The Journal of Supercomputing (2021) 77:11881–11893
https://doi.org/10.1007/s11227-021-03728-1

Implementation of real estate contract system using zero knowledge proof algorithm based blockchain

SoonHyeong Jeong[1] · **Byeongtae Ahn**[2]

Accepted: 7 March 2021 / Published online: 30 March 2021

Abstract

A real estate contract is a high-risk contract with a large amount of money, and there are many problems in the risk and reliability of fraud. In particular, online contracts have low reliability and high risk. If you can manage information on these issues and prevent forgery and duplication of contract information, you can maintain low cost and high efficiency. Blockchain technology is being used as an alternative to the enormous cost and reliability issues associated with offline real estate contracts. When blockchain technology is applied to an online contract management system, reliability and scalability are efficient and confidentiality can be ensured. However, as real estate transactions using blockchain technology increase online, problems arise in scalability. Therefore, in this paper, we implemented an Ethereum-based online real estate contract system that solved the scalability using the zero-knowledge proof algorithm. The real estate contract system enables online contract management and discrimination of contract forgery through blockchain. In particular, it was possible to prevent fraud until the contract was concluded and the contract was terminated.

Keywords Blockchain · Zero knowledge · Ethereum · Smart contract · Real estate

✉ Byeongtae Ahn
ahnbt@anyang.ac.kr

SoonHyeong Jeong
kevin.j@onther.io

[1] Onther Inc, 527, Gangnam-daero, Seocho-gu, Seoul, Republic of Korea

[2] Liberal & Arts College, Anyang University, 22, 37-Beongil, Samdeok-Ro, Manan-Gu, yang430-714, Anyang, South Korea

1 Introduction

In the world, the number of transactions in the real estate market is increasing every year, but the damage and fraud of real estate transactions are constantly increasing. In order to prevent contract fraud, contracts are executed according to procedures, but various fraud cases have occurred by exploiting loopholes in the process. In real estate contracts, it is common practice to enter into contracts with third-party brokers to prevent fraud. However, fraud occurs frequently due to the lack of credibility of third-party brokers. In addition, other fraud cases frequently occur in duplicate contracts that are not lacking third parties [1]. And when a transaction is made through a third-party broker, a high commission is incurred.

In order to solve these problems, a technical system that prevents forgery of contracts and guarantees trust in advance is needed. Therefore, in this paper, a P2P trust-based network using blockchain technology that guarantees 100% reliability has been implemented to implement a system that can securely establish real estate contracts without intermediaries. The advantages of blockchain technology are very high reliability and safety, but there are many problems with scalability [2]. In particular, as the number of contract transactions increases, a problem occurs in that the transaction throughput decreases. This system solves the scalability problem by improving the transaction throughput by using the zero-knowledge proof algorithm. In addition, the forgery problem was solved by creating a contract using smart contracts on the basis of Ethereum, a distributed computing platform. This system provides the function for contract management for contract related services such as real estate brokerage and stock brokerage by providing an API for managing contracts as an Open API. Brokerages utilizing this platform's API can handle secure and reliable contract management online at low cost by paying a small fee to the system. Therefore, in this paper, a real estate contract platform was developed using a virtual machine with improved scalability. It is the first real estate contract platform with improved reliability and scalability compared to existing real estate contracts.

2 Related research

In Sect. 2, we will look at the zero-knowledge proof algorithm mentioned in Introduction and examine the blockchain-related studies to develop a blockchain-based system.

2.1 Zero-knowledge proof algorithm

Blockchain technology can be divided into a simple type of blockchain made of UTXO (Unspent Transaction Output) and a complex type of blockchain that deals with the State Tree [3]. Currently, in the simple form of blockchain, zero-knowledge proof is used at the protocol level only in some transaction processing. However, although some complex forms of blockchain use smart contracts using smart contracts, there are limitations in terms of performance and utilization because they are

implemented in the upper layer. Proof size of a single operation created through the proposed SNARKs algorithm is about 1,500 bytes (1.5 kbytes) [4].

* Bullet Proof algorithm.

-Transaction size of UTXO-based blockchain = in * 254 * 146 + out * 254 * 33 + 10.

And it increases arithmetically according to the number of * in, out used.

* It occupies about 45,000 bytes (45kbytes) based on 1 in and 1 out.

-Regardless of the type of transaction, the transaction size can be fixed to 1.5 kbytes, and even the simplest transaction standard is more than 70% economical [5].

