경영학 석사학위 논문

Blockchain based resource management model for civil documents

블록체인 기반 민원 문서 관리 모델

2019년 2월

서울대학교 대학원

경영학과 경영학 전공

정 한 비

Blockchain based resource management model

for civil documents

지도교수 박진수

이 논문을 경영학 석사학위 논문으로 제출함

2018년 12월

서울대학교 대학원

경영학과 경영정보 전공

정한비의 석사학위 논문을 인준함

2018년 12월

Chair 장 정 주 (Seal)

Vice Chair 유 병 준 (Seal)

Examiner 박 진 수 (Seal)

Abstract

Blockchain based resource management model for civil documents

Hanbi Jeong

Management Information System

The Graduate School of Business, Seoul National University

With the advent of Information and Communication Technology (ICT), e-governance becomes an essential part of the government. Among the services provided by the Korean government, Minwon24, which issues civil documents, is the most used one. However, it has some limitations, namely: (1) for the documents that are impossible to get, people should go to a government agency which is open only from 9 a.m. to 6 p.m.; (2) it provides a checking authenticity service.; (3) people cannot know what happened even if the agency handles the documents arbitrarily.

To address the issues outlined above, Blockchain can be an alternative. Put simply, Blockchain can be defined as “the distributed digital ledger”. It has a tremendous potential in that it has maximal transparency and low risk of being hacked. Resource management is one of the areas where Blockchain is frequently used.

In the present study, we suggest a new model based on Blockchain for Minwon24; the proposed model is a type of resource management. There are three

participants: Issuer, Owner, and Receiver. The proposed model has two stages: issuing and exchanging. Issuing is creating civil documents on the database, which is BigchainDB in our study. Exchanging, the next stage, is a transaction between the owner and the receiver. Based on this model, the actual program is conducted with the programming language Python. To evaluate our model, we use various criteria, such as privacy, security, and scalability, and compare some key properties of the proposed model to the existing model. It shows the excellence of our model compared to others in prior literature.

Keywords: Blockchain, e-governance, database, resource management

Student Number: 2017-20241

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

Blockchain can be simply explained as “the distributed digital ledger”. It has emerged with the advent of Bitcoin, which is the most typical cryptocurrency. The information in Blockchain cannot be fabricated or hacked. In addition, it makes it possible to achieve maximal transparency, as everyone involved can see all transactions that happened on Blockchain.

Due to the technological potential of Blockchain, it attracts considerable interest of various private and public sectors. Actually, it has extended its usability to the technological foundation of business, such as financial service, banking, trading, insurance, data protection, voting, intellectual property, identity authentication, leasing, and government service (Atzori, 2015; De Meijer, 2016; Fanning and Centers, 2016; Peters and Panayi, 2016; Swan, 2015; Trautman, 2016; Wall Street Journal [WSJ], 2015; Yermack, 2017; Zyskind et al., 2015). Public sector is one of them, and Blockchain is considered in it as the core technology of e-governance.

With the advent of Information and Communication Technology (ICT), e-governance has become an essential part of the government. E-governance, meaning ‘electronic governance’ is using information and communication technologies(ICTs) at various parts of the government and the public area and more, for the purpose of enhancing the whole government activities (Bedi, Singh and Srivastava, 2001; Holmes, 2001; Okot-Uma, 2000). Through e-governance, citizens can easily access civil documents and government serviced. In Korea, one of the most representative bodies of e-governance is Minwon24. User face of

Minwon24 is shown in Figure 1.

Figure 1. User face of Minwon24

Minwon 24 is an e-governance service which grants citizens access to civil documents without having to visit a government agency. Through Minwon24, citizens can get civil documents needed for various purposes, such as applying for a loan, dealing with real estate issues, or applying for a job. Minwon24 is useful to citizens who do not have time to get documents through government agencies that are typically open from 9 a.m. to 6 p.m., i.e. the usual working time for most people. Therefore, according to National Statistical Office, Minwon24 is the most used webpage of the entire government website (KOSIS, 2018; see Table 1 for relevant statistics). However, although Minwon24 allows citizens to easily access their documents, it has several limitations.

Table 1. Statistical results of using e-governance services

E-governance service	Percentage (%)
Minwon 24	68.4

Government agency	36.9
NEIS service (Education)	27.4
Bokjiro (Welfare)	18.1
E-people (Petition)	14.5
Open-info (Release of the information)	16.8

Minwon24 provides service for checking whether the documents are faked or not. If a person who receives documents wants to check whether it is authentic, it should be checked, and this checking process is cumbersome. In addition, even if the agency handles the documents arbitrarily like giving them to other agencies, citizen who has the ownership of documents do not know what happens. This limitation can be overcome with the Blockchain technology.

Through request of information disclosure, it is known that the five most frequently issued document types, land registration, building registration, resident registration, certificate of local tax payment, land(forest) registration, account for 89.9% of all documents requested for issuance from Minwon24, which indicates that approximately 90% of the tasks require a higher work efficiency. Therefore, the present study will focus on the aforementioned five most frequently issued documents.

The remainder of this paper is organized as follows. In Chapter 2, we review related works. In Chapter 3, our alternative for Minwon24 is explained. Then, the results are presented in Chapter 4. The evaluation of our model will be discussed in Chapter 5. Finally, in Chapter 6, the contributions of the present study are discussed and the limitations are acknowledged.

Chapter 2. Related Works

2.1. The basic concept of Blockchain

2.1.1. The definition of Blockchain

Blockchain appeared with the advent of Bitcoin. Bitcoin is one of the most famous cryptocurrencies and was introduced by Satoshi Nakamoto. Blockchain is the fundamental network for Bitcoin at first. It is needed to know how to deal with bitcoin.

Transactions occurring in bitcoins are quite different from those with traditional currencies (Nakamoto, 2008). It uses asymmetric key encryption. The parties participating in a transaction have two keys: the public key and the private key. These are paired keys, so the contents encrypted by one of them can only be decrypted with the other. Once the payer encrypts the content of a transaction with the payee's public key, the payee can open the content with his/her private key. With that encryption method, the content is protected against hacking. The procedure of asymmetric key encryption is depicted in Figure 2.

Also, the payer sends his/her digital signature to the payee. Digital signature is the hash encrypted with the payer's private key. Using it, the payee can verify the ownership of content. However, there is a risk that the payee cannot prove that the payer did not double-spend the money in the content. This issue can be solved by the proof of work which is explained in the next section. The procedure of digital signature is depicted in Figure 3.

Figure 2. Asymmetric key encryption

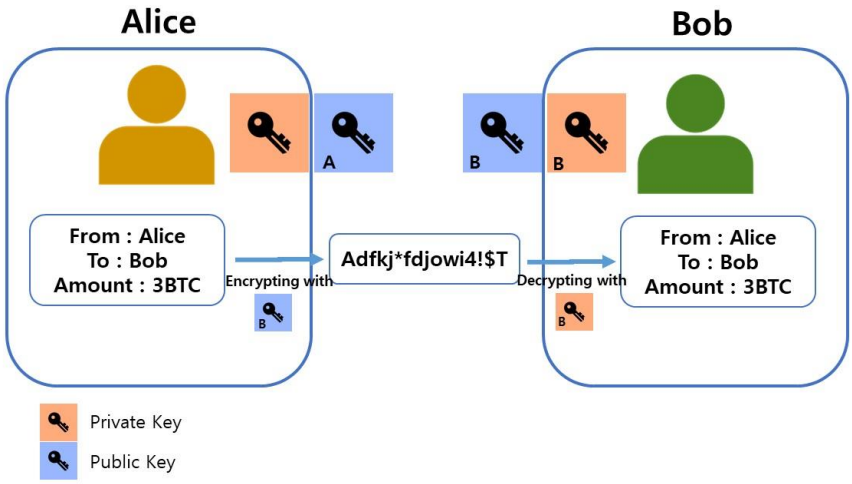
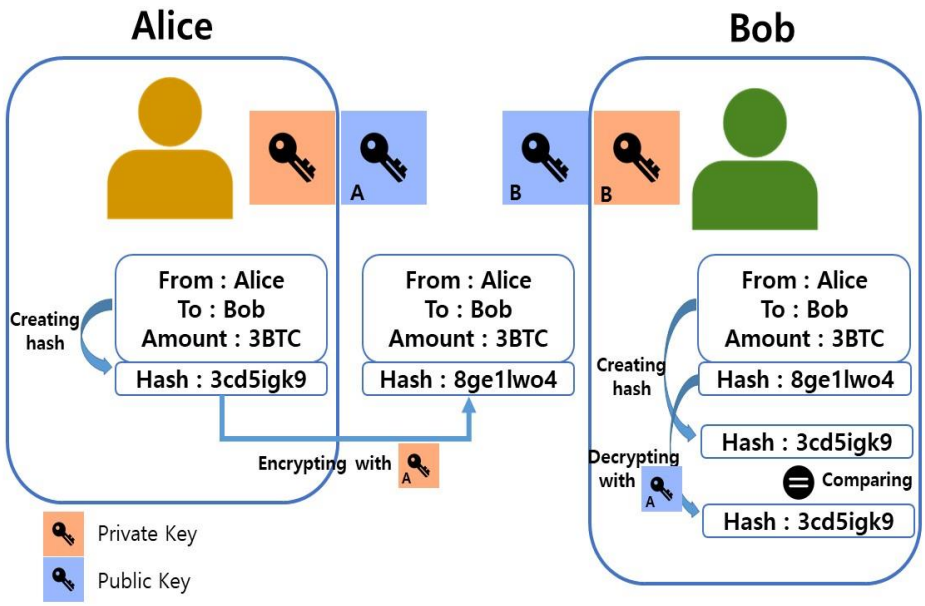


Figure 3. Digital signature



Blockchain is a public distributed ledger and is continuously growing due to the increasing number of the participants in Blockchain (Nakamoto, 2008). Blockchain was devised to prevent double-spending in peer-to-peer transactions and consists of blocks containing the transactions occurring during a certain period. Every participant updates and saves it.

