International Journal of System Dynamics Applications
Volume 11 • Issue 3

Ballot Blockchain System Design and Development on Ethereum Platform

Meng-Hsuan Fu, Shih Hsin University, Taiwan
An-Shyang Lee, Shih Hsin University, Taiwan

ABSTRACT

Ballots are often held for fair decisions such as party theme selecting; however, the existing traditional ballots have some problems involving amount of human resources, cost of places, equipment, time and traffic, and repeated procedures. In order to solve the aforementioned issues, a ballot blockchain system is designed and implemented based on the smart contract of Ethereum. It is designed on the core blockchain technologies of the decentralized ledger technology, using a secure hash algorithm, anonymous user, incorruptible data, and adopting a public blockchain. The ballot blockchain system is implemented based on the MetaMask verification and the Remix interface development environment. The smart contract plays the role of the decision maker for controlling ballot activities instead of numerous human tasks. All ballot transactions are recorded in the ballot blockchain permanently when the ballot is completed. The aim of the ballot blockchain system is to achieve a fair, less time-consuming, secured, and transparent environment.

KEYWORDS

Anonymity, Consensus Algorithm, Cryptography, Distributed Computing, Flexible Voting, Incorruptibility, Online Voting, Secure Hash Algorithm

INTRODUCTION

Information technology, including sensors, radio frequency identification (RFID), infrared, the Internet of Things (IoT), automation mechanisms, geographic information systems (GIS), identification systems, and so on, are based on a high-density internet infrastructure and are adopted in various industries to raise productivity and efficiency and reduce costs and human resources. Although considerable information technology has been adopted, such as communications, the internet, data-processing methods, automation, and so on, technology is still upgraded and advanced. Applications are increasingly developed under existing information technology using adaptive processing mechanisms, especially for internet security and data privacy, to build a safer internet environment and suitable services.

Blockchain was noticed because Nakamoto (2018) proposed electronic currency in 2008. Gartner is a major data analytics company, which presents the top 10 strategic technology trends for the next year during the fourth season of the year. Blockchain was listed from 2017 to 2020, consecutively, which indicates its influence in data technology (Gartner, 2020). Blockchain has been applied in many fields, including financial applications, food supply chain management, medical services, logistic business, and education.

DOI: 10.4018/IJSDA.287115

The ballot is a fair decision technique that is held for various purposes, such as representative member voting between nations, presidential elections in countries, decision-making in businesses, and opinion voting in families. Presently, voting is usually held in a traditional way, which includes attending the specified place in person and then receiving and casting a paper ballot. This method seems open and fair but has a higher cost on front-end works, including human and paper resources, and it is time-consuming and causes some unexpected chaos. Therefore, some businesses operate online voting using internet providers or communication applications, where the ballot environment is built based on internet-connected devices and a related application installation. Unlike traditional elections, online elections could be achieved without limitations of time and places. However, some privacy and internet security issues may be concerned; therefore, blockchain technologies are used to remedy them in this research. In which, there are some challenges of implementing ballot blockchain should be considered including the verification procedures of voters, the programming rules of the smart contract, and the policy making of preventing the vote cheating.

The existing traditional ballot has some problems involving amount of human resources, cost of places, equipment, time and traffic, and repeated procedures. This study is extended from the previous research (Fu, 2019), in order to solve problems with traditional elections, the blockchain mechanism design and development of ballot activities are proposed. The design is divided into three parts, including user registration, a secret ballot, and the ballot creation of the whole structure. In this mechanism, the characteristics of blockchain include decentralization, anonymity, incorruptibility, and encryption, which is adopted to build a reliable ballot blockchain. In this paper, the contributions include 1) accelerate the ballot processing procedures, 2) decrease the cost of human resources, software development and hardware devices, 3) raise the willingness of people to vote by reducing traffic time, and 4) increasing the flexibility of the voting procedure under the full security environment of the real-time ballot blockchain. The study is organized as related work which is composed of smart contract, blockchain and its technologies discussed in Background section. The procedures and tools of the ballot blockchain mechanism are described in the section of Main focus of the article, and the practical result of the real-time ballot blockchain is shown in Implementation section. Then, the concise conclusion is written in the end.

BACKGROUND

Smart Contract

The smart contract is built with infrastructure, contract, operation, intelligence, manifestation, and application layers (Wang et al., 2019). A smart contract is executed depending on the rules written in the programming language. Transactions proceed safely by following the logic rules of the smart contract without third-party intervention. Transaction content cannot be changed once the smart contract is deployed. Therefore, the smart contract may reduce the error rate and increase transaction fairness and reliability. Presently, the authors of Mohanta et al. (2018) and Shukla et al. (2018) proposed the research that smart contract is widely used in many blockchain-related applications.