-The blockchain-based distributed application market is expected to grow from about $ 3.2 billion in 2019 to more than $ 60 billion in 2024 (Blockchain Market Shares, Market Strategies, and Market Forecasts, 2018 to 2024, IBM, 2018).

Among them, the market with 'transaction processing' as a profit model is expected to reach 55% of the total [6].

Since such a blockchain-based distributed application is generally provided on the basis of open source, transaction fees rather than content usage fees are inevitably accepted by users. Therefore, the economic value of the technology to efficiently process transactions is very positive. Even if all verification nodes do not participate in block verification, the general operation is verified with the same security strength as all nodes participated and verified using zero-knowledge proof technology, thereby providing the same effect as saving the entire transaction without saving all transaction data [7].

Currently, as the value of using personal information increases, discussions on how to provide personal information have been actively conducted. Currently, one of the most common methods of providing personal information is a group that uses personal information to obtain personal consent and use personal information [3]. However, the above method has two problems. First, information that is more than the information required by the institution for personal information is being exposed. Second, whenever a company requests personal information, there is a problem that a trusted party must provide authentication information for the information to the company. In order to solve the above problems, this paper proposes a privacy-protected personal information management method using zk-SNARK (zero-knowledge Succinct Non-interactive ARgument of Knowledge) technique and blockchain [8]. zk-SNARK is a modification of the existing ZKP to be more succinct and applicable in a non-interactive environment. This logic was first proposed in 2012, and due to its characteristics, ZKP can be implemented in a blockchain environment. In the case of a blockchain transaction using zk-SNARK, the validity of the transaction can be communicated to nodes other than the sending and receiving node without exposing information such as a receiver, a sender, and a transfer amount [9]. ZCash is the first application of zk-SNARK, and related contents were applied to Ethereum's Byzantium hard fork [10]. The zk-SNARK is largely divided into two parts, one is the process of converting the problem to be proved to a specific form, and the other is the process of actual proofing using the converted problem [7]. The privacy-protected personal information management technique can guarantee the privacy and reliability of information when providing personal information through zk-SNARK. In addition, it is possible to manage personal information data while ensuring the

integrity of the data through the blockchain, and sharing personal information can be performed more easily than the existing authentication method [8]. The privacy-protected personal information management technique can guarantee the privacy while ensuring privacy when providing personal information through zk-SNARK. In addition, it is possible to manage personal information data while ensuring the integrity of the data through the blockchain, and sharing personal information can be performed more easily than the existing authentication method [9].

2.2 Algorithm for transaction validation

Proof of zero knowledge must satisfy the following three conditions.

* completeness: If a condition is true, a trusted verifier must be able to understand this by a trusted prover.

* soundness: When a condition is false, a dishonest verifier can never convince the verifier that the condition is true by lying.

* zero-knowledge: When a condition is true, the verifier knows nothing other than the fact that this condition is true.

The study intends to utilize zero-knowledge proofs for various types of transactions that the user wants, as well as predefined types of transactions. Circuits that can produce evidence of current zero-knowledge proofs can only perform operations in a predefined form. In order to be able to utilize this in various types of transactions desired by users, a circuit capable of verifying general operations is required. General operation means universal and various operations, not specific predefined operations. Therefore, the research team researched a circuit capable of verifying general operations and designed the method to apply it to the virtual machine. In addition, by using the zero-knowledge proof technology, even if a blockchain participant does not know the contents of the block, it can quickly verify that the contents of the block are not forged or tampered with by the node performing the proof and reporting role among all nodes. Also, by rapidly increasing the block sync speed for participants, new participants can quickly join the network.

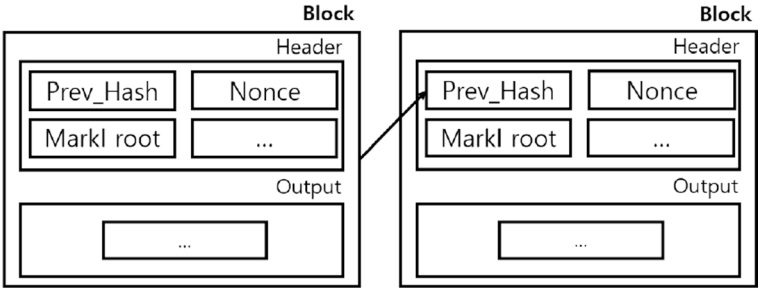

Fig. 1 Block head architecture in blockchain

2.3 Ethereum

In Sect. 2.2, let us look at the blockchain-related studies for developing the blockchain-based system mentioned in Introduction and then look at what systems are going back to the blockchain [10].