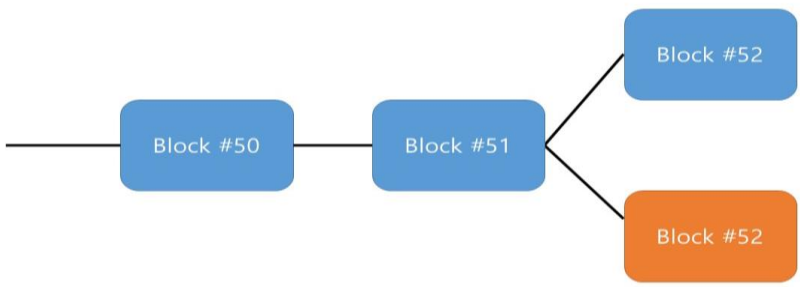
2.1.2. The principle of Blockchain

In his paper, Nakamoto (2008) thoroughly explains the principle of Blockchain. Blockchain is composed of blocks, so it is better to know the structure of the block. Block is composed of two parts: the header and the body. The header has the hash of previous block, calculated hash of transactions, timestamp, the level of PoW (Proof of Work) algorithm's difficulty, and Nonce. Hash of the block contains the hash of the previous block, so hackers should counterfeit the whole Blockchain if they want to fake one block. It is nearly impossible to do it. Calculated block is open to all participants of Blockchain and these participants automatically monitor Blockchain.

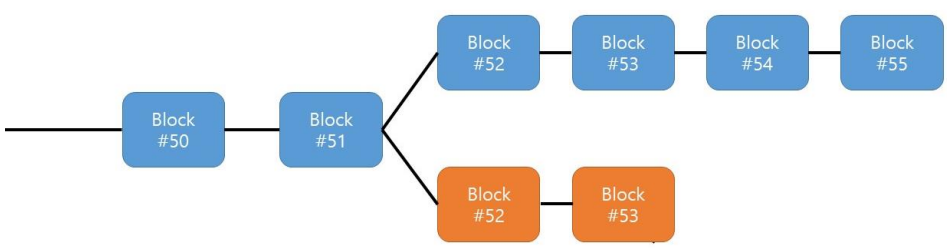
Blockchain is made of the blocks composed of transactions in a certain period of time. It is created by the following procedure. First, a new transaction is broadcast to all nodes in Blockchain. A node is the same as a participant in Blockchain. They collect new transactions into a block. Each node does PoW. PoW is finding a nonce by incrementing it until a value is found that gives the block's certain hash. Next, when the node finds a PoW, it broadcasts the block to all nodes. The node accepts the block when the transactions in the block are valid and not already spent. Nodes express their acceptance for a block by using a previous hash when calculating the next block.

Most votes to Blockchain are represented by the longest chain. It means that the longest chain becomes the valid chain and it is accepted by all nodes. Even if the two nodes create new block at the same time and it makes Blockchain forked, the problem is solved when a new block is added to the chain that makes one of chains become the longest one. The process of choosing Blockchain is illustrated in Figure 4.

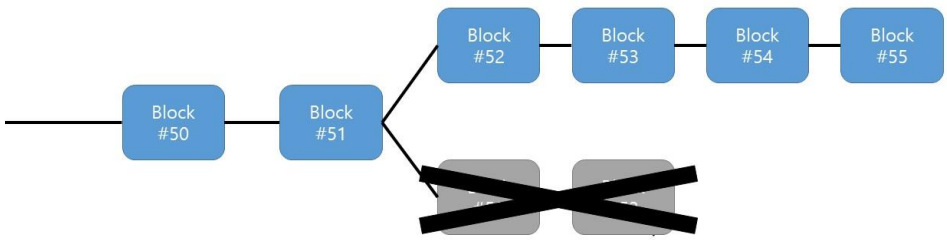
Figure 4. The process of choosing Blockchain



① Two nodes create new blocks at the same time.



② Miners make other blocks with the forked Blockchain.



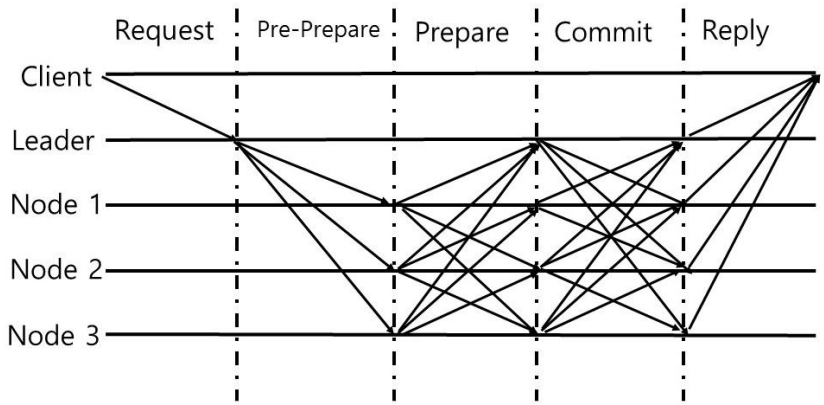
③ The longest Blockchain is chosen, and the other is banished.

The principle of Blockchain makes it particularly valuable (Underwood, 2016). Blockchain is open to all participants, so it achieves maximal transparency. Even though the transactions are open to the public, the parties in transaction are anonymous. Thus, while there is no privacy protection problem, a third-party has the risk of a personal information disclosure. Also, Blockchain does not need intermediaries, which reduces the cost of transactions and time for the third-party. As the ledger is distributed to all participants and there is a consensus algorithms (PoW), the risk of forgery is almost zero. Therefore, Blockchain is trusted, auditable, and immutable by itself. That's why Blockchain is thought to empower not only commerce and industry, but also the government.

There are various consensus algorithms like PoW (Proof of Work), PoS (Proof of Stake), DPoS (Delegated Proof of Stake), PBFT (Practical Byzantine Fault Tolerance), and Raft. PoW, PoS, and DPoS are good for public Blockchain, but others are good for private Blockchain (Mingxiao et al. 2017). In PoW, vote depends on CPU, but, in PoS, it depends on the stake invested in voting. DPoS is similar to PoS. On the other hand, PBFT is known to be operated better in private

Blockchain. PBFT works that are totally different from the algorithms mentioned above. The principle of PBFT is shown in Figure 5.

Figure 5. The principle of PBFT



The process for PBFT has four steps: (1) Leader collects Client's request and sends it to other nodes; (2) Nodes who got the request execute it and send the results to other nodes; (3) If nodes get the same results from over 2/3 of all nodes, they decide to add block to their Blockchain. If not, they decide not to do it. (4) After the process, all nodes have the same data agreed upon by over a half of nodes (Migule et al., 1999).

2.2. Blockchain and Database

2.2.1. Scalability issue in Blockchain

While Blockchain has a huge potential in many areas, there is the scalability issue. To handle more data in the future, Blockchain should solve the problem related to scalability, such as throughput, latency, capacity, and network

bandwidth.

Throughput is the amount of data Blockchain can handle. For example, the Bitcoin network processes on average from two to four transactions per second (tps). It could handle a higher throughput if each block were bigger, though right now making blocks bigger would lead to size issues. This throughput is unacceptably low as compared to the number of transactions in other institutions like Visa (2, 000 tps average) (Trillo, 2013), Twitter (5, 000 tps average), advertising networks (500,000 tps average), trading networks, or email networks (global email volume is 183 billion emails/day or 2,100,000 tps) (Sourabh, 2014). An ideal Blockchain should support high throughput in multiple uses.

Latency can simply be described as 'waiting time'. This means the time spent on process transactions. Each block on the Bitcoin blockchain takes 10 minutes to process. In every ten minutes, a new block is broadcast. To ensure security, it is better to wait for at least an hour, giving more time to nodes for the consensus. By comparison, a transaction on the Visa network is accepted in seconds at most. The confirmation of Blockchain is considerably delayed in Blockchain.

Capacity and network bandwidth refers to the amount of data that Blockchain can handle. The Bitcoin Blockchain is about 50 GB; in 2015, it grew by 24 GB (Blockchain Info, 2015). It takes a day to download the entire data and transactions in Blockchain. If the throughput increased by 2,000x to Visa levels, the additional transactions would result in the database growth of 3.9 GB/day. At 150,000 tps, Blockchain would grow by 214 PB/year. If the throughput were 1M tps, it could completely overwhelm the bandwidth of any node's connection, which is counterproductive to the democratic goals of Blockchain.

However, the technology Blockchain has several problems. Specifically, PoW takes too much time to quickly handle transactions. If the network wants to accelerate the transaction speed, it has to compromise the security. The other cause of the problem is that all nodes have to download the entire database. It is difficult to keep all data on one hard drive, and this makes nodes give up seeing the records anymore. Ironically, this leads to centralization. With the growth of the amount of the data, only nodes who can save the data would keep data. They become the only participants on Blockchain. Finally, Blockchain uses broadcast as communication protocol. Bitcoin uses a simple broadcast network to propagate transactions, meaning that bandwidth use increases as a square of the number of nodes in terms of bandwidth overhead. To solve this problem, Blockchain communities consider many alternatives, and the BigchainDB can be the one for it.