The Ethereum Virtual Machine (EVM) is an independent environment for operations of smart contracts; thus, general online services may not be executed on it. The resource unit in the EVM is called gas, which has a limit. The smart contract would fail in execution if its functions were too complex to exceed the gas limitation. Ether (ETH) is a famous cryptocurrency issued by Ethereum. MetaMask is a browser plug-in Ethereum wallet used in managing ETH, but other wallet information, such as the wallet password and the private key, is not stored in MetaMask. Users could execute MetaMask using an Ethereum smart contract or Decentralized Applications (DApp) on the browser instead of building Ethereum nodes. Solidity is an Ethereum programming language used in EVM coding, whose syntax is similar to JavaScript. It is a high-level object-oriented programming language built for realizing smart contract operations. In other words, the rules of a smart contract are written

using Solidity and are executed in EVM (Zinca & Negrean, 2018). Remix is an Integrated Development Environment (IDE) based on the browser, which does not require the installation of any package for executing the programs. Remix provides the functions of content management, smart contract compiler, input and output, and relative parameter setting. In addition, Remix could perform the testing, debugging, and deploying of the DApp on the browser (Nguyen, 2019). Moreover, Remix is a powerful compiler and is integrated with Solidity as the execution environment, which is primarily used in building smart contracts and supports both online and offline development. Remix also achieves smart contract interaction by connecting with MetaMask. In this study, the technology of smart contract will also be adopted for reducing amount of human resources, time consuming, and task errors. EVM is selected as the implementation environment for the smart contract operating and the programming language of Solidity is the embedded language in Ethereum. The MataMask wallet will be used to manage ETH when transactions happened. Remix IDE is chosen as the environment of the real-time ballot blockchain in this work.

Blockchain

Nakamoto (2018) presented the electronic cash system based on a cryptography mechanism, in which peer-to-peer transmission technology was used for cash payment without third-party intervention. Moreover, each transaction was stored with timestamps and a hash and then was recorded in a block using a time sequence that solves the problem of double spending. Decentralization, transmission, encryption, and timestamps are methods adopted in blockchain (Fromm, 2017).

A transaction in a block is a record, and a block can store many transactions. A block is a basic unit, and each block connects, forming a blockchain (i.e., a blockchain is composed of many blocks). All data in the blockchain are encrypted using cryptography and are stored using distributed ledger technology (DLT). Because of the DLT, the blockchain is accessed and maintained by all nodes in the same blockchain network (Nakamoto, 2018)(Wang et al., 2018). In detail, the parts of a block are divided into the block size, block body, and block header. The block size records the data size using 4 bytes. The block body is for storing transactions, and the block header comprises three groups of metadata. The metadata provides information related to the data, such as the description, explanation, query information, and model structure. The first group of metadata records the previous block hash value to link two blocks together, and the second group of metadata includes the timestamp, difficulty target, and nonce to describe the mining competition. The third group of metadata calculates the hash value of all transactions in the current block using the Merkle root (Dhumwad et al., 2017).

The Merkle root uses a secure hash algorithm (SHA) to transform data into a meaningless hash value. For example, SHA-256 is one of the major SHAs used in the blockchain and generates a 256-bit hash value for data messaging with 32 bytes. The first block in the blockchain is called the genesis block, and the remainder are called blocks. A block is composed with a block header and a block body. A block header is used to store the metadata which includes data description, location and explanation. A block body is used to store all information related to the transactions. The elements and description of the genesis based on the Ethereum are shown in Table 1. The elements and description of the block header are shown in Table 2 and the elements and description of the block body based on Bitcoin are shown in Table 3, (Zhang et al., 2019)(Antronopoulos, 2017).