Blockchain's Ethereum uses a Proof of Work algorithm [7]. PoW is an algorithm used to acquire a bitcoin. When a computer presents a difficult math problem, all nodes participating in the chain solve the problem and discover random numbers. The node that finds a random number is successfully transmitted and has a structure that is connected by the hash value with the previous block as shown in Fig. 1.

In addition, the node is provided with a certain bitcoin, and this is called mining. In addition, the transmission is performed by two methods, the first being the abovementioned mining, and the second is when the bitcoin is sent to another user. At this time, the output section contains information on which users are sending and how much money is sent [11].

Smart Contract is a concept first proposed by Nick Szabo in 1994, and the existing contract is in writing, so a person must perform it manually to fulfill the terms of the contract, but the digital contract is automatically collected according to the terms. Blockchain creates trust of digital data through a method in which multiple nodes verify data and share it among nodes based on the verified ones. Based on reliable data, smart contracts have become an issue with blockchain [12].

The block chain operation structure using the smart contract method is as follows. Assuming that there are buyers and sellers who want to purchase the goods, the seller registers the goods to be sold on the blockchain. At this time, the smart contract registers a new node in the blockchain according to the transaction and updates it in real time. The buyer can search the relevant blockchain using Query, and if a product is purchased, the smart contract updates the database of the corresponding node.

Ethereum is a programmable blockchain that can be added by developing various blockchain services. Ethereum is a distributed network platform that can be operated as a client program [13].

The currency unit of Ethereum is ether, which is used as a transaction fee. In Ethereum, programming is not required in Gold Dragon, but it is supported for

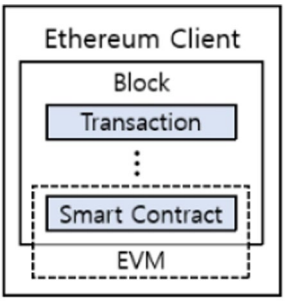

Fig. 2 Smart contract runtime environment in ethereum

smart contracts. It is delivered in the form of a transaction (financial transaction) or contract (nonfinancial transaction) between the sender and the receiver, and the ether, which is a fee, is also delivered to protect the system when delivering. There are two types of currency: ether for general transactions, finney for small settlements, szabo for payment of fees in transactions, and wei [14].

Figure 2 shows the process of running smart contracts on the Ethereum base. The contract to be developed is included in the block and transmitted to other blocks in the blockchain to be executed during verification. In order to execute the smart contract, it is driven through the Ethereum Virtual Machine (EVM) [15].

3 Design of zero-knowledge circuit

Due to the nature of the blockchain that stores all transaction data, the data on the blockchain continue to increase over time. When zero-knowledge proof technology is applied to storage of transaction data, data storage space can be saved by compressing the data by pruning actual data and leaving only proof of data. As time goes by, the data of the blockchain will gradually accumulate, and accordingly, the computing resources required to operate the full node will gradually increase [16].

In the case of Ethereum, it is already difficult for an individual to operate a full node, and in the future, it is expected that only large companies or large hands that can have sufficient computing resources can operate the full node. This will cause the centralization of the blockchain, and this problem can be solved by reducing the resources required for data storage and verification through a virtual machine with zero knowledge authentication technology.

Figure 3 is designed to apply the zero-knowledge proof algorithm to the virtual machine. It shows the flow of the operation method of the Ethereum virtual machine for applying zero-knowledge proof technology. Since Solidity, the smart contract language of Ethereum, is a language created for human understanding, it needs to be changed to a machine language understandable by a virtual machine in order to operate in a virtual machine. Code written in Solidity is converted to Ethereum bytecode by the compiler. This bytecode is executed by EVM, Ethereum's virtual machine. When a specific bytecode is executed, all nodes in the Ethereum network execute the same bytecode, respectively, to verify the transaction. At this time, if