2.2.2. The BigchainDB

BigchainDB is the database based on Blockchain. It has Blockchain characteristics, such as decentralization, immutability, and owner-controlled documents. In addition, it has also inherited the desirable database properties such as low latency, high transactions rate, and high capacity. It combines the advantages of Blockchain and database (BigchainDB, 2018).

The data stored on BigchainDB can be shared with all participants and cannot be erased or changed. BigchainDB uses MongoDB, and all participants have its copy. Even if some nodes are changed and corrupted, other nodes are not affected and still have the copy of all data. Also, it uses digital signature for all transactions, so if someone wants to change the data, s/he has to change the digital signature that is cryptographically encrypted.

The throughput and latency problem is solved with Tendermint-based network used by BigchainDB. Tendermint consensus can process thousands of transactions per second, with latencies of one or two seconds. It is benchmarked by 64 nodes distributed across 7 data centers (Cosmos, 2018).

There are many use cases with BigchainDB, such as supply chain, intellectual property rights management, digital twins and IoT, identity, data governance, and immutable audit trails. Therefore, much attention is paid to BigchainDB in the area which should handle massive data.

In the BigchainDB, there are two types of transactions: CREATE and TRANSFER. The CREATE transaction literally means creating asset in a database. It needs the owner's key pair and the contents of the document. When the owner provides a signature with his/her private key, the transaction is completed. The TRANSFER transaction means transferring the data to someone. In this transaction, a new owner provides a signature with his/her private key and it changes the signature on the document. It means that the new owner is authorized to spend this document.

The process of transactions in BigchainDB is simple. There is a client and a server node. The client sends transactions to the server node and requests to process them. The server node processes transactions and sends it to Blockchain. In the server node, there are backlog and Blockchain. Backlog is a transactions set that actually processes the transactions. Blockchain validates the block, which means "etching content into stone". It checks whether the block is valid or invalid. Every block starts from the undecided status. When most votes for a block are positive, it is decides that the block is valid; when most votes for a block are negative, as the block is determined as invalid. An invalid block is sent back to

backlog. Chainification, adding block to Blockchain, happens at voting time, not the time the block was first written to the database.

2.3. Resource management by Blockchain

Blockchain facilitates resource management. Every transaction and process is remained on Blockchain. It cannot be changed and faked. Using Blockchain, resources can be managed in the safest way.

Digital resource management based on Blockchain has been already suggested (Chakraborty et al., 2017). The digital resource is referred to any resources made by online personal or organizational activity online. It is not limited only to the SNS profile, e-mail, and so on. It can be any online activity. As most of resources are dealt with in an online environment and contain personal information, there is a fatal risk if server managing them is hacked. To prevent hacking and make them safely saved, Blockchain-based resource management is designed.

It uses a multichain which makes each Blockchain exchange its resources without the compatibility problem. In a multichain, four streams are made for key generation, resource creation, and sharing with another user. The algorithms can be explained by online educational certificates. In Stream 1, every student in the system has the public key. In Stream 2, the course completion data is saved and encrypted with the data's public key. Every public key is also saved. In Stream 3, private key for the data is saved. It is encrypted with the public key of each student. In Stream 4, the actual transaction is held. Students encrypt private key of the data and their digital signature with public key of employers, which is called access key. Employers get the access key by decrypting it with their private key. By using

private key of the data, they can get data saved in Stream 2. The data can be accessed within a limited time if users set the limitation. By using Blockchain, the actual owner manages his/her own data.

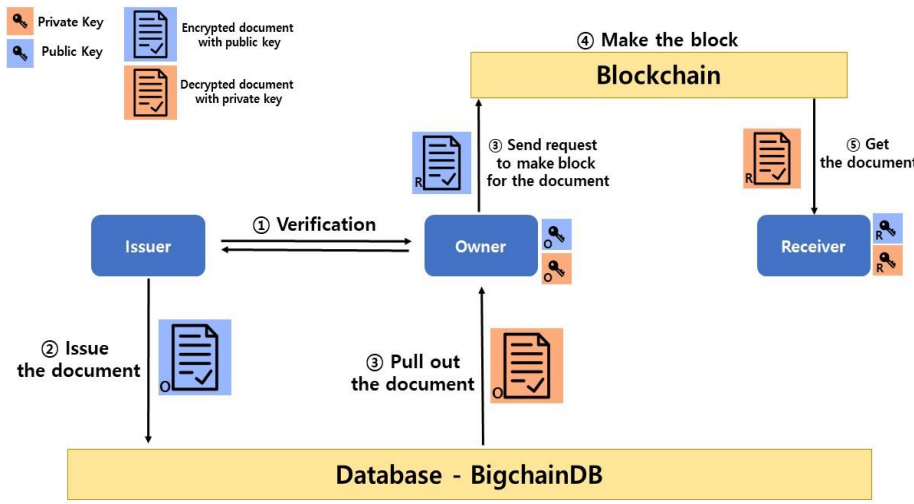
Kishigami et al. (2015) also suggest the digital content distribution based on Blockchain. It is also designed to guarantee the rights of actual owners. That is, holders can manage their own digital contents and operate the suggested system. The system is aimed at simple, easy, but reliable operation. Reasonable security and simplicity should be realized. The first target of the system is the super high resolution video. There are three entities: licensor, licensee, and mining server. Licensor can only control its contents and has the permission to send these contents. Licensee gets the permission for contents from the licensor. The mining server generates a new block and adds it to Blockchain. It shows the simulator and proposes future function to be added.

Many sectors in industry actually have used Blockchain for resource management. The representative applied industry is agriculture and healthcare. The supply chain of agriculture should be traced and controlled by a credible system. The Blockchain system with RFID can be the answer to this issue (Tian, 2016). Originally, the hospital or insurance company has most healthcare information pf he patients. However, healthcare data are personal data, so they should be managed by their owner. To realize it, Blockchain has been introduced in the healthcare system. Owners can manage and send them only if they want (Yue et al, 2016; Rakic, 2018). If Blockchain is operated with IoT like a wearable device, it is possible to get real-time data without any effort (Linn and Koo, 2016). Furthermore, Blockchain is also useful to resource management in patent pledge (Shanahan, 2016) and human resource information (Wang et al., 2017).

Chapter 3 Model

In order to make the model for the Minwon24 system, it should be considered that the system is comprised of two parts: (1) Issue and (2) Exchange. The overall design for the model is shown in Figure 6.

Figure 6. The overall design of the proposed model



3.1. Participants

First, to demonstrate the system, we identified various participants involved in the system. The three participants in the system are as follows: (1) Issuer, (2) Owner, and (3) Receiver.

• *Issuer* The entity that will issue the document in the database. There is only one issuer in the system, and it participates as a node of BigchainDB.

In our model, the issuer would be the government.

• *Owner* The user who has the ownership of document in the database and who can control access to this document.

• *Receiver* The user who wants to get an access to the document and proves that resource is sent from the real owner of document. Its identity can be known by public key it has.

Issuer controls all of resource only in the database. To approve that the resource is verified, all resources in database are owned by issuer. That is, the public and private keys of the issuer are used for issuing resources. Even if the owner of resources is the issuer, the content in document is encrypted with the owner's public key. The contents in document cannot be arbitrarily changed by issuer.

For an exchange, owner and receiver need to have the public and the private keys. It is similar to having a digital wallet in a Bitcoin transaction. The private keys should be kept safely. We assume that they already have their own public and private keys. The receiver can be a person or an organization.

3.2. The new form of civil documents

To issue the documents and make them secure, a new form of civil documents is required. The new document is made as a dictionary. It is better to find the information than the document used before, because it sorts the information by items. Also, since the old one is a PDF or an image file which takes more capacity as compared to that of a dictionary, it reduces the capacity of

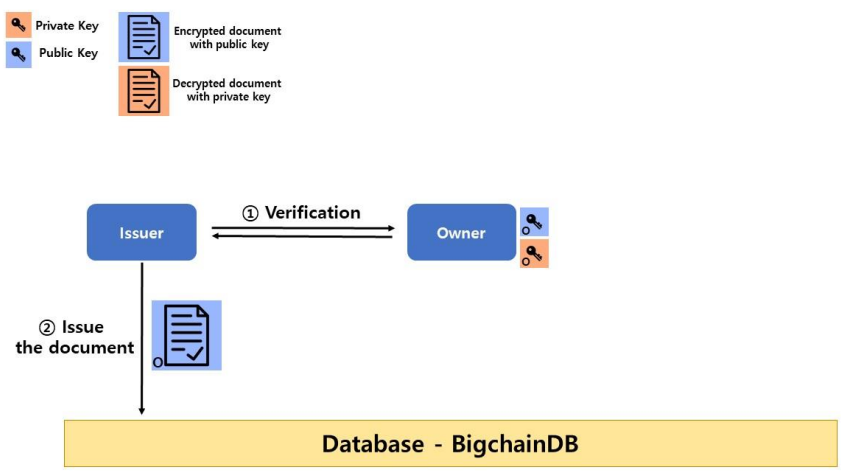
documents. To make the documents secure, the digital signature is attached at the concluding part of the document. The digital signature is made by a private key of the issuer and is verified with its public key. It means that if receiver wants to check whether the document is faked or not, it can be easily verified with issuer's public key. The process of signing and verifying the digital signature is presented in Appendix.