According to the degree of data transparency, the blockchain is divided into three types, including the public, private, and consortium blockchains. In the first, the public blockchain is a fully transparent blockchain, in which each person can be allowed to join the public blockchain as a miner or a node. Unlike traditional organizations with a central management department, all nodes in the same public blockchain network have the same right to access and maintain it. The data are distributed and stored in each node's ledger to form a decentralized database. In other words, all nodes are allowed to access and view the content in the public blockchain. The most famous examples of a public blockchain are the cryptocurrencies of Bitcoin, Ether, and Litecoin. In this study, the public blockchain is adopted for ballot. Second, the private blockchain is the strictest. Each person must receive permission to

Table 1. The Elements of a Genesis Block

Column Name	Description
Block ID	Block ID denotes block sequence in the blockchain, and block ID of the genesis block is 0.
Timestamp	The time of block inserts into the blockchain.
Data	The message in the block, and "This is a genesis block" in data of genesis block.
Hash	The block head of the block is transformed into a hash which is a unique value.
Previous Hash	The block head of the previous block is transformed into a previous hash. The previous hash is empty in the genesis block because there is no previous block.
Difficulty Target	Difficulty target defines the difficulty of mining by modifying the algorithm in order to set the difficulty of confirming the hash value, in which the difficulty target will refresh in every 2016 blocks.
Gas Limit	Gas is the unit of transaction fee. Gas fee = Gas limit * Gas price Each transaction operation should pay the gas fee, which depends on the complexity of transaction. The maximum of gas limit is set as 21000, and the gas price is changeable and it affects the renew speed of blockchain. The remain of gas will be returned into user's account after the transaction complete.
Nonce	The counter of the algorithm, hash value will be changed once the nonce modified.

Table 2. The Elements of a Block Header

Column Name	Size	Description
Version	4 bytes	The blockchain version is used to trace the software and its related protocols.
Timestamp	4 bytes	The timestamp of the block is built in the blockchain.
Previous Hash	32 bytes	The block head of the previous block is transformed into a previous hash. The hash of this block is generated considering with the previous hash.
Hash	32 bytes	The hash is generated with all the transactions in the block using the algorithm of Merkle root.
Difficulty Target	4 bytes	The POW difficulty of the block.
Nonce	4 bytes	The number of the POW processing times.

Table 3. The Elements of a Block Body

Column Name	Size	Description
Version	4 bytes	The blockchain version is used to trace the software and its related protocols.
Transaction ID	4 bytes	The transaction ID is used to make queries of the transaction information.
Transaction Hash	32 bytes	The transaction hash of the Merkle root is used to confirm the transaction is correct without any duplication or counterfeit.
Sender Address	16 bytes	The sender address of this transaction.
Receiver Address	16 bytes	The receiver address of this transaction.
Sender Transaction Counter	4 bytes	The amount of Bitcoin is sent in this transaction.
Recipient Transaction Counter	4 bytes	The amount of Bitcoin is received in this transaction.
Digital Signature	4 bytes	The digital signature information of sender and receiver such as the private key and public key.
Timestamp	4 bytes	The timestamp of the transaction is added in the blockchain.
Transaction Size	2 bytes	The size of this transaction.

join the private blockchain. The actions of the nodes are limited in the private blockchain, which is usually built for a specific purpose such as university certification, medical treatments, and so on. It is the blockchain with the lowest degree of decentralization, and a third party exists in it. The strict of the consortium blockchain is between the public and private blockchains, the nodes are given the permission to join the consortium blockchain. The actions of the nodes have some limitations in the consortium blockchain, which is usually built for organizations, companies, or cooperation between them. In this study, each ballot will be seen as a transaction in a block, which contains many transactions. All the ballots will be recorded as the block format as above in the blockchain without any modification. The real-time ballot blockchain will be designed as a public blockchain.

Blockchain Technology

The SHA is used as a data encryption method in blockchain, and it has several forms, such as SHA-1, SHA-256, and SHA-512. Of these, SHA-256 is famously applied in Bitcoin. In general, each bit of content is hashed into a unique message using the SHA, and the hashed message cannot be traced back to the original data. However, a hashed message might be traced back to the original message when the data use the same SHA to hash. Moreover, SHA-256 is used to compress the data into a meaningless message, and the length of the compressed data is fixed in 256 bits, regardless of the original data length (Abe et al., 2018).

Due to the distributed process in the blockchain network, the consensus algorithm is used to deal with a) the cheating problem on the decentralized system, b) the transaction synchronizing process, and c) the fairness issue (Sankar et al., 2017).