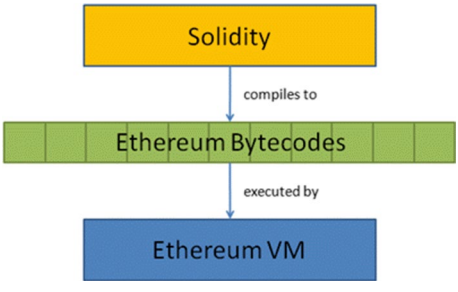

Fig. 3 Execution process of ethereum virtual machine

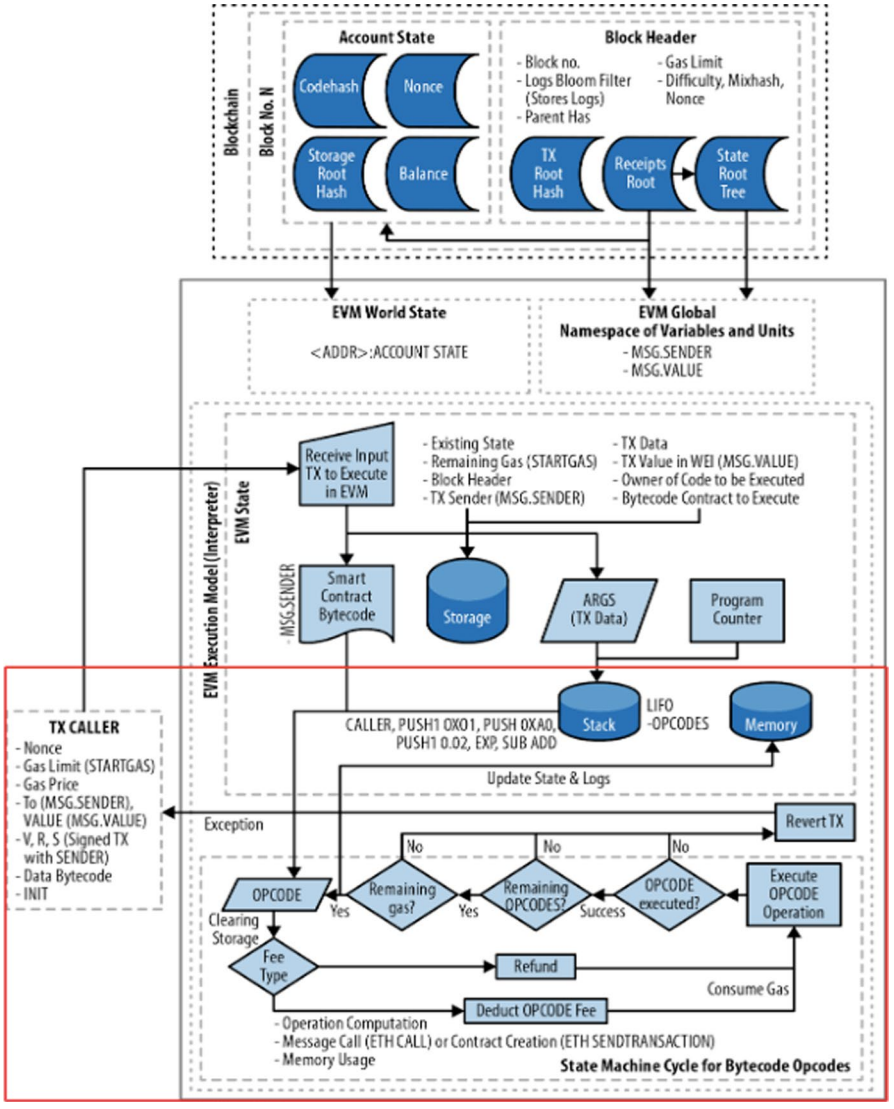

Fig. 4 Architecture and execution flowchart of ethereum virtual machine

zero-knowledge proof technology that can perform general operation verification is applied to the virtual machine, even if the virtual machine does not execute the transaction, it is possible to know whether the corresponding transaction is the correct transaction by performing verification on the zero-knowledge evidence [17].

In order to modify the virtual machine, it is necessary to understand the structure. Therefore, Fig. 4 shows the architecture and execution flow diagram of the Ethereum virtual machine. Once you understand how the virtual machine is running,