The items used in the new form follow the old form. The needed in the documents, such as a land registration map and a forest land map, are made as string type. Figure 7 shows the new form of the documents among the five most frequently issued documents. The remaining types of documents are shown in Appendix.

Figure 7. The new form of civil documents(Land register)

```
land_register = {'서류명' : '토지대장', '내용': {'고유번호' : '1144012500-10226-12345', '토지소재':'서울특별시 마포구 성산동',
                                              '지번':'123-4', '축척' : '1:600',
                                            '항목1': {'토지표시': {'지목': '08 대', '면적(제곱미터)':'144.5', '사유':'구획정리 완료'},
                                            '소유자' : {'변동일자':'1980.12.23', '변동원인' :'소유권 이전', '주소' : '서울시 행운구 행복동',
                                                        '성명' : '홍길동', '등록번호':'560702-1******'}},
                                            '등급수정':{'1':{'년월일': '1983.05.16', '토지등급':'75'},
                                                        '2': {'년월일': '1984.07.01', '토지등급':'200'}},
                                            '개별공시지가' : {'1': {'기준일':'2009.01.01', '지가(원)/m2' : '30100000'}}},
                                            'DS' : b'\xaf\xf7\xb8\x1c\xac\tK\x907\xbe\xb4\x9b\x06]s\x9b\x0b\xd7\xef\x8f\xecU\t\xbf\xb7\x01\
```

3.3. Stage 1: Issue



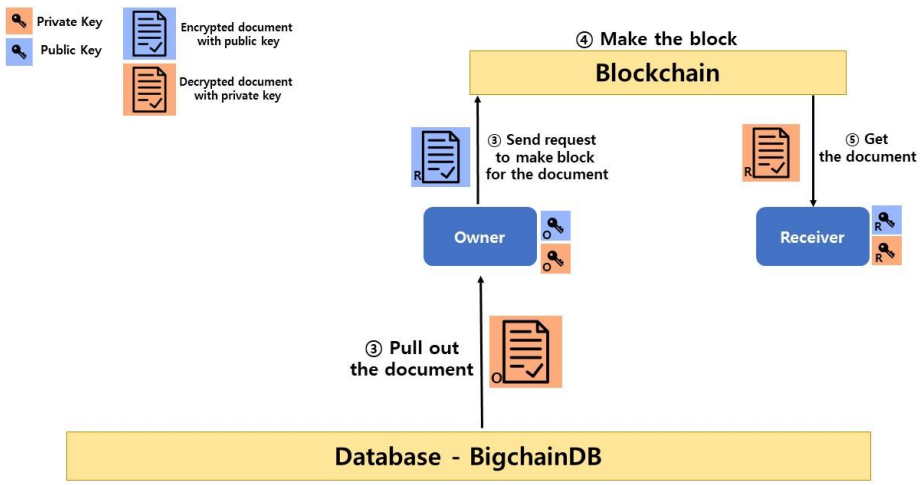
The first stage for the model is ‘Issue’. The ‘Issue’ stage is literally issuing the document to the database. The Issue’ stage is needed to verify the document which will be exchanged. The document, a civil document, should be approved by a third party, such as the government. It is a public document and should be trusted by anyone. Without trust, the civil document cannot take effect when it is needed.

To issue a document, general Blockchain is not enough. As mentioned above, Blockchain has the scalability problem. There are so many civil documents that a database which will solve the scalability problem is needed. The database used in the proposed model is BigchainDB.

Before issuing a document, the document should be encrypted with the public key of the document’s owner, as the content in BigchainDB can be seen to everyone participating in the database. When the process for approving document is completed, it is ready to issue the document to database. After encryption, the document is issued to by CREATE transaction in BigchainDB. To prove that the document is verified by the issuer, all documents should be issued with the public and private keys of the issuer.

While it might seem that the issuer is the owner of the document, the document is encrypted by the public key of the owner, and the content can only be accessible with the private key of the owner. No one can read and change the content of the document. The issuer just performs an administrative task.

3.4. Stage 2: Exchange



After the ‘Issue’ stage, owners can use their own documents as they want. When they want to send their document(s), one more stage is needed. The second stage is the ‘Exchange’ stage.

Without the limitation of time and place, there should be a credible route to send a document. The document contains private and important information. If this information is leaked, owners can experience a critical damage. To prevent this from occurring, Blockchain is the best alternative.

With the asymmetric cryptography that uses public and private keys to encrypt the data, the content in the document cannot be read and leaked. In addition, the Blockchain allows users to know all information about transactions that have been made. This means that users can trace their transactions and have control over their own documents. It is possible to set up conditions like time, the extent to open, and so on.

When an owner wants to send his/her document, the document is pulled out from the database. Since it is encrypted with the owner’s public key, the owner

decrypts it with his/her private key. After decryption, the owner gets the content of the document and encrypts it with the public key of the receiver again. Due to encryption, the document is protected by another person except for the owner and the receiver. After that, owner sends the request to Blockchain to make the block for documents receiver wants. As in traditional Blockchain, miners calculate the hash of collected transactions, and the new block is added to Blockchain. Once the block is added, receiver can get the document which is encrypted with its public key. Document will be decrypted with receiver's private key and receiver can access to content of document.

Chapter 4 Results

In the present paper, we incorporate the actual model with Python, one of the most widely used programming languages. The proposed model has the following two parts: (1) Stage 1 (2) Stage 2. The code for the whole process is provided in Appendix.

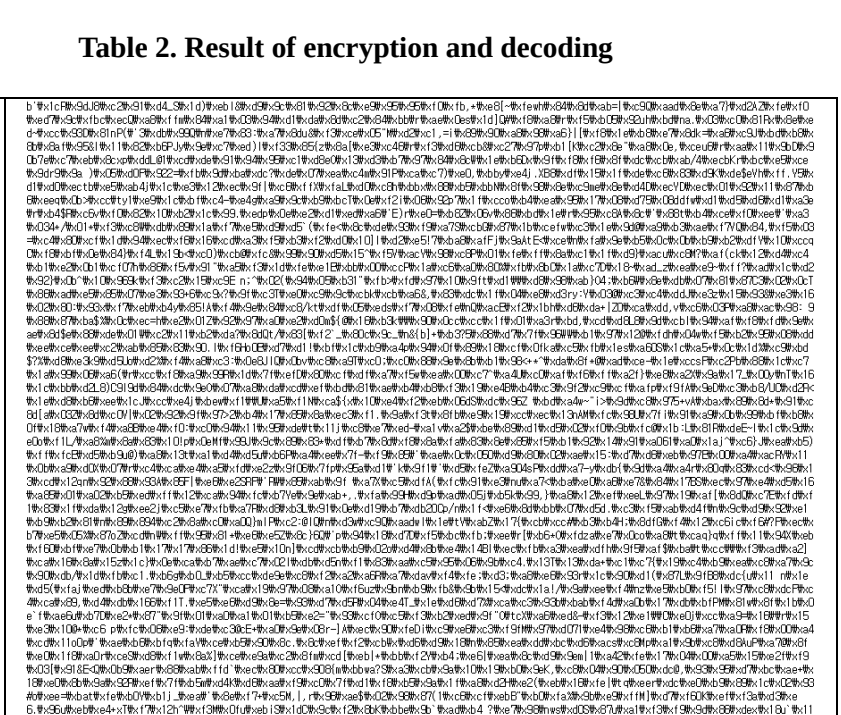
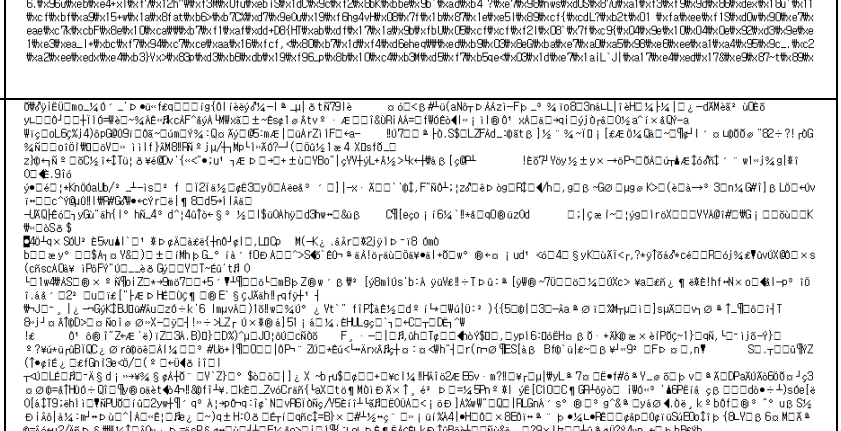
4.1. Stage 1

In Stage 1, the first thing to do is to create the public and the private keys of the owner and the receiver. To his end, the ‘pycrypto’ module is used, which is a collection of both secure hash functions (SHA256 and RIPEMD160) and various encryption algorithms (AES, DES, RSA, ElGamal). Among them, RSA is used for creating public and private keys. RSA is one of the first public-key cryptosystem that is widely used for transmitting data. In RSA, the encryption key is public, but the decryption key is possessed by only owner and thus private and confidential.

After the creation of the key pairs, the data should be encrypted with the public key of the owner. It also uses the ‘pycrypto’ module. Among the packages in that module, the ‘Random’ package is used to avoid making an attack on the data.