Proof of Work (POW) and Proof of Stake (POS) are two popular consensus algorithms. The POW gains rewards through competing in computation, and POS obtains rewards based on the volume of the stakes. In POW, miners solve mathematical problems using various devices, such as a Central Processing Unit (CPU), Graphics Processing Unit (GPU) or Application Specific Integrated Circuit (ASIC) to compete for the block-entry right. The complexity of the problem is automatically adjusted by the volume of the miners; thus, the computation is challenging for the miners but is easy for the service provider to verify. Here, the first miner who correctly solves the problem receives the reward. However, POW consumes considerable resources, such as electricity; hence, the cost might be more

than the reward (Ogawa et al., 2018). Thus, minority miners who hold the majority of the resources may control the network. In this case, minority miners who control more than 51% of the computation ability could control the blockchain. Because of the complex procedures of POW, the transactions spend a substantial amount of time on waiting processes; therefore, time consumption is currently the biggest issue for POW. Unlike POW, POS does not require expensive devices and considerable resources for competing for the blocks; instead, the authority of producing a new block depends on the asset amount. The miners who own more coins have a higher probability of obtaining the block-entry right. Nevertheless, the issue of time consumption still exists in POS (Košťál et al., 2018).

The threat modeling of security and privacy for blockchain applications may concern. The software-centric threat modeling, also known as system-centric threat modeling, is one of the common threat modelings, it focuses on architecture or system design and purposes to find the threats. For example, Herbert and Cerbo (2019) list the security threats of blockchain, and propose the analysis approach and then implement it. Almashaqbeh etc. (2019) also present a framework of a cryptocurrency-focused threat modeling, which is used to search the cryptocurrency risks by using matrices. The security development lifecycle of Microsoft uses the system-centric threat modeling to search the attacks. There are some blockchain applications of the software-centric threat modeling proposed (Dimitri et al., 2019).

Two main blockchain platforms are introduced as follows. First, Hyperledger Fabric, which is one of the major projects produced by the Linux Foundation, is a branch of Hyperledger. Hyperledger Fabric focuses on building a DLT with a higher security level (Clincy & Shahriar, 2019). It supports smart contract development in many programming languages, including Java, Go, and Node.js. (Silvestre et al., 2019). Second, Ethereum develops smart contracts through DApp, which is written using the programming language Solidity. Ethereum has a cryptocurrency called Ether (ETH), which is used on message transactions. These two are open-source blockchain platforms and support smart contract development. However, Hyperledger tends to provide services for registered members, and statistics are opened in Ethereum, which tends to the public blockchain. The blockchain platform of Ethereum is used for the experiment in this study. In this study, the technologies of blockchain include the data encryption method of SHA, POS consensus algorithm, decentralized, and user anonymous will be adopted in the real-time ballot blockchain built on Ethereum.

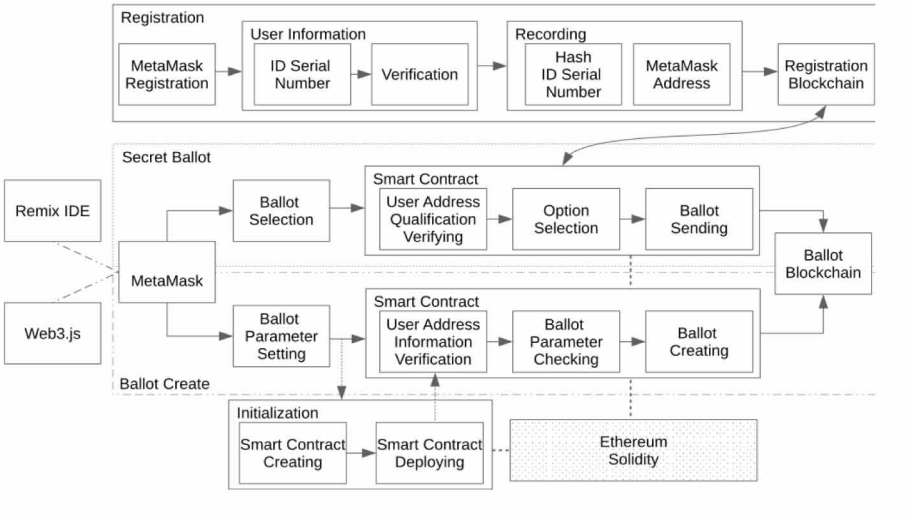
MAIN FOCUS OF THE ARTICLE

Mechanism of Real-Time Ballot Blockchain

In mechanism design and testing of the real-time ballot blockchain, MetaMask, Remix IDE, and Web3.js are applied, and their relationships are described as follows. In this setup, MetaMask is a browser plug-in for the ETH wallet for users to manage their ETH easily and safely. Remix IDE is used to develop a smart contract, which is written using the programming language Solidity. In addition, Remix is used to connect the blockchain testing network, main network, or private network of Ethereum. Moreover, Web3.js is used to connect and interact between webpages and the blockchain and smart contract, such as performing nodes status transferal or smart contract content checking. Here, the smart contract is deployed to one of the MetaMask testing networks using Remix, and web3.js connects with the testing network to implement the methods of the smart contract in the ballot blockchain mechanism design environment before deploying in the Ethereum main network.

