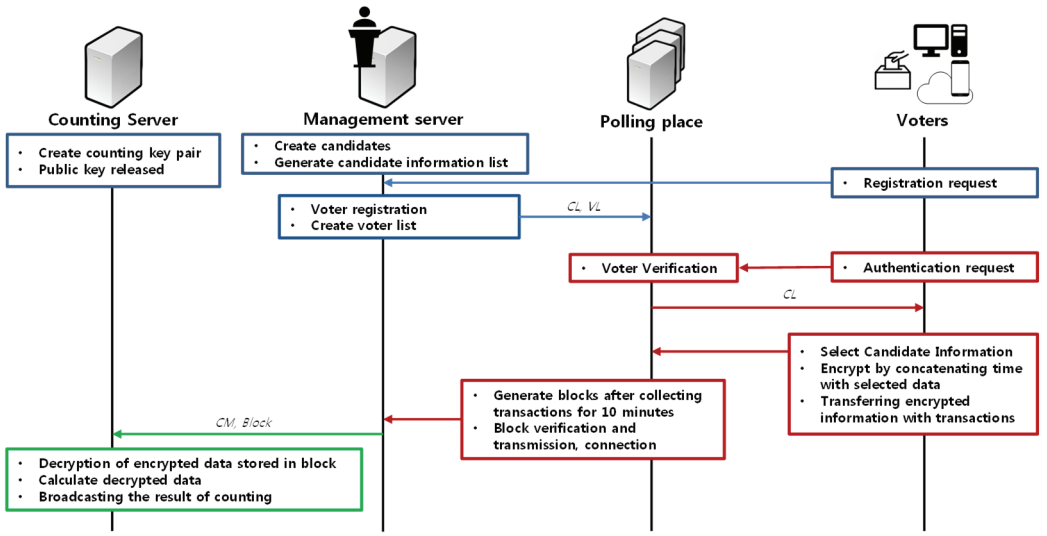

Fig. 4. Scenario of the proposed scheme.

Table 3. System parameters

Parameter	Description
ID_{V_i}	ID of voter V_i
PW_{V_i}	Password of voter V_i
D_{SK_C}	Decryption with secret key SK_C
CL	Candidate list
TS	Transaction
VL	List of voters
RC_i	Registration code of i-th voter
VM	Verification message
CM	Counting message
PK_C	Public key of the counting server
SK_C	Secret key of the counting server
E_{PK_C}	Encryption with secret key PK_C
CI_i	i-th candidate information
$Block$	Block that can be connected to a blockchain
V_i	i-th voter
PK_{V_i}	Voter's public key
SK_{V_i}	The voter's private key
Sig_{SK_V}	Signed by voter's private key

System parameter

The system parameters in the proposed scheme are as follows in Table 3. Each participant object indicates: M is management server; P, polling place; C, counting server; and V, voter.

Preparation phase

In the preparation phase, the management server sets up a vote. The setting will include the voting subject, content, and time. The counting server has a phase of generating a public private key pair and transmitting the public key to polling place.

Step 1. The M establishes the poll using data such as voting topic, content, time, and so on.

Step 2. The C creates its *PK*/*SK* pair and sends it to the P.

Registration phase

In the registration phase, candidates and voters are registered and candidates submit their information to generate a candidate list. The management server generates VL through the registration code RC, transmitted after the identity authentication, and sends the public/private key to the V.

Candidate registration (Fig. 5)

Step 1. Candidate requests registration to the M.

Step 2. The M requests CI. CI can include name, age, affiliation, and so on.

Step 3. Candidate sends his information CI to M.

Step 4. The M receives CI and generates CL.