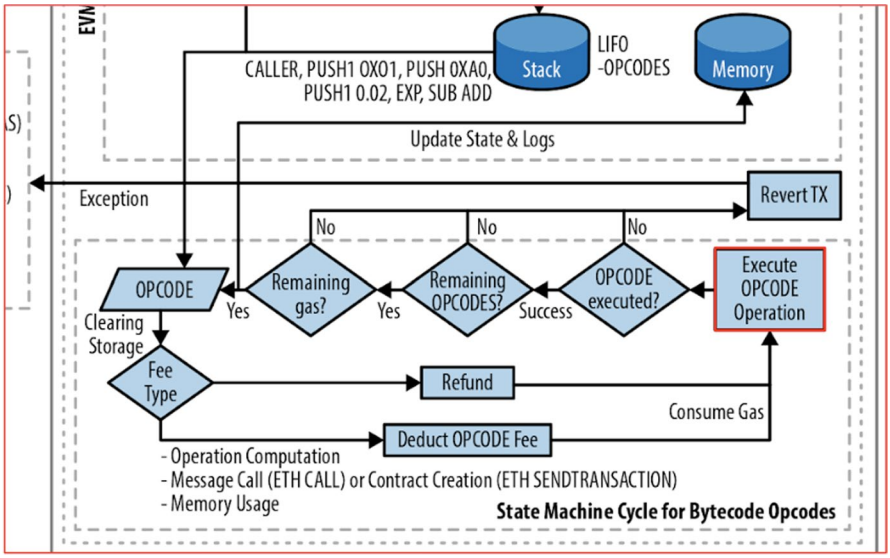

Fig. 5 Change part in virtual machine

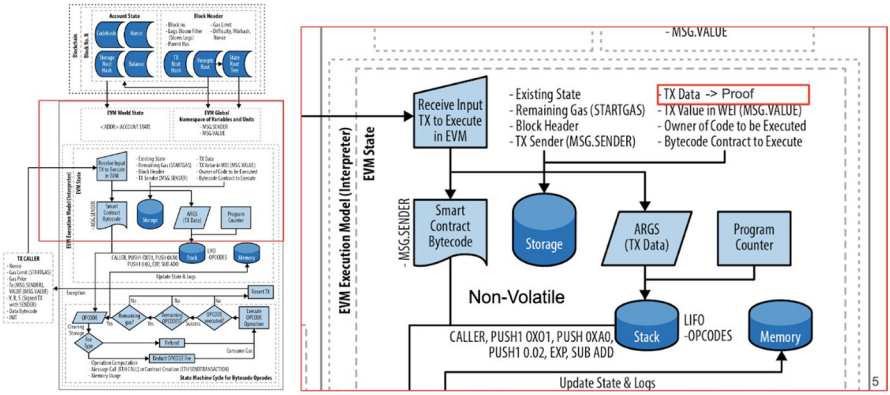

Fig. 6 Data changes after zero-knowledge proof technology

you need to figure out what parts of the virtual machine need to be modified to apply zero-knowledge techniques.

In order to execute the transaction, it needs to be changed to Ethereum bytecode as mentioned. These bytecodes are decomposed into what are called opcodes, stacked on the stack and executed one by one. You must subtract the gas cost for running the virtual machine before the opcode runs. The opcodes are now executed if the gas cost is not insufficient. Figure 5 shows the parts that need to be changed in the virtual machine when the opcodes are executed.

In the figure above, the part marked with a red box is the part to which zero-knowledge proof technology should be applied, and the part to create a universal circuit that can execute the opcode. Figure 6 shows the change in data stored after the zero-knowledge proof technology is applied. After the zero-knowledge proof technology is applied to the part that performs the opcode, TX data among the data stored in the existing storage are replaced with the proof of the zero-knowledge proof. And we will create and test a virtual machine with zero-knowledge proof.

4 Design of zero-knowledge circuit

The real estate transaction management system manages the contracts of transactions online that have no choice but to include offline procedures. This system has two advantages. First, it prevents forgery and alteration of the contract. In this paper, blockchain is used to store and manage contract information and contracts. When storing and managing contracts in a blockchain, forgery of the contracts is impossible because the contracts are stored on the trust-based network. Second, it prevents duplicate contracts. When a duplicate contract is executed for an item that has already been contracted, the contract is filtered in advance. However, the item to be contracted must be registered in this system and given an ID [3].

This system is composed of five modules for each function (USER, CONTRACT, FILE, TRANSACTION, MAILING). The USER module is classified into createUser, getUser, log-in, modifyUser, and deleteUser. The CONTRACT module is classified into addContract and findContract. The FILE module and the MAILING