In the program, it is assumed that the data type is dictionary. The data type of encrypted data is bytes, but BigchainDB cannot accept byte type as an asset. Before putting it to database, the decoding process should be performed. For decoding, ‘ISO-8859-1’ is used, which is a commonly used character encoding. The results of encryption and decoding are shown in Table 2.

Table 2. Result of encryption and decoding

Encryption with public key of owner	
Decoding	

The last step for Stage 1 is issuing a document to the database. As specified above, in the present paper, we are going to use BigchainDB and, among various drivers, the Python driver for BigchainDB is used. To get the BigchainDB network, the virtual network environment is used. To issue a document to the database, the key pair for issuer is used. The issuer seems to be the owner of the document only in the database. The ‘CREATE’ transaction is created with the issuer’s public key and is sent to BigchainDB network to complete the transaction.

The completed transaction is shown in Figure 8.

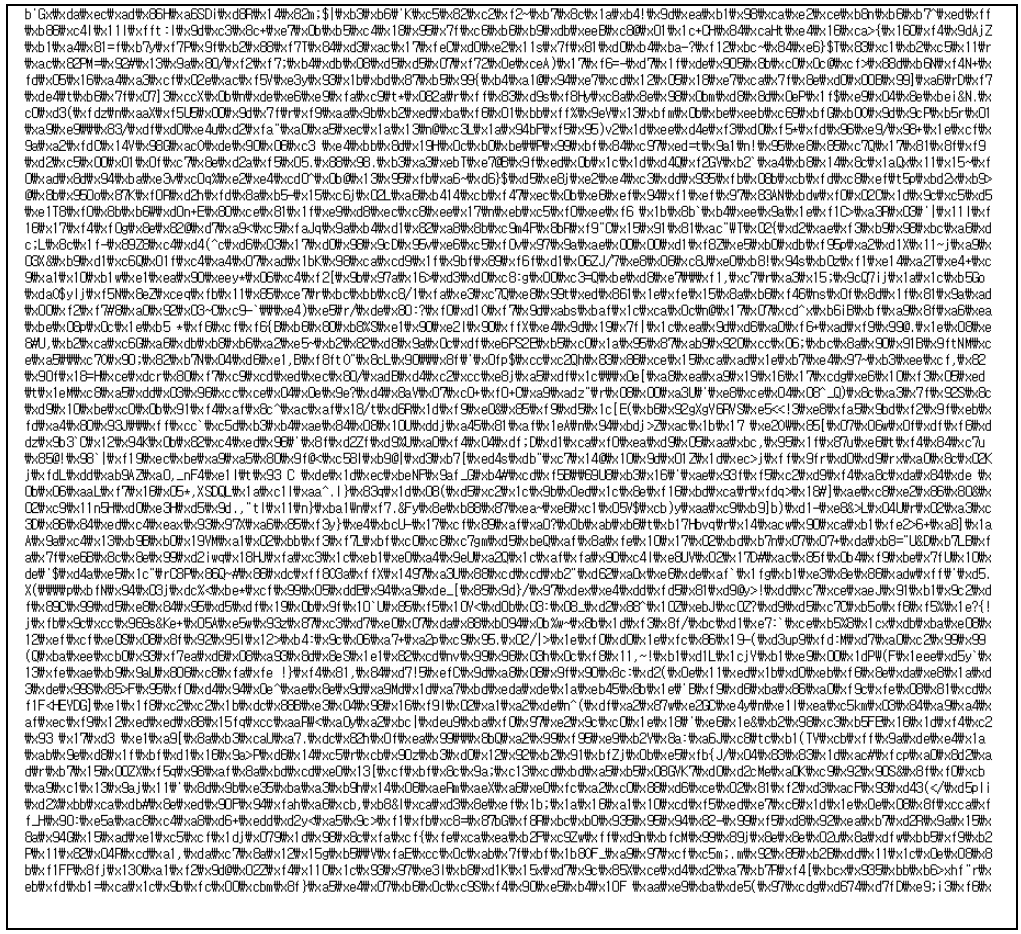
Figure 8. Completed transaction

{'inputs': [{'owners_before': ['3TpRAVQaBwxTAzHdRXPm9rCVa3Vh6BmzAHuqhJnvo4xP'], 'fulfillment': 'pGSAICSWFst9zAr8Wg5L4CqNFX4NtAIGTg-h2NzmSMJ8PUbVgUD8SBNlrU3Hs9cJSiQaUzlxyeSKcSJjoOkWyxgpAHiwNHvTr4HTVpA2l7u f8N8cWlbnngLR-Vp7Vh_KQaVo8rQB', 'fulfills': None}], 'outputs': [{'amount': '1', 'condition': {'details': {'public_key': '3TpRAVQaBwxTAzHdRXPm9rCVa3Vh6BmzAHuqhJnvo4xP', 'type': 'ed25519-sha-256'}, 'uri': 'ni:///sha-256;MIFcKHOcKDeTZ87OmImEyKVDvtS8JEWKOqS2hS-xrzE?fpt=ed25519-sha-256&cost=131072'}, 'public_keys': ['3TpRAVQaBwxTAzHdRXPm9rCVa3Vh6BmzAHuqhJnvo4xP']}], 'asset': {'data': {'content': '{"k1cFR%xBdJ&k%x910_S#x1d)ël&U%x9c%x81%x92%x8c#%x9E%x95ă ù.+ê[~þ wh%x94%x8d<=[éQª d%x8e§}ðAZþ ă í'%x9cûciQ¯ ym%x84¡w%xO3%x94kU%xBdA%x84-%r@%xOes%x1d]Q#ø ¨ %rõ° 5%x92uh½%na.%xO3A%x81Rx%x8eí~í%x93D%x81nP(%'30%x99Q%ng%x83: §%xBdu&ôí%xO5¨M0A,=i%x89%x90¨ %x98;}|[ø %x1e¸ ç%x8dk=¦ÉJ½ ¸ %xB0%xBafl%x95&l%x11%x82¶ PJy%x9eçí)ló3M%x85{z%x8a[šÃ6%r6OE&A'7%x9fþ±[KA%x8e¨ ¨%xOe,ïuß%r ª %x11%xBbD%x9Ob7eçe%x8cxpýL@1ì Þ %x91%x94%x95Aø eO%x13ó· %x97%x84%x8c%x1e¶ Dx%x9f ø %x8fUE«/4ibKr¼ã!%x9dr5%x9a)%x05ÐF%x922=o%x9d ª Ü?Þ %xO7ëÃm%x91PEç)à,¬yáj.XB8ß %x15%x1fÞ Æ %xB3OKÞ $eVhý.V5ñ Ð¡itbá÷4j%x1cã%x12i%x9f|Æ ýXúL Ð Eh÷»%x88µ>N%x8f%x98%xBeEme%xBe0DiVDi%xOl%x92%x11%x87¶ ¡q%xOb>l%ty1é%x1c ¿ Ã¬ãg@%x9c' ¼T%xOeòi%xO6%x92p7%x1flo ´ é%x95%x17%xO8× 5%xO8ddfwÑõõ%Ee%r ´ $RÆv ä %x82%x1O² %x1c%x99.ip%xOeÃñ¦;Ÿ'E)ra=¸ Z%xO6v%x86½%x1e%r%x95EA%xBc%r'%x88t ´ î ã i%'ë%xO34÷/%xO1+ó£W0%x89%x1a÷ãûÖ`(þ<%xBcÞ %x93ù§ SE@%x87%x1bîfw%x1e%x9d@0ª @ ÷ VQ%x84,ó%xO3=A%x8Oî%x1d%x94ió%x16ìæõª ò Ð%x10] lòà!'7ª 8~F}%x9aAtE<i%nù%x9eµ%xOc%xOb' ² ß V%x1OlqC» ¿ %xOe%x84}ôL%x19b<Ä)E@ù&%x99%x90ð%x15~ôV~V%x9EF%xO1þ y%x8aÃ%x1fÙ}¬uEM7~(ck%x12Ôà±á%xOb1ì O7h%x86ôv%x91'¥ó%x1dþ áE÷%xOOIP%x1aÆ %xaO%x8OX ù%x8bO%x1açD%x18~%xad_zèé~ý?%xad%x1c0%x92}%xOb^%x1O%x96Bkóá%x15EE n;^%xO2(%x94%x05ª 1^ü>ý%x97%x1O%x9ftï%W#ø%x98÷}O4;¶ W%x8e0%xO7%x81%x87C3%xO2%xOcT%x86%xad&%x8F%xO7ã%x93÷6ëx?%x9fXTáé%x9cEk£;&,%x83ù%x1f%xO4eórý:V%xO3@%Áÿ.Jš2%x15%x93&%x16W%xO2%x8O:%x93÷ë ´ y%x85!Ã6%xSe%x84E/kt ß %x05is÷ %x08þ %nQ~Eã%x1bhûU+|ZDEÝ,vÆ %xO3P¨ ¬%x98: 9W%x88W%x87ª $%W%xOci=ha%xOIZ%x92%x97W%xaOi Ð m§{ø%x16ª k#W%x9O%xOcc!%x1f%xO1ë r½,!ø L$W%x9dE!W%x94¬ ø ÿW%x9e@W%x8d$e%x88Þ %xO1W%x11ª ú?%xBdQt/W%x83[à`_W%x8Oc%x9c_W%&[b]+° 2%x88×%x7f%x96WW ±W%x97W%x12@ýh%xO4w6ª %x95W%xO8Yîîîà·W%x85W%x83W%x90.loHoOE× ñ! ¿ %x1c' ¤ p#%x94W%xOf%x89W%x18Ï%xOfkaÃú%x1es;O9W%xlc¥+W%xOc%xld%E½$?%ø ãk%90Uo0%ö¯ Ã:W%xOe8JlQW%xObvE0TÃ;A%x88W%x9e%x8b ± %x98<++^0W%x8ï+0W%xdi~W%x1eIsFÃPb%x88W%xlc ç%x1a#%x9W%xO6;{Wrí ø@W%x9RW%x1dW%x7fiDW%x8Oï§ ö%x#%xOQ;^¤ UÀ~öÿ¿f}ê¿X%x9a%x17_W%xOyWnT%x16W%xlc»oL8)C9I9d#%x84O%x9eO#%xO7¨ Ûlï½%x81@ ' ¸ ó%x19ÃB ´X%x9f2Ei-pùfA#%x9eDÃ¸ /UCORÆ%x1eØ¶ ï%xlcJíáj¾w%WW¥N£${x%xlOàë%xO6dSU%x9GZ ½¤w~ˆi>%x9dE%x975÷vÃª xW%x89%x8d+%x91Ed[a#%xO3Z%x8dÁV|%xO2%x92%x9f%x97>2´ %x17%x8B%x8aiÃ6.