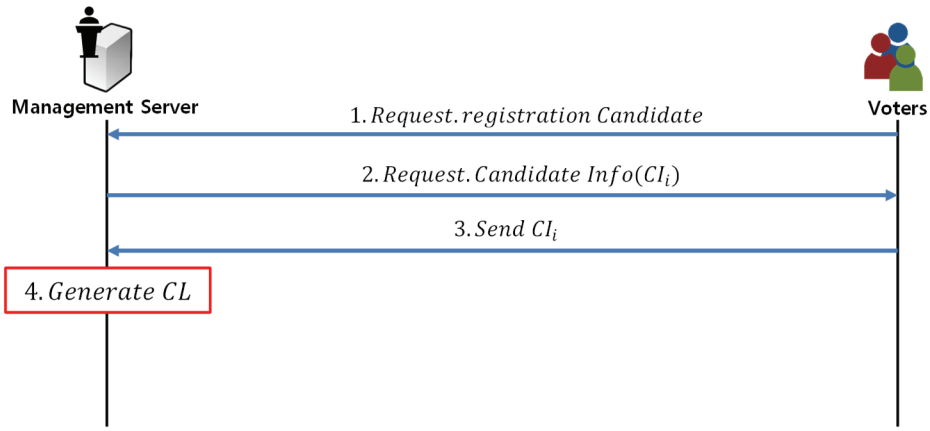

Fig. 5. Candidate registration phase.

Voter registration (Fig. 6)

Step 1. V requests registration of V to the M after register the member with ID/PW.

Step 2. The M requests the V's identity information VI. VI can contain name, age, address, and so on.

Step 3. V verifies his or her identity with data such as passport and ID card.

Step 4. The M sends the registration code RC to the V. RC generates a PK/SK pair with a key generation algorithm and a random seed value.

Step 5. V generates PK/SK using RC and sends PK to the M.

Step 6. The M receives the PK and generates CL and VL through the VI and sends it to the P.

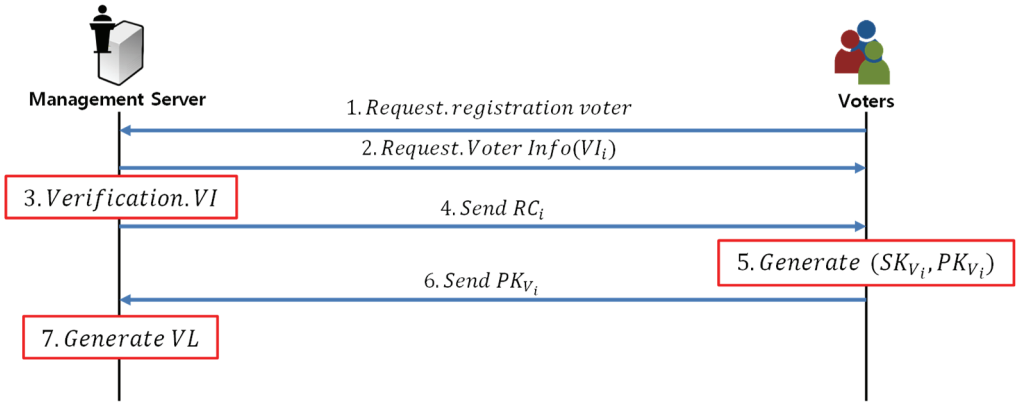

Fig. 6. Voter registration phase.

Voting phase

In the voting phase (Fig. 7), the voter has the process of voter authentication process at the polling place, receives the candidate information, selects the candidate, concatenates the selected data with the voting time, encrypts the encrypted data, and inserts the encrypted data into the transaction. The structure of the transaction is shown in Table 4.

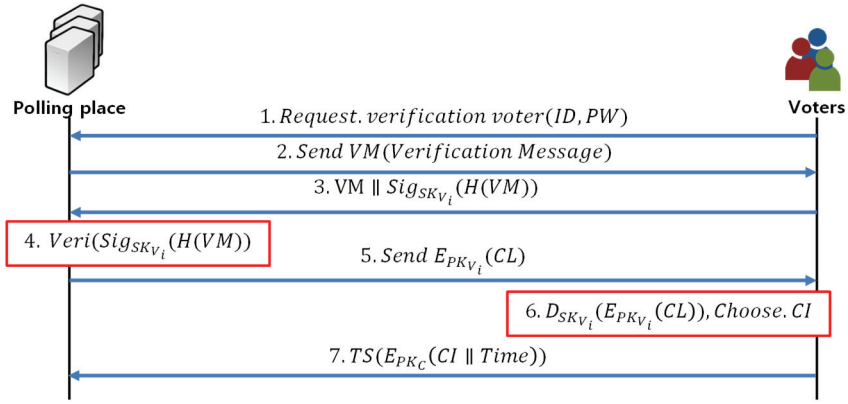

Fig. 7. Voting phase.