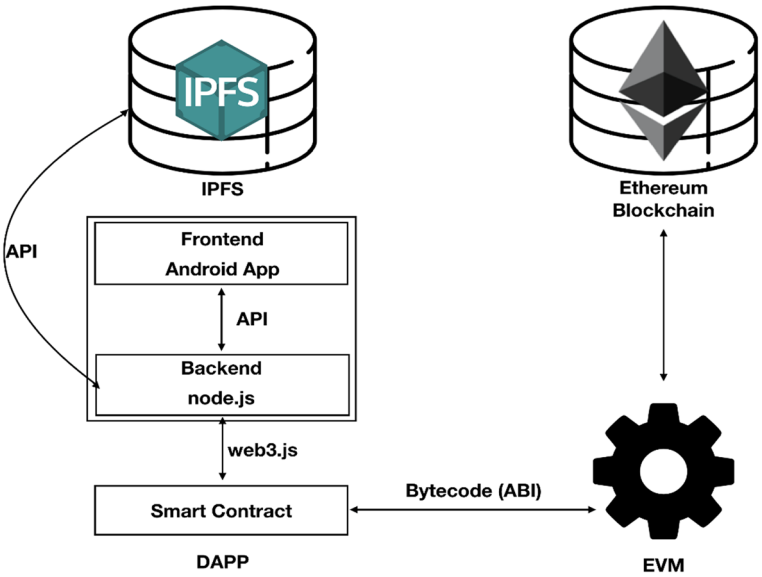

Fig. 7 System architecture

module are composed independently. And the TRANSACTION module is classified into createTransaction, getSenderTransaction, getRecipientTransaction, acceptTransaction. The API was designed and modeled for each module [8].

The configuration diagram of this system is made to operate using the blockchain-based IPFS storage method. Figure 7 shows the configuration of this system.

This system uses a node.js server to maximize system performance and uses Ethereum to support a blockchain-based distributed computing network. For storage of smart contract documents, IPFS, a dedicated storage system for blockchain, was used, and document access was made possible using the API provided by IPFS [9]. I used web3.Js to move documents in IPFS. In the Ethereum virtual machine, it was made to move with the smart contract in byte code. Since this system manages contracts through blockchain, it manages contracts safely and reliably online. Registering a transaction on this platform to manage the contract does not have the same legal effect as reporting to an institution, but just being managed on this platform can have a notary effect. Therefore, in the case of a trading company utilizing the system, it can be proved that the customer's trading company makes the contract transparent to customers who intend to use the trading company. And, from the customer's point of view, it is possible to prevent fraud by contracting with a company that manages contracts in this system.

5 Implementation of real estate transaction

The operating system of this system runs on mocOS and Linux, and intelliJ and Atom are used as a smart contract integrated development environment. Android Studio was used as the Android app development environment. As a backend environment, node.js, which was stable and minimized overload, was used. And for

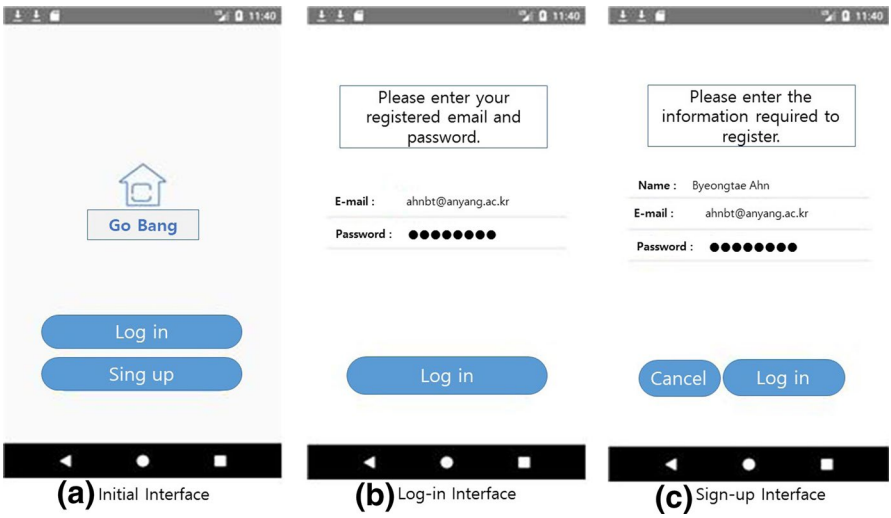

Fig. 8 Implementation initial interface using smartphone

smooth data processing, memory DB mongo DB was used, and web3.js was used for data processing on the web. Java and JavaScript languages were used as development languages, and Solidity was used as an Ethereum-based token development language. We used AWS to build a stable server.

Ethereum was used to implement a blockchain-based distributed computing platform capable of executing DApps and smart contracts. Ethereum is an EVM operating system on a peer-to-peer network node, providing a mechanism based on cryptocurrency and operational tokens. The contract of this system is created as a pdf file and stored in IPFS as a hash value. When uploading an existing contract to a smart contract, high cost occurs and processing speed is delayed, so only the hash value of the pdf file is stored in the smart contract using IPFS.