The structure of the ballot blockchain mechanism is shown in Figure 1. In the preparation stage, users must install the MetaMask plug-in into the browser of Google Chrome, which supports the MetaMask ETH wallet to connect with the webpage of Web3.js to call the methods of the smart contract. MetaMask connects with Remix to link users and the smart contract of Ethereum. In the first stage, users should register the MetaMask account to manage their ETH for attending the ballot transactions. In MetaMask registration, users would be asked to set their respective passwords for

Figure 1. Mechanism of Real-Time Ballot Blockchain



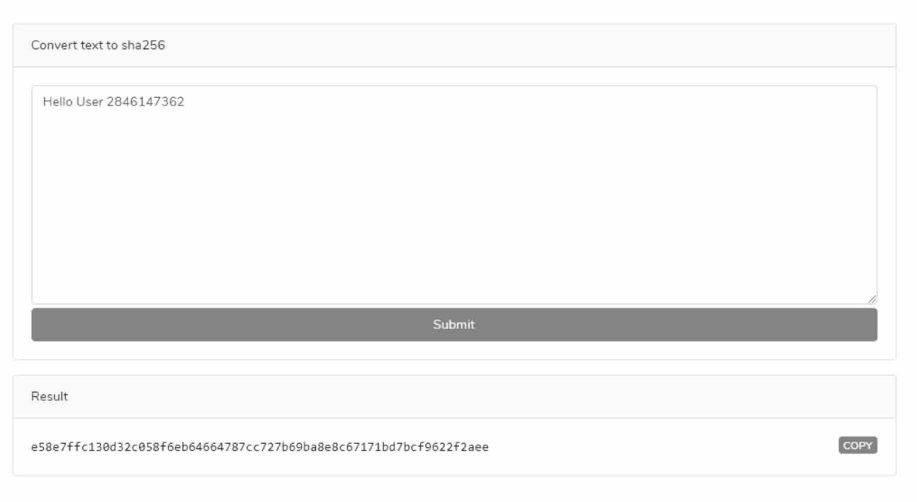
accessing their wallets. Then, 12 words are given by MetaMask that should be stored carefully and secretly to recover the wallet when needed. MetaMask provides an Ethereum main network, some testing networks, and a private network for the user selection, and it also offers five ETH for accessing the services, which, in this study, is ballot transaction operation. After MetaMask registration, users that register for a ballot need only two pieces of information: their ID serial number (on the backside of a Taiwanese ID card) and the MetaMask address.

The ID serial number is verified through a combination of checking rules. Then, the ID serial number is hashed into a meaningless string to protect the user's privacy using SHA-256, where a message of any length is hashed into 256 bits of data with 64 strings of hexadecimal numbers. For example, the ID serial number is combined with 10 digital numbers and is hashed into a hash, the textual content 'Hello User 2846147362' is converted to 'e58e7ffc130d32c058f6eb64664787cc727b69ba8e8c67171bd7bcf9622f2aee' of SHA-256 shown in Figure 2. The SHA-256 is sensitive of the characters, all minor changes will alter the hash including a space, punctuation, and upper case or lower case of letters. In the registration phase, only the hash of the ID serial number and MetaMask address are recorded in the registration blockchain.

The ballot procedure is divided into a secret ballot and ballot creation, and all attendants can check the ballot held on the ballot blockchain platform. In the secret ballot stage, the users should log into MetaMask, a browser plug-in Ethereum wallet, which connects with functions of Remix and Web3.js to execute the smart contract at the beginning. Here, the users log into the MetaMask account, which is not connected to the users' personal information, to operate the anonymous ballot. Afterward, users can select a ballot from the list and then confirm that selection, which triggers the smart contract to verify the user qualification of the ballot. The user address is checked regarding whether the user ever attended this ballot, if so, the user is unable to attend the ballot, and the process ends. If the user passes the verification of ballot qualification, the user is allowed to enter the ballot and make a decision. The decision is sent to the ballot, which is renewed. Then, the transaction is recorded in the ballot blockchain.