Step 1. V use ID / PW to request voting at the P after login.
Step 2. The P sends the VM to the V.
Step 3. V signs the VM with the private key of the revocation and sends it to the P.

$$signs \rightarrow VM \ \| \ Sig_{SK_v}(VM)$$

Step 4. P verifies the signature with PK.

$$\text{Verify} \rightarrow \ D_{PK_v}(VM) = VM$$

Step 5. P transmits the CL encrypted with the public key.
Step 6. V decrypts the encrypted data CL and selects the candidate.
Step 7. After encrypting the selected CL with PK, insert CL || Time into TS and send it to the P.

$$\text{TS}(E_{PK_C}(CL \ \| \ \text{Time}))$$

Step 8. The P receives the TS and checks that the V has voted.

Table 4. Components of a transaction

Field	Description
From	Sender
To	Recipient
Signature	Verifier's signature
Data	Fields that can contain messages

Block creation phase

The block generation step transmits the transaction in which the voter has submitted the encrypted data to the polling place and has block creation (Fig. 8). The polling place then checks the validity of the transaction with the other polling places. The validated transaction is sent to the M, and the M collects the transaction for one minute, generates a block, and sends it to all polling places to connect to the blockchain (shown in Fig. 9). The configuration and examples of the blocks are shown in Table 5.

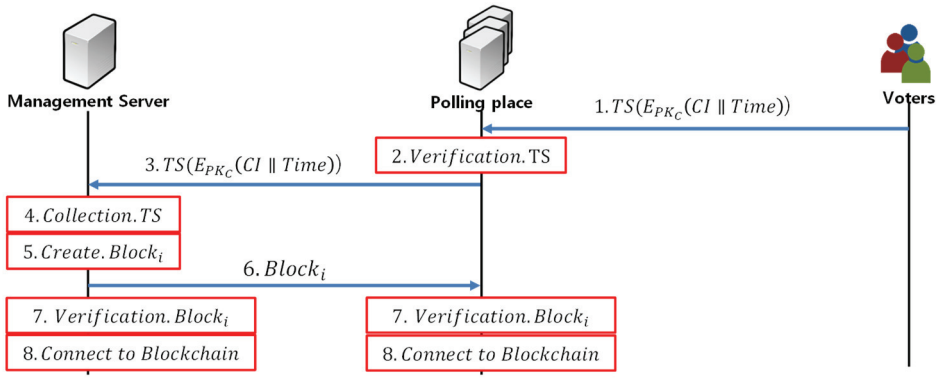

Fig. 8. Block creation phase.

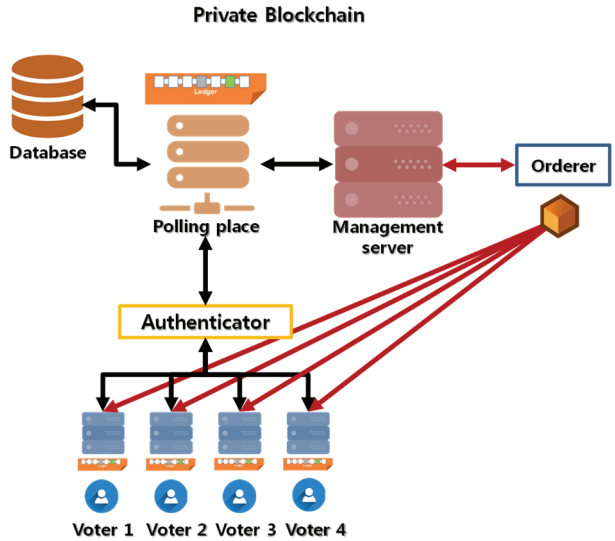

Fig. 9. Block creation.

Step 1. V shall disseminate all his/her votes to the polls.

Step 2. P requests the validity of the TS transmitted to the other P. The validity check checks whether the user who created the transaction is a registered user and whether the user voted once.