Figure 8 shows the initial screen by accessing the system using a smart phone. (a) When the system is implemented, the initial screen appears. Clicking the log-in button in Figure (a) starts Figure (b). If you are registered as a member, enter your email and password and click the log-in button to go to the main screen. If this is your first visit, you must register by clicking the sign-up button. Figure (c) is the initial screen where the member registration button is pressed. Enter the name, email, password, and click the log-in button to register.

In Fig. 9, (a) is the main screen that shows the contract registration and the list of registered contracts after logging in. You can create a new contract by clicking on the contract registration button. (b) shows the process of registering a new contract and actually registering it. (c) The contract registration is completed and the contract.pdf document is generated and the registration date is provided.

In Fig. 10, (a) shows the process of contract.pdf document being stored in the actual registration repository when the send button is pressed after registering the contract. When the final document is stored in the repository, return to the main screen to receive a new contract. (b) shows the contract.pdf document where the

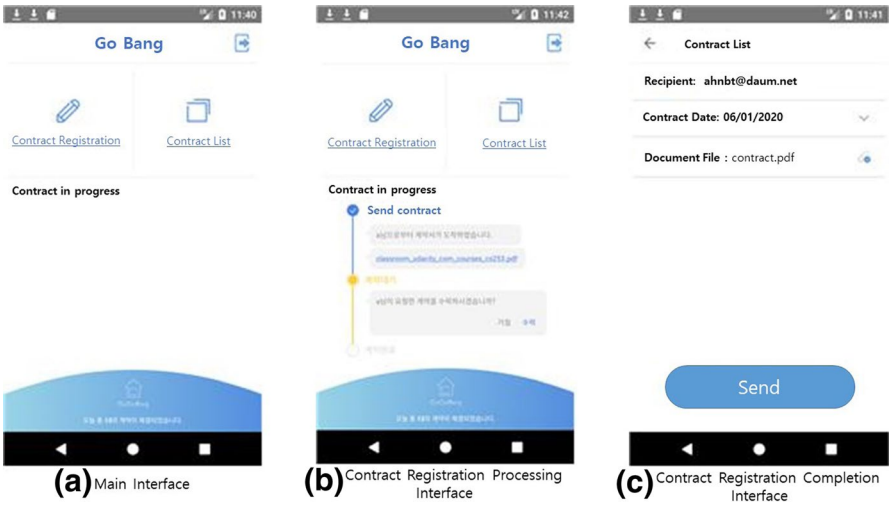

Fig. 9 Implementation interface in smartphone

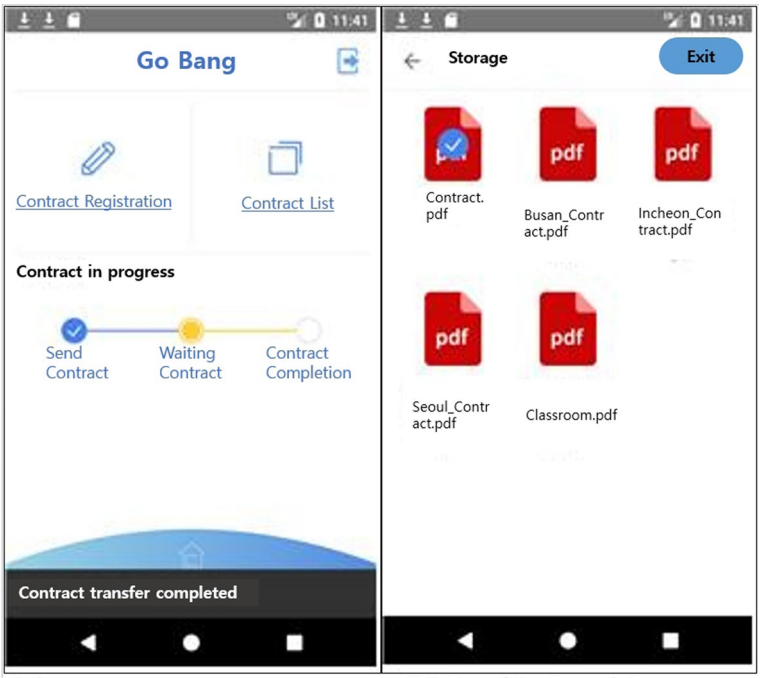

Fig. 10 Contract registration interface in smartphone

actual document is stored in the repository. And press Exit button to go to the main screen.