%x9aót%x8fb€%x19Ii%x13nAMú%x98U%x7fi%x91ó%xOb%x99¿ ¸ %xOf%x18§ w6¨ Bà ã :Ã%x94%x11%x95Þ %t%x11jÐ£çí~¡ vç$¾%x89%O%xO2ã %x9bu0%x1b:L%x81RÞ E~|%x1c%x9dañL/¨ Xw%x8a%x83%x1O!pW%xOeMf%x89.W%x9c%x89%x83÷ß · %x8d ø %x8au%x83%x8e%x856 ± %x92%x14W%x91%xaO61%xaO%x1ajˆÆ }Jêµ)ÿüE0¹ u@)¨ %x13t ¡ ô0u ¶ P¤ï %x7f~uÐ%x85#'@%xOc%xO6O0%x8O%x02@ %x15:× ÒeW%x97E%xOO¤ ¬FV%x11%xOb@Ð X%x07#rÃÉ¥ý ãz%x9f0E%x7fø#%x95aW#'k%x9f1#'ðp Ži04sPÝ§ ¬y0{%x9d¤ ¤ r%x8Oø%x83l<%x9W%x13l%x12qn%x92%x86%x93W%x85F|æ ãSRP#'FW%x85÷%x9f §ÝÃß A(ú%x91ã%nu§ <ª a!;ç8%x84%x17ªBSi%r97aõ%x16¨ 5W%xOl%xaO2µíý%x12E%x94ù · Ve%x9e«+,.ú%x99HOp%xadW%xO5jµk%x99,}¨ %x12irLW%x97%x19-[%xBdQçEýr%x83%x1fO%x12ø1ž2Ã¿ú§ Rø ° L%x91%xOeÑ9 · O2OCp/n#x1f<æ%xBd÷%xO7Ôd.Xõ÷Of%r%xGcO%x92à' ² %x8l%r%x89#x894A%x8aA%xaOJ%mlPÃ:@l Q#nówEQª dwl%x1e#tV÷Z%x1?{Ei#³ ´ H:%x8dfG%r%x12Æ ícø#?Fi · ã%x05X%x87oZI%n#ý%x95%x81+æ ãZ%x8c)6O#'p%x94%x18× D6¼ú;r%r[¶ +Oyzaç%xOco¨ Ÿt€q}qy%x1l%x94Xë0Ú¿ ç%xOb±%x17%x178%x88%xld!ã%x1Oñ]lÊ' %xO2o0%x8ba%x14Blûú€ß h%x9f5~$ª Ÿt|%W6%xad@]E%x16W%xBa%x15z%x1c}%xOaÊ · @ç%xO2lOÕn#%x83ª A%x95#%xO6#%x9bÃ.%x13T%x13O+Aç{%x19Ã' âÊ§ %x9c%x9OO/%x1daÀ.¶ g° _µíÞ 9aEòø¦R§ Úvõþ :ó;¯ æ%x93r%x1c%x9Oñ(%x87LW%x9fB8O{uW%11 n%x1eõ(újí¸ b÷%x9eOFçX"E%x19W%x97W%xO8¡ Oèuz%x9bn' ú&W%x9b%x15dJ%x1a!/W%xBaïë#nzâ° õ!W%x97ëUFA£%x89,00W%16ñT.â æ U%xBe=W%x93× ðFW%xO4aT_W%1eõ× %EA%x93b ª bôd#%xaOb%x170¿ PW%x81wW%x8f%x1bW%Oe`f@Eu · Dê+%x87`W%xf%xO1W%xaO¡ %xO1µâ=`W%x93lOÃó² íW%9f`OW†cX¡ íã~ê%x12ã%W Ãj!@=%x1E#W+%x1S%x1OÐ+Æ pú%xO6ë:Þ X@cE+W%aO%x9e%xO8r-]Ai%x9Op DíÉ æ Xúf MW%x97Ð 7la%x98Æ ± ¶ § %xaOR÷ %xOOα i%x11oOp#'@ ¶ ¿ qúVîµ%x90W%xBc.%xBcïöÉW%xOU%x18W%x85aÝ¼ô~sÉMp¡ %x9bÊØ AuP§@W%x8fa%x1f8%xaOrîSØñW%x8a%]îéaÄ%x8fml[ê|+÷>ôV ´;æ|ê%x8cU#x9em||¤2Þ %x17W%xO4W%xOO+%x15fu%xO3[W%x91ßE<Q%xOb9@ r%x88÷ÿd`i%x8Ol#%x9O8{m÷wa?Sc£W%x9a%x1O%x19° W%SeK,ê%xO4W%x9OW%xO6OU@,W%x39W%x95× ¼@+W%18a%x8b%x9aW%x92Ri%x7fµmÖKõª ùÄW%x7fÑø µW%x9aW%x1f ¨ ÒHá(ë%x16þ |Wtq!rÙà' W%x89W%lcW%O2W%x9Woi=ª tþ ° V ±j_ë#'W%x8e÷+AM,|,r%x96@ $W%O2W%x98W%x87(lÆ îëB"° ú%W%x9b€yM]× ôOKíóaOæ .W%x9Gu8ä+xT÷ %x12h"WóMW%OfuëiS%x1dOW%x9cö%xBbK÷e%x9b`W%xad ´ ?ç%x98WnwsÐ SW%x87u¡ ôù W%x9dW%x86Þ xW%x18u`W%x11ï¿ @W%x15+w%x1a%x8fat¶ > · C%× %x9e0uW%x19ghg4vHW%OEW%x7f#x1bW%x87W%x1eá!W%x89Ï{1L?ª t%xOI úïñSÐ w%x90¿ëeçkëF%x8e%x1OE%WP · ñ-Ÿ+D6{HT÷ß %x17W%x1a%x9bùL%xO5Ïìò]W%O8'%x7fÉ{W%O4W%x9e%x1OW%O4W%xOe%x920%x9eaã¢_l+¼ ÷ %x94çÏª %x16ûf,<%x80· %x1dôõeheq₩₩ï' %xO3W%xeG°g%xaOe₩%x98æ î¡ ¤ W%x95W%x9c_.Âçî¡xã° }Vx>W%x83þO¶ O₩x19ú6_p₩x8bW%1Oã° MÖ÷µqe<W%xO3W%x1dç%x1aiL`J¦ ¡ 7ã i%x17&%x87~t₩x89W%9byïW%xOdEW%x1b!W%aOQÃaA₩%x8c:d[á<ô₩x9c%x13æ ôÝ%x8530pß¦ dymoÏï° ª a₩%x81ayejÊñçï¨ W%x83E%x11%x850W%x12e%x90W%x87ö¤iP0.'fW%x1eGiß%W%x91ð @8nv ø @ù:¸ ¸ ~W%xOf₩%xO6ú4zH8=îá_W%x890W%x17(° W%x90Ki₩%xOeá£W%x1cT· Ÿh₩₩W%x13lØW%x8OW%x87$W%xOej₩%x1E₩%xadSaGblñØ ýW%x9c₩%xOO~W%x97W%xada7ã {₩'W%xO5W%x8O₩ª ã?EO₩%x87W%xOcmA₩%x1e9W%xO2GA₩%xO6½₩%x14Ø÷ú[Jb₩%xadæß ^%X>Ø0Ï¼ ¿ ° E₩%x93W%x1f₩%x8b₩%xO3E!₩%x1a₩%x86%x₩%x8eß <÷ sO₩%xOfN₩%x87· ₩t₩T₩r .₩%x85ÿZ₩%x99Cg,AEïã'₩%xO2<₩n₩%x9a₩%x8eãÙ· çhPû~4S₩%x14₩%x9dµù₩%x10₩%xOEN× ₩%x9bt₩%xad₩%x11>{1é½ME,ô0₩%x7fç»æ ¡ >ú¡f₩%x08æ₩%x9df!₩%x8de₩%x820X¨ ¨r₩r%x13à£ë!Vº +÷₩%x16₩%x99Huµ%ãï₩%xBO₩0]À ' ₩%x9e · ŸG± ¬ï ´ ¡ ₩%x82₩%xad₩%xOc₩%xO6₩Þ Ù² èÛ ª %x14₩%xO2Ã'^AvVV9Ab æ ₩%x13₩%xOl&₩%x850₩' ₩%x9E₩%x96y₩%x840₩%x14<XÉûr² ô£₩%xaOEnhXtL¤ _₩%x1B<«₩%x82gÀ² ₩%x91ô÷₩%x17b₩%x88ø i₩%x1c3V° L0₩%x9fúk' ª₩%x8d³ i£Q µ₩%x1O§₩%xO2<ø3₩'^₩%9bç÷ é₩₩O₩%x89ã₩%x8fA₩n₩%x9b₩%x11ó ª ëDGk ô₩%x1d₩%x82ç₩%xO7a₩%x9O₩%x9G`=p ã§kL₩%x04í₩%xOfg»æ Ã₩' § Ÿ₩%x1cí%x1diT₩%xO3M₩%x87₩%x8O-pá?₩%x19₩%x7f5₩%x9cv¼ Ø lÿczP6P6½ +₩%xO1× ₩'ú¼₩%xO4₩%x16TnEN`dW\>æ Ei₩'Þ £}(ô₩÷ . Ð 5Z₩%x9eiO₩%xO7G₩%x12-Ù ö ë Ài¼₩%xO6 rÃ ¶₩%x16₩%x95₩wÃ ø₩t &o ¸ ÿ₩%x8G₩%xOOLR{₩%x95₩%9d6e¨ï~₩%x8e ¸ +sZ)e÷₩%x1c÷ ₩%x1e₩%x8fó ø÷₩%x92₩%xOb₩%x7f5₩%x95Ã? ´ ó₩%x986₩rX½₩%x16@Dov!P₩xad₩%x88ÿ@Ø ø ° ₩%x16₩%x89Ø₩%x9e₩%x91ôß FD/₩%x8fy+!tyn#¸Ù¸ ₩%x994₩%x8c> ¶ i₩%AÐ3₩%x8d § æ X₩₩>ßÙÃ₩%x83µß ₩%x12,₩%x8g₩%x95@%kh₩%x87₩%xadiBÃ}æ₩%x86tú¿ ₩%x140Ð ð wNi6lEA%sç1₩%x8fð Ù₩%x8bD² ûr₩%x19₩%x7fñÃiú₩%x86]ë₩%x8f æ₩%x96´ ÷₩%x99₩%x1d[ûTšÃOi₩%x8a6w₩%x9ae₩%x8a₩%x1eö₩%x13dÃL00¿ ₩%x9e_₩%xOIr₩%x93₩%x92`mo9Z₩%x13=f₩%xO8jzdlQU₩rqRçW£á¤ `}}, 'operation': 'CREATE', 'metadata': None, 'id': '257237c42cbac289e53c3e0a5611353 44b7a3032 46cde7f2b953f5fb61alcf6c', 'version': '2.0'}