Step 3. When the validation is completed normally, P signs the TS and sends it to the M.

Step 4. The M collects the TS, generates the *Block*, and transmits the *Block* to all P.

Step 5. All P connect the *Block*s received to the blockchain.

Table 5. Components of a block

Field	Description
Index	Block generation number
Hash	The hash value of the block
Previous Hash	The hash value of the previous block
Next Hash	The hash value of the next block
Transactions	The hash value of the next block

Counting phase

The counting phase is carried out by the management server sending the counting message and the entire block to the counting server. The counting server decrypts the encrypted data in the transaction of the block with the private key of the counting server to proceed with counting (Fig. 10).

Step 1. When the voting time is over, the M sends CM and all *Block*s to the C.

Step 2. Extracts all TS from *Block*.

Step 3. Decrypt the encrypted candidate information inserted in TS into SK.

$$D_{SK_C}\left(E_{PK_C}(CL \parallel \text{Time})\right) \rightarrow \text{CL}$$

Step 4. Confirm the CI and broadcast the results after completing the counting.

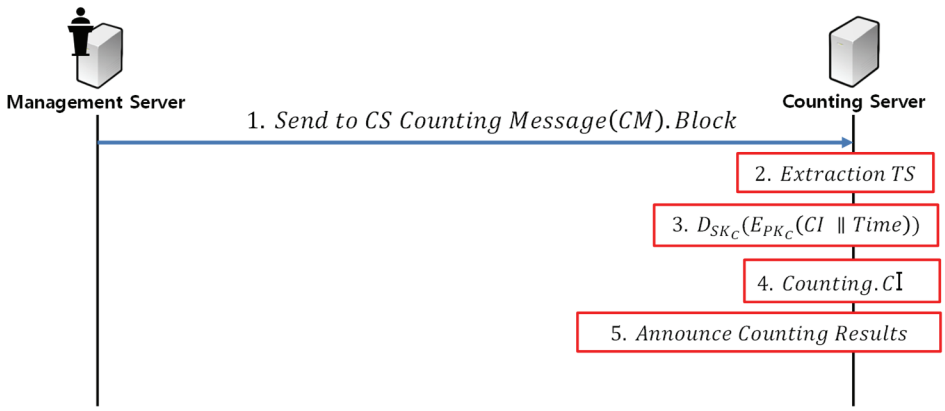

Fig. 10. Counting phase.

Validation phase

The verification step proceeds by requesting the election private key to the counting server and requesting voting data from the blockchain. It receives both the election private key and the voting data, and decrypts and validates the voting data with the election private key.

Step 1. The verification requester requests SK_C of C.
Step 2. The verification requester requests all *Block*s of blockchain.
Step 3. Decrypt the encrypted candidate information inserted in TS into SK.

$$D_{SK_C}\left(E_{PK_C}(CL \parallel \text{Time})\right) \rightarrow \text{CL}$$

Step 4. Confirm the CI and validate the results after completing the counting.

4. Analysis

This section analyzes the satisfaction and processing speed of security requirements in the proposed electronic voting system based on the private blockchain. This proposal is unique in that it involves an electronic voting system that provides confidentiality and fairness among the security requirements for encrypting voting data and creating transactions (shown in Table 6).

- Completeness: This proposed scheme, which operates on blockchain, provides the integrity of the data and provides a complete vote. → $Offered\ by$ TS $\left(E_{PK_C}(CL \parallel \text{Time})\right)$ $in\ block$
- Soundness: It is possible to identify the wrong voter with this list of voters in proposed scheme. In the case of a duplicate vote, only the first round of voting is accepted as a normal. → $Offered\ by\ Voter\ List\ and\ Block\ Creation\ in\ Step\ 2$
- Ballot Data Privacy: The contents of the ballot form in the proposed method are encrypted to the candidate's information by the counting server's public key, so the contents of the ballot form are kept secret. → $Offered\ by\ E_{PK_c}(CI \parallel Time)$
- User Privacy: User privacy is provided to generate and proceed with the new election party key. → $Offered\ by\ registration\ phase\ in\ voter\ regestration$
- Unreusability: When a voter completes a vote, they check that they have voted, and when they receive a transaction, they cannot vote to check whether the polls are duplicated through validation. → $Offered\ by\ Block\ Creation\ in\ Step\ 2$
- Eligibility: The management server creates a voter list and verifies the signature through the public key list of the voter list, so the voter cannot vote. → $Offered\ by\ Voter\ List$
- Fairness: The proposed method does not affect voting because the contents of the ballot are transmitted in encrypted form and therefore not disclosed. → $Offered\ by\ E_{PK_c}(CI \parallel Time)$
- Verifiability: After the election is over, the verifier can verify the results by disclosing the private key of the counting server. → $Offered\ by\ validation\ phase$