In the ballot creation stage, users should log into their MetaMask account first, similar to the secret ballot. If no ballot has been created, the smart contract is created and deployed in the ballot blockchain initially, and the smart contract is written by the programming language, Solidity, of

Figure 2. Hash of ID Serial Number (https://timestampgenerator.com/tools/sha256-generator)



Ethereum. Moreover, the ballot creator can set the relative ballot parameters through the ballot setting interface. Then, the smart contract verifies the user address information to create a new ballot. All the ballot parameters are checked to avoid any collisions from happening and being sent into the blockchain when completing the new ballot creation.

In the ballot blockchain mechanism, the smart contract plays the role of recording the user's attending information and the ballot content, which replaces the traditional paper contract. Thus, it should be deployed in the ballot blockchain to deal with various situations. When a user has a vote, the smart contract operates the procedures of user voting, including checking the user address, selecting the ballot, making options, renewing the statistics, and sending the transactions to the ballot blockchain using the methods and functions of the smart contract. Furthermore, the smart contract executes the processes of ballot setting, using the creator's address, ballot ID, date period, opinions, types, public or private, and so on, and the related parameters are employed when the new ballot is created. The smart contract can be called and operated through the Web3.js function library and connected using MetaMask to display it on the webpage.

IMPLEMENTATION

In the experiment, the ballot webpage was built on the WampServer infrastructure, and users must register a MetaMask account. Users can manipulate the ballot through a personal computer with a Windows, Mac, or Linux operating system or smart devices with an Android or iOS operating system. To operate the smart contract, a web browser, such as Google Chrome, should be installed on the personal computer to operate MetaMask. The app should be downloaded to operate on smart devices, such as smartphones or tablets. In the ballot blockchain platform, anyone could create ballots or attend them and check the results of the ballots through the user's MetaMask account. Thus, personal information is not exposed and remains anonymous. The practical test is divided into two parts: the ballot creation and secret ballot.

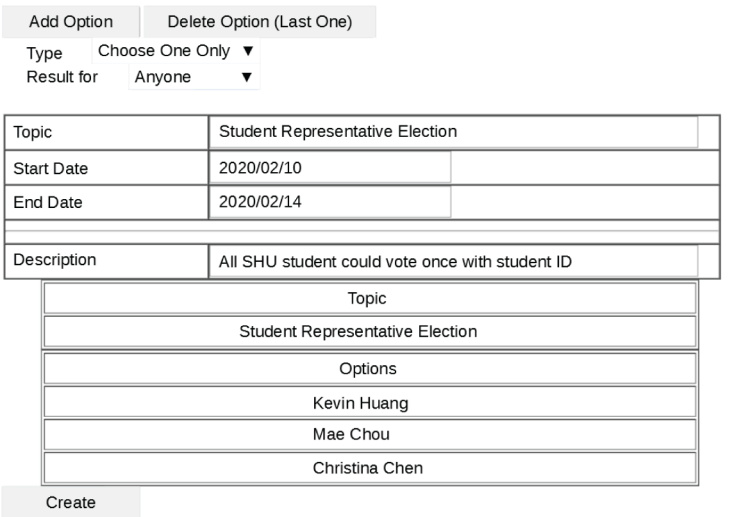
First, the smart contract is created using Solidity, the programming language of Ethereum. Then, it is deployed to operate the ballot on the ballot blockchain. The average time of the deployed smart contract is 38 seconds in four tests. The smart contract executes the tasks of verifying the users' accounts and passwords, confirming the users' qualifications, creating and recording the ballot, and determining the statistical results. All transactions are recorded and can be checked by anyone in the

ballot blockchain. However, all transaction content is encrypted as a hash using SHA-256; thus, the content is difficult to convert to the original message. Moreover, asymmetric cryptography is also adopted in the blockchain for data security. It is constructed by a pair of public and private keys. The private key consists of 64 strings of hexadecimal numbers. The message can only be decrypted using the exact pair of public and private keys. In this study, the ballot blockchain is built on the Ethereum platform, and the public key is open to everyone. A message can be encrypted using the receiver's public key and the sender's private key, and it is decrypted by the sender's public key and receiver's private key. Thus, the transaction transmission is secure in the ballot blockchain.

In the setting interface of the ballot parameter, the ballot type, result display, ballot topic, duration, description, and option settings are provided for the ballot creator. The ballot type includes a) choosing only one and b) choosing more than one for selection shown in Figure 3.