4.2. Stage 2

After Stage 1, the actual transaction for the exchange occurs on another Blockchain. Firstly, the owner has to encrypt the document with the receiver's public key to send it. The owner pulls out the document from the database and decrypts it with his/her private key. Once the original data are obtained, they are encrypted with the receiver's public key again. The encryption process is the same as in Stage 1. It is ready to be sent to the receiver. The results of encryption are shown in Figure 9.

Figure 9. The results of encryption with the receiver's public key

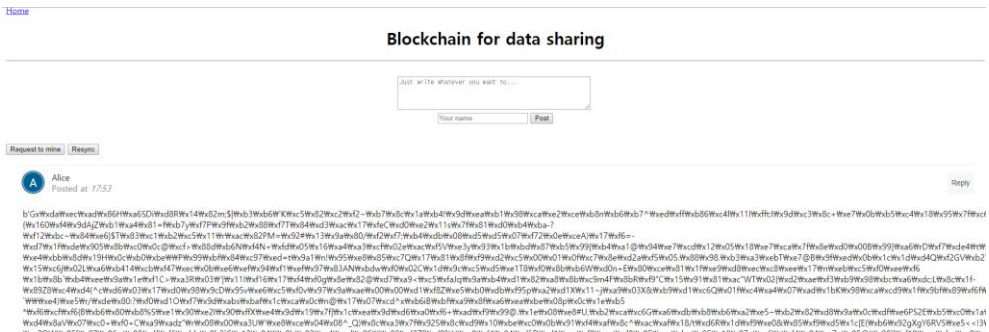


Blockchain for transaction consists of two parts: the Class for Blockchain and the application to realize the class. In Class for Blockchain, various functions are included, such as computing hash, creating block, adding block to Blockchain, as well as mining and validating block. In the application, 'Flask', a Python web framework, is used. Functions for data input are included, such as getting chain, mining, validating, consensus, and announcing the new block. The code for Blockchain is derived from Kansal's code (2018). The application is shown in Figure 9.

Figure 10. Application for Blockchain

The owner puts the encrypted data with the receiver's public key and clicks on the post. Collected transactions are subjected to be mined. Next, owner requests mining and the new block is added to Blockchain. It would be announced to other participants in Blockchain. The application looks like Figure below after announcing.

Figure 11. The application after announcing



The final step for Stage 2 is decryption. The receiver decrypts the document with his/her own private key and gets the original data from it.

Chapter 5. Evaluation

In this section, we evaluate the proposed model based on various criteria, such as identity management, fine-grained access control, scalability and distance access, and compare proposed model to the existing models according to these criteria. They are related to data sharing and resource management based on Blockchain. The explanation of criteria is also referred. The properties used in evaluation are derived from Zhang et al (2018).

• *Identity management*. Participants in Blockchain should know each entity's role. This means that, without revealing its private information, each entity should know who will manage the documents. This can be achieved by giving the identity to the participants. In the previous model, identity was confused, and it was not clearly defined even if it had a lot of participants (Tian, 2016). Also, in the other model, it just shows the participants, not their role and how to trust each other's identity (Yue et al, 2016).

In our model, three participants' identities—issuer, owner and receiver—are precisely defined. There is only one issuer who issues the documents, and receivers have their distinct public key. Participants can recognize them easily and trust each participant in our model.

In addition, identity management has to deal with the privacy problem. In particular, in our model, the privacy issue is the most important thing to be discussed, because civil documents contain much private information. Documents in the model are encrypted and stored in a database, Bigchaindb. On Blockchain, only encrypted documents are exchanged. At the same time, any access to private information of owners is impossible. The information is accessible only with their

public key. Also, access to the documents is possible only upon the owner's permission.

• *Fine-grained access control* Our model is a secure data sharing and resource management system that offers a fine-grained data access control and is based on the use of cryptographic techniques. The security of the model is operated on two levels. The first level is the asymmetric key encryption of the documents. To add the documents to the database, the issuer should encrypt the documents with the owner's public key. The issuer and the owner also have to follow the verification process to ensure that encryption is properly done. No one can open the documents without the owner's private key. This creates basic trust between the parties and rules out the possibility of falsifying the documents by a third party.

The second level is digital signature. There is a risk of performing malicious operations with documents by the owner. When the owner opens the document, it is possible to change its contents. To prevent this, the digital signature made by the issuer is attached to the document. Even if the owner changes the contents, the receiver will know it by comparing the digital signature made by the issuer and the other calculated by the receiver. Also, the entire exchange process is done on Blockchain, so the owner will be notified whenever the documents are sent and used.

Most previous studies (Kishigami et al., 2015; Azaria et al., 2016; Tian, 2016; Yue et al., 2016; Dubovitskaya et al., 2017) do not explain access control sufficiently well or do not have it at all.

• *Scalability.* Scalability problem is the limitation on the amount of transactions that Blockchain can process. The amount of transactions increases

tremendously as people use it more, but the throughput and speed of the process cannot keep up with it. It is the most well-known problem of Blockchain.

Scalability issue is not properly dealt with previous studies. It is either not mentioned at all (Kishigami et al., 2015; Chakraborty et al., 2017), or the issue seems to be solved in a theoretical way. Its solution is not tangible (Azaria et al., 2016))

However, in our model, the scalability problem is dealt with a practical way. Specifically, all documents are stored in the database called Bigchaindb. In Bigchaindb, the votes for issuing documents are limited to permissioned entities. It means that the speed and throughput for processing documents are tremendously increased. Therefore, the obstacle to scalability is removed.

• *Distant access.* Participants can be in a situation that they need data regardless of time and space restrictions. Therefore, data sharing should not be limited even if data exist far away from participants. Data should be accessible wherever participants are located.

Blockchain uses a shared network, so most previous studies suggest how to access data from a distance; however, one paper does not mention how to use the model presented (Azaria et al, 2016). In our model, participants can exchange data simply by connecting to the network. They can access to database, Blockchain, for an easy data exchange without any device.

Furthermore, Table 3 shows a comparison between our model and the previously proposed resource management models. As can be seen in Table 3, our model works better than the others.

Table 3. The comparison between models suggested in literatures

Reference	Identity management	Fine-grained access control	Scalability	Distant access
Kishigami et al (2015)	Y	N	N	Y
Azaria et al (2016)	Y	N	N	N
Tian (2016)	N	N	Y	Y
Yue et al (2016)	N	N	N	Y
Chakraborty et al (2017)	Y	Y	N	Y
Dubovitskaya et al (2017)	Y	N	Y	Y
Our study	**Y**	**Y**	**Y**	**Y**

Chapter 6. Conclusions

6.1. Implications

In the present study, we proposed the new model based on Blockchain for Minwon24, which can contribute to e-governance by introducing a technology that guarantees efficiency and safety. Although Blockchain has a huge potential for many areas, there is no actual model for e-governance. Previous studies mostly focused on the applications of Blockchain on the industrial level and the use of the theoretical framework, rather than the actual program. In the present study, we have addressed such limitations by suggesting the actual model using a programming language and provided partial solutions for them.