The method using the Bitcoin platform [9,10] also uses the characteristics of cryptography and trustworthy TTP (Trusted Third Party) institutions to vote but does not guarantee fairness because it does not encrypt the voting data. Also, [11,12] does not guarantee confidentiality and fairness because it confirms that the Ethereum platform will vote on the transaction of the cryptographic currency, but the transaction is open and sent to the candidate address. In this proposed scheme, when the transaction is created, the voting data is encrypted and sent to the polling place, ensuring confidentiality and fairness, and ensuring reusability and unreusability through validation of voter list and transaction. In addition, it is possible to provide user privacy to newly generate and proceed with the election key, and to provide

privacy to the data generated by the user to encrypt the data voted by the user. Verification also ensures that after the voting is done, the private key of the counting server will be made public so that everyone can verify it.

Table 6. Comparison of security requirements of existing blockchain-based electronic voting system

Item	Lee et al. [9]	Wu et al. [10]	Liu and Wang [11]	Hjalmarsson et al. [12]	Proposed scheme
Completeness	Offered blockchain	Offered blockchain	Offered blockchain	Offered blockchain	Offered by using blockchain
Soundness	×	×	×	×	Offered
Ballot Data Privacy	×	×	×	×	Offered by using $E_{PK_c}(CI \parallel Time)$
User Privacy	×	Offered ring signature	Offered Blind signature	×	Offered by using generate keys every election
Unreusability	Offered hash table	Offered TTP	Offered smart contract	Offered smart contract	Offered by using TTP, PBFT algorithm
Eligibility	Offered hash table	Offered TTP	Offered smart contract	Offered smart contract	Offered by using voter list
Fairness	×	×	×	×	Offered by using encryption
Verifiability	Offered blockchain	Offered blockchain	Offered blockchain	Offered blockchain	Offered by using blockchain

The comparison in table 7 of the processing speed is compared with the algorithm of the consensus on the Bitcoin platform used in [9,10] and the algorithm of Ethereum platform consensus of [11,12] with the PBFT consensus algorithm used in the proposed method. The comparison will be made by 100 people, 1,000 people, 5,000 people according to the number of votes. The PBFT algorithm can process about 1,000 to 1,500 transactions per second [13]. This can vary depending on the system configuration and scale. The PBFT algorithm differs greatly from the processing speed of the cryptography platform and I think that this algorithm can be used in medium and large electronic voting.

Table 7. Number of votes per second per platform (unit: second)

	Number of transactions			Processing speed (TPS)
	100	1,000	5,000	
Bitcoin consensus (PoW) [9,10]	5	200	1,000	15–20
Ethereum consensus (PoW) [11,12]	25	250	1,250	3–4
Proposed scheme (PBFT)	0.06	0.6	3.3	1,000–1,500

5. Conclusions

In the online environment, the electronic voting system has been undergoing many studies to ensure the voting process and the reliability of the results. Although many approaches have been proposed for solving these problems using blockchain, many security threats and inherent characteristics in blockchain

platforms have not yet offered the security requirements in electronic voting environments [14]. In this paper, we propose an electronic voting system using blockchain by using voting data encryption in the way that voting data is disclosed in electronic voting using existing blockchain platform, We propose a system that satisfies all the security requirements of the system and improves the performance of the algorithm by about 50 times than the algorithm of the algorithm in the existing public blockchain by using the algorithm of the private blockchain. In addition, the most important feature of the proposed electronic voting system is that the voting data can be transmitted by encrypting the transaction, so that it cannot be negated when generating the blocks by collecting the transactions at the management server, so that a transparent voting result can be expected. If an electronic voting system is developed and commercialized in this way, it will be expected to reduce the cost of voting and increase the turnout rate.