6 Conclusion

In this paper, the transaction processing rate in real estate contracts is improved by using the zero-knowledge proof algorithm. In addition, a real estate transaction system using blockchain technology has been implemented to falsify and prevent false documents. This system developed the system using Ethereum among public blockchains to strengthen the prevention of duplication of existing real estate contracts. This system function enhances reliability and security in contract transactions that require offline procedures such as real estate contracts and stock contracts. And since the contract is managed on the P2P trust-based network, transparency is guaranteed. Lastly, a reduction in fees occurs significantly due to the exclusion of third-party certification agencies, and various services for electronic transactions through the platform are available. As future tasks, it is necessary to develop various DApps and expand and develop platforms. In addition, a zero-knowledge proof algorithm for improving transaction processing efficiency should be additionally developed.

Acknowledgement This study was supported by the Institute for Information & communication Technology Planning & evaluation grant funded by the Korea Ministry of Science and ICT(No. 2020-0-00105).

References

1. Park HS, Chung JW, Kim UM (2017) A study on shared EMR (Electronic Medical Record By BlockChain (Ethereum). In: Proceedings of KIIT Summer Conference, pp 436–437
2. Ko YS, Choi HS (2017) Changing business paradigm and its application—focused on the block chain technology, Korea Science & Art Forum. 27 https://doi.org/10.17548/ksaf.2017.01.27.13
3. Ben-Sasson E, Chiesa A, Tromer E, Virza M (2013) Succinct Non-Interactive Zero knowledge for a von neumann architecture. https://eprint.iacr.org/2013/879.pdf
4. Ben-Sasson E, Chiesa A, Genkin D, Tromer E (2019) Fast reductions from RAMs to delegatablesuccinct constraint satisfaction problems. In: Proceedings of the 4th Innovations in Theoretical Computer Science Conference, ITCS '13, pp 401–414
5. Capellán RU, Ollero JS, Pozo AG (2021) The influence of the realestate investment trust in the real-estate sector on the Costa del Sol. Eur Res Manag Bus Econ. https://doi.org/10.1016/j.iedeen.2020.10.003
6. B-Sasson E, Chiesa A, Tromer E, Virza M (2018) Scalable zero knowledge via cycles of elliptic curves (extended version) In: Advances in Cryptology—CRYPTO, 8617
7. Bowe S, Grigg J, Hopwood D (2019) Halo: recursive proof composition without a trusted setup. https://eprint.iacr.org/2019/1021.pdf
8. Ben-Sasson E, Chiesa A, Genkin D, Tromer E (2013) Fast reductions from RAMs to delegatablesuccinct constraint satisfaction problems. In: Proceedings of the 4th Innovations in Theoretical Computer Science Conference, ITCS '13, pp 401–414
9. Valiant P (2008) Incrementally verifiable computation or proof of knowledge imply time/space efficiency. In: Canetti R (ed) Theory of Cryptography. Springer, Berlin, Heidelberg, pp 1–18
10. Sasson EB, Chiesa A, Tromer E, Virza M (2014) Scalable zero knowledge via cycles of elliptic curves (extended version). Advances in Cryptology - CRYPTO 79(4):1102–11602
11. Agrawal S, Ganesh C, Mohassel P (2018) Noninteractive zero-knowledge proofs for composite statements. In: Annual International Cryptology Conference, pp 643–673
12. Bünz B, Bootle J, Boneh D, Poelstra A, Wuille P, Maxwell G (2018) Bulletproofs: short proofs for confidential transactions and more in Bulletproofs: Short Proofs for Confidential Transactions and More. IEEE, pp 315–334
13. Katz J, Koilesnikov V, Wang X (2019) Improved non-interactive zero knowledge with applications to post-quantum signatures. University of Maryland and Georgia Tech, pp 525-537
14. Sah CP, Jha K, Nepal S (2016) Zero-knowledge proofs technique using integer factorization for analyzing robustness in cryptography. In: Proceedings of the 10th INDIACom; INDIACom-2016 3rd International Conference on Computing for Sustainable Global Development, pp 638–642
15. Jiang S et al (2018) Blochie: a blockchain-based platform for healthcare information exchange. In: 2018 IEEE International Conference on Smart Computing (SMARTCOMP), pp 49–56
16. He X, Alqahtani S, Gamble R (2018) Toward privacy-assured health insurance claims. In: 2018 IEEE International Conference on Internet of Things (iThings), pp 1634–1641
17. Rose JD, Snowden KA (2013) The new deal and the origins of the modern American real estate loan contract. Explor Econ Hist 50(4):548–566