One of the results is set for anyone, which means everyone could check the results after the ballots

Figure 3. Interface of the Ballot Create

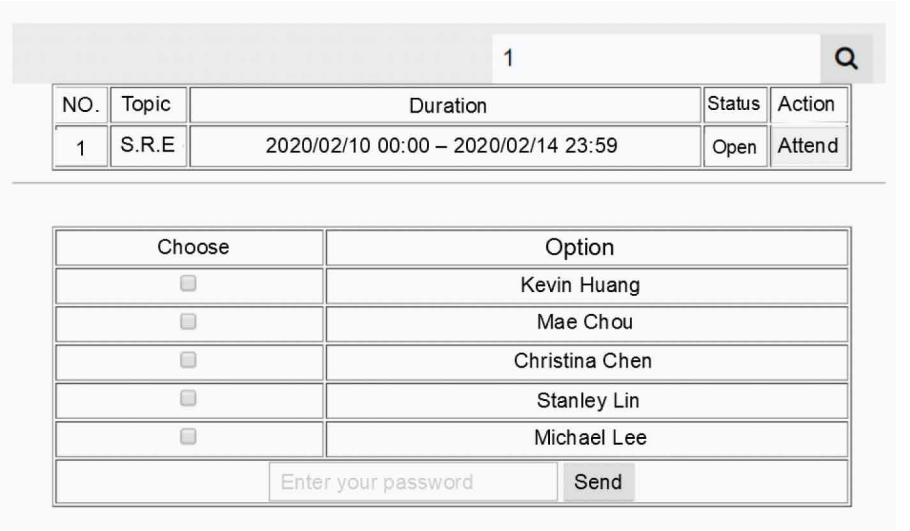


are finished, and the other is set for voters only. The ballot creator can set a topic, start date, end date, and description by filling in the blanks and can add or delete options using the related buttons. After completing the ballot parameter settings, the smart contract is triggered by clicking the 'Create' button. In the meantime, MetaMask requests the authority from the ballot creator to operate the methods of a smart contract and pay the ballot creation fee using ETH. The smart contract replaces the majority of human tasks in a traditional ballot, including verifying the user ID, checking the rule settings, creating the ballot, and sending the ballot to the blockchain. When the transaction is completed, the status and information of the transaction can be queried in MetaMask, and the ballot is visible to search. The average time of the transaction procedure of the ballot creation took around 59.75 seconds in the 12 tests.

In the interface of the secret ballot, the ballot search bar, information, and options are displayed. First, voters search for the ballot by typing the ballot number in the ballot search bar of upper-right corner, and then the related information is shown on the webpage if that ballot exists. The related options appear below the ballot after clicking the 'Attend' button shown in Figure 4.

Voters can select a ballot according to the ballot setting, either choosing only one or choosing more than one option and then clicking the 'Send' button if the ballot is open for everyone. If it is a private

Figure 4. Interface of the Secret Ballot



ballot, voters should enter the password, which is requested by ballot creator and then send to voters email for MetaMask. Each voter would receive the unique password, the password invalid once password has used. Afterward, the smart contract is triggered by clicking the 'Send' button (Figure 4). The ballot is sent to the functions of the smart contract, which verify the user's address, password, and whether the voter has already voted. In addition, the smart contract operates the ballot calculation and statistics afterward. The ballot is sent to the ballot blockchain when all procedures are confirmed by the smart contract. In the meantime, the MetaMask operates the ETH payment of sending the ballot if verified, and the users can check the transaction status. The average time of the transaction procedure of a secret ballot took around 45.7 seconds in the 10 tests. The duration of the transaction procedure depends on the testing network operating status, ballot type, execution period, and number of ballot options.

CONCLUSION

In this study, the ballot blockchain environment is built on solving the traditional ballot problem of time consumption, ballot locations, and traffic issues, reducing the number of manual tasks and repeated procedures. The ballot interfaces are developed for both secret ballots and ballots created through the integrated technologies of MetaMask and Remix. A smart contract is created to replace the procedures of human resources, including user ID verification, ballot dispatch, statistics, and announcement. The user information is not disclosed or recorded after verification, and all ballot content is hashed using an SHA and is stored in the ballot blockchain. The experimental results reveal the procedures for the secret ballot and ballot creation in practice. Furthermore, the transaction time in both actions takes less than a minute.

One of the limitations of the study is the verification method for the ID serial number. Currently, the ballot blockchain mechanism design is not connected to the government; thus, the verification method is limited to a textual combination without considering the validation. The other limitation is that all users could attend any ballot; therefore, the ballot may have irrelevant people attending. The former limitation might be solved when the platform links with the related organization, and the last limitation could be solved by considering a second verification in the ballot.