In future, it will be necessary to research on the processing speed and safety that can be extended to real world by applying to private blockchain platform.

Acknowledgement

This paper was supported by Soonchunhyang University and ICT (MSIT), Korea, under the Information Technology Research Center support program (No. IITP-2019-0-00403) supervised by the Institute for Information & communications Technology Planning & Evaluation (IITP).

References

[1] A. Fujioka. T. Okamoto, and K. Ohta, "A practical secret voting scheme for large scale elections," in *Advances in Cryptology – AUSCRYPT'92*. Heidelberg: Springer, 1992, pp. 244-251.

[2] A. Ali Mohammed and R. A. Timour, "Efficient e-voting android based system," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 11, pp. 43-48, 2013.

[3] S. Anandaraj, R. Anish, and P. V. Devakumar, "Secured electronic voting machine using biometric," in *Proceedings of 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, 2015, pp. 1-5.

[4] G. Rubner, "mbclick: an electronic voting system that returns individual feedback," in *Proceedings of 2012 IEEE 7th International Conference on Wireless, Mobile and Ubiquitous Technology in Education*, Takamatsu, Japan, 2012, pp. 221-222.

[5] K. Wust and A. Gervais, "Do you need a Blockchain?," in *Proceedings of 2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, Zug, Switzerland, 2018, pp. 45-54.

[6] S. Nakamoto, "A peer-to-peer electronic cash system," 2009; https://bitcoin.org/bitcoin.pdf

[7] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of Blockchain technology: architecture, consensus, and future trends," in *Proceedings of 2017 IEEE International Congress on Big Data (BigData Congress)*, Honolulu, HI, 2017, pp. 557-564.

[8] Z. Zhao and T. H. H. Chan, "How to vote privately using bitcoin," in *Information and Communications Security*. Cham: Springer, 2015, pp. 82-96.

[9] K. Lee, J. I. James, T. G. Ejeta, and H. J. Kim, "Electronic voting service using block-chain," *Journal of Digital Forensics, Security and Law*, vol. 11, no. 2, article no. 8, 2016.

[10] Y. Wu, "An e-voting system based on Blockchain and ring signature," M.S. thesis, University of Birmingham, UK, 2017.

[11] Y. Liu and Q. Wang, "An e-voting protocol based on Blockchain," 2017; https://eprint.iacr.org/2017/1043.pdf.

[12] F. P. Hjalmarsson, G. K. Hreidarsson, M. Hamdaqa, and G. Hjalmtysson, "Blockchain-based e-voting system," in *Proceedings of 2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, 2018, pp. 983-986.

[13] H. Kim, "Analysis of security threats and countermeasures on Blockchain platforms," *Journal of KIIT (Korean Institute of Information Technology)*, vol. 16, no. 5, pp. 103-112, 2018.

[14] G. Z. Qadah and R. Taha, "Electronic voting systems: requirements, design, and implementation," *Computer Standards & Interfaces*, vol. 29, no. 3, pp. 376-386, 2007.

Chang-Hyun Roh **https://orcid.org/0000-0002-1883-0550**

He earned a bachelor's degree in Computer Software Engineering from Soonchunhyang University in 2017. He currently holds a master's degree in computer science from Soonchunhyang University. His research interests include blockchain, electronic voting, and smart contract.

Im-Yeong Lee **https://orcid.org/0000-0002-8856-0103**

He received the B.S. degrees in Department of Electronic Engineering from Hongik University, Korea, in 1981 and the M.S. and Ph.D. degrees in Department of Communication Engineering from Osaka University, Japan, in 1986 and 1989, respectively. From 1989 to 1994, he had been a senior researcher at Electronics and Telecommunications Research Institute (ETRI), Korea. Now he is a professor in Department of Computer Software Engineering from Soonchunhyang University, Korea. His research interests include cryptography, information theory, computer & network security.