REFERENCES

Abe, R., Watanabe, H., Ohashi, S., Fujimura, S., & Nakadaira, A. (2018). *Storage Protocol for Securing Blockchain Transparency.* 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan. doi:10.1109/COMPSAC.2018.10298

Almashaqbeh, G., Bishop, A., & Cappos, J. (2019). ABC: A cryptocurrency-focused threat modeling framework. *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. doi:10.1109/INFCOMW.2019.8845101

Antonopoulos, A. (2017). *Mastering Bitcoin*. O'Reilly Media.

Clincy, V., & Shahriar, H. (2019). *Blockchain Development Platform Comparison*. 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Milwaukee, WI, USA. doi:10.1109/COMPSAC.2019.00142

Dhumwad, S., Sukhadeve, M., Naik, C., Manjunath, K. N., & Prabhu, S. (2017). *A Peer to Peer Money Transfer Using SHA256 and Merkle Tree.* 2017 23RD Annual International Conference in Advanced Computing and Communications (ADCOM), Bangalore, India. doi:10.1109/ADCOM.2017.00013

Dimitri, V. L., Sion, L., Vandeloo, E., & Joosen, W. (2019). *On the Applicability of Security and Privacy Threat Modeling for Blockchain Applications. Computer Security*.

Fromm, K. (2017). *How Blockchain and serverless processing fit together to impact the next wave*. https://read.acloud.guru/blockchain-and-serverless-processing-similarities-differences-and-how-they-fit-together-c12142373287

Fu, M. (2019). Ballot Mechanism Design Based on Blockchain Methodologies. *International Conference on Computing and Big Data*. doi:10.1145/3366650.3366656

Gartner. (2020). *Smart With Gartner, Gartner Top 10 Strategic Technology Trends for 2020*. https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2020/

Hebert, C., & Cerbo, F. D. (2019). Secure blockchain in the enterprise: A methodology. *Pervasive and Mobile Computing*, *59*, 59. doi:10.1016/j.pmcj.2019.101038

Košťál, K., Krupa, T., Gembec, M., Vereš, I., Ries, M., & Kotuliak, I. (2018). *On Transition between PoW and PoS*. 2018 International Symposium ELMAR, Zadar. doi:10.23919/ELMAR.2018.8534642

Mohanta, B., Panda, S., & Jena, D. (2018). An Overview of Smart Contract and Use Cases in Blockchain Technology. *International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. doi:10.1109/ICCCNT.2018.8494045

Nakamoto, S. (2018). *Bitcoin: A Peer-to-Peer Electronic Cash System*. https://bitcoin.org/bitcoin.pdf

Nguyen, V., Pham, H., Tran, T., Huynh, H., & Nakashima, Y. (2019). *Digitizing Invoice and Managing VAT Payment Using Blockchain Smart Contract*. 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Seoul, South Korea. doi:10.1109/BLOC.2019.8751256

Ogawa, T., Kima, H., & Miyaho, N. (2018). *Proposal of Proof-of-Lucky-Id(PoL) to Solve the Problems of PoW and PoS*. 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada.

Shukla, S., Thasmiya, A. N., Shashank, D. O., & Mamatha, H. R. (2018). *Online Voting Application Using Ethereum Blockchain*. *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Bangalore, India.

Silvestre, M. L. D., Gallo, P., Sanseverino, E. R., Sciumè, G., & Zizzo, G. (2019). *A new architecture for Smart Contracts definition in Demand Response Programs*. 2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), Genova, Italy. doi:10.1109/EEEIC.2019.8783960

Siva Sankar, L., Sindhu, M., & Sethumadhavan, M. (2017). *Survey of consensus protocols on blockchain applications*. 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India. doi:10.1109/ICACCS.2017.8014672

Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X., & Wang, F. (2019). Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends. *IEEE Transactions on Systems, Man, and Cybernetics. Systems*, *49*(11), 2266–2277. doi:10.1109/TSMC.2019.2895123

Wang, S., Zhang, Y., & Zhang, Y. (2018). A Blockchain-Based Framework for Data Sharing With Fine-Grained Access Control in Decentralized Storage Systems. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 38437–38450. doi:10.1109/ACCESS.2018.2851611

Zhang, R., Xue, R., & Liu, L. (2019). Security and Privacy on Blockchain. *ACM Computing Surveys*, *52*(3), 1–34. doi:10.1145/3316481

Zinca, D., & Negrean, V. (2018). Development of a Road Tax Payment Application using the Ethereum Platform. *2018 International Symposium on Electronics and Telecommunications (ISETC)*, Timisoara, Romania. doi:10.1109/ISETC.2018.8583975