In this study, we have taken a different approach that the one used in previous research. We have tried to not only make a theoretical model, but also to propose the actual program for our model. Hence, we have looked through the application which can be applied to the model and found the best application for our model. When it is difficult to find it, an application is made which is fit in the model.

6.2. Limitations and future research

Our study shows the possibility that the model suggested actually can be operated in adequate environment. It suggests the alternatives for forthcoming technology. Nevertheless, it has some limitations. Those limitations provide the room for future research.

First of all, it would be better to make a program with other consensus algorithm, PBFT. In the proposed model, the civil documents are private and need to be kept safely. Furthermore, people want them to be transferred as quickly as

possible. Due to these characteristics afforded by the proposed model, the private Blockchain which accepts only permissioned participants is a better choice. In this study, PBFT is not applied on consensus algorithms. In future research, the proposed model can be devised using the PBFT algorithm.

Furthermore, the actual program would be more reliable when it is operated with multiple nodes. The proposed program is the initial model, so it is tested with just one node. It is difficult to check with the multiple nodes. Since the target of the model is all people in a country, with multiple nodes, it becomes close to what the model really seeks to achieve. In the future, experimenting with multiple nodes would make a better model.

Finally, applications in real environment would demonstrate the effectiveness of the model, as well as the problem we did not account for. First, some trials with the application in the government with application is done. After wards, the problem has to be fixed before it is introduced to the government.

Bibliography

Atzori, M. (2015). Blockchain Technology and Decentralized Governance: Is the State Still Necessary? Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id¼ 2709713

Azaria, A., Ekblaw, A., Vieira, T., & Lippman, A. (2016). MedRec: Using Blockchain for Medical Data Access and Permission Management. 2016 2nd International Conference on Open and Big Data (OBD). doi:10.1109/obd.2016.11

Bedi, K., Singh, P.J. & Srivastava, S. (2001) government net: new governance opportunities for India. New Delhi: Sage

BigchainDB. (2018). BigchainDB 2.0 : The Blockchain Database. https://www.BigchainDB.com/whitepaper/BigchainDB-whitepaper.pdf

Blockchain.info. Blockchain size. https://blockchain.info/charts/blocks-size

Chakraborty, S., Dutta, K., & Berndt, D. (2017). Blockchain Based Resource Management System. SSRN Electronic Journal. doi:10.2139/ssrn.3104351

Cosmos. (2018). Cosmos: A Network of Distributed Ledgers. https://cosmos.network/resources/whitepaper.

De Meijer, C. R. 2016. The U.K. and Blockchain technology: A balanced approach. Journal of Payments Strategy and Systems, 9(4), pp. 220–229December, 30th 2015.

Dubovitskaya A., Xu Z., Ryu S., Schumacher M., Wang F. (2017). Secure and trustable electronic medical records sharing using blockchain. arXiv

preprint arXiv:1709.06528. [PMC free article] [PubMed]

Fanning, K., and D. P. Centers. 2016. Blockchain and its coming impact on financial services. Journal of Corporate Accounting and Finance, 27(5), pp. 53–57. doi:10.1002/jcaf.22179

Holmes, D. (2001) eGov: eBusiness Strategies for Government. London, U.K.: Nicholas Brealey.

Kansal ,S. (2018). Python blockchain app. [Source code]. https://github.com/satwikkansal/python_blockchain_app

Kishigami J, Fujimura S, Watanabe H, Nakadaira A, Akutsu A. (2015). The blockchain-based digital content distribution system. Big Data and Cloud Computing (BDCloud), 2015 IEEE Fifth International Conference on, pp.187–190.

Linn LA , Koo MB. (2016). Blockchain for health data and its potential use in health IT and health care related research. ONC/NIST Use of Blockchain for Healthcare and Research Workshop. Gaithersburg, Maryland, United States: ONC/NIST.

Migule, Castro and Barbara, Liskov. (1999) Practical Byzantine Fault Tolerance. 3rd OSDI, 1999.

Mingxiao et al. (2017) A Review on Consensus Algorithm of Blockchain. Systems, Man, and Cybernetics (SMC). 2017 IEEE Fifth International Conference.

Nakamoto, S (2008). Bitcoin: a peer-to-peer electronic cash system, http://bitcoin.org/bitcoin.pdf

Okot-Uma, R.W. (2000) Electronic Governance: Re-inventing Good Governance. London, U.K.:Commonwealth Secretariat

Peters, G. W., and E. Panayi. (2016). Understanding modern banking ledgers

through Blockchain technologies: Future of transaction processing and smart contracts on the Internet of Money. In Banking Beyond Banks and Money, pp.239–278. New York, NY: Springer International Publishing.

Rakic, D. (2018). Blockchain Technology in Healthcare. Proceedings of the 4th International Conference on Information and Communication Technologies for Ageing Well and E-Health. doi:10.5220/0006531600130020

Shanahan, Nicole. (2016). Overcoming Information Asymmetry in Patent Pledge Records. Overcoming Information Asymmetry in Patent Pledge Records, in Patent Pledges: Global Perspectives on the Private Ordering Frontier of Patent Law, Edward Elgar Publishing Limited (Jorge Contreras & Meredith Jacob eds., Forthcoming). Available at SSRN: https://ssrn.com/abstract=2880919

Sourabh. (2014). How Much Email Do We Use Daily? 182.9 BillionEmails Sent/Received Per Day Worldwide. http://sourcedigit.com/4233-much-email-use-daily-182-9-billion-emails-sentreceived-per-day-worldwide/, February 2014.

Swan, M. (2015). Blockchain: Blueprint for a New Economy. Boston, MA: O'Reilly Media, Inc.

Tian, F. (2016). An agri-food supply chain traceability system for China based on RFID & blockchain technology. 2016 13th International Conference on Service Systems and Service Management (ICSSSM), pp. 1-6.

Transaction Rate. (n.d.). Retrieved from https://www.blockchain.com/charts/transactions-per-second

Trautman, L. J. (2016). Is Disruptive Blockchain Technology the Future of Financial Services? Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id¼ 2786186

Trillo , M. (2013). Stress Test Prepares VisaNet for the Most Wonderful Time of the Year. http://www.visa.com/blogarchives/us/2013/10/10/stress-test-prepares-visanet-for-the-most-wonderful-time-of-the-year/index.html, October 2013.

Underwood, S. (2016). Blockchain beyond bitcoin. Communications of the ACM, 59(11), 15-17. doi:10.1145/2994581

Wall Street Journal (WSJ). (2015). A Bitcoin technology gets NASDAQ test. Available at: http://www.wsj.com/article_email/a-bitcointechnology-gets-nasdaq-test-1431296886-lMyQjAxMTE1MzEyMDQxNzAwWj

Wang, X., Feng, L., Zhang, H., Lyu, C., Wang, L., & You, Y. (2017). Human Resource Information Management Model based on Blockchain Technology. 2017 IEEE Symposium on Service-Oriented System Engineering (SOSE). doi:10.1109/sose.2017.34

Yermack, D. (2017). Corporate governance and Blockchains. Review of Finance. doi:10.1093/rof/rfw074

Yue, X., Wang, H., Jin, D., Li, M., & Jiang, W. (2016). Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control. Journal of Medical Systems, 40(10). doi:10.1007/s10916-016-0574-6

Zhang, G., Li, T., Li, Y., Hui, P., & Jin, D. (2018). Blockchain-Based Data Sharing System for AI-Powered Network Operations. Journal of Communications and Information Networks, 3(3), 1-8. doi:10.1007/s41650-018-0024-3

Zyskind, G., O. Nathan, and A. Pentland. (2015). Decentralizing privacy: Using Blockchain to protect personal data. In Proceedings of the 2015 IEEE Security and Privacy Workshops (SPW), pp. 180–184. Available at: http://dl.acm.org/citation.cfm?id¼ 2867781

KOSIS(통계청, 주요서비스별 이용경험), 2018. 4. 6.

Appendix

1. The new form of civil documents

1. Digital Signature

```
from Crypto.Signature import PKCS1_v1_5
from Crypto.Hash import SHA256
from Crypto.PublicKey import RSA
import json
import hashlib
```

```
def sign(private_key, text):
  """sign the text with private key"""
  # Make dictionary to json
  text = json.dumps(text).encode('utf-8')
  # Make the document into hash
  _hash = SHA256.new(text)
  # Encrypt the hash with issuer's private key
  signer = PKCS1_v1_5.new(private_key)
  return signer.sign(_hash)


def verify(public_key, text, signature):
  """verify the signature"""
  # Make dictionary to json
  text = json.dumps(text).encode('utf-8')
  # Make the document into hash
  _hash = SHA256.new(text)
  # Decrypt the hash with issuer's private key and return whether it is True or not
  verifier = PKCS1_v1_5.new(public_key)
  return bool(verifier.verify(_hash, signature))
```

2. Define documents

1) Land registration

```
land_register = {'서류명' : '토지대장', '내용': {'고유번호' : '1144012500-10226-12345', '토지소재':'서울특별시 마포구 성산동',
                                          '지번':'123-4', '축척' : '1:600',
                                        '항목1': {'토지표시': {'지목': '08 대', '면적(제곱미터)':'144.5', '사유':'구획정리 완료'},
                                        '소유자' : {'변동일자':'1980.12.23', '변동원인' :'소유권 이전', '주소' : '서울시 행운구 행복동',
                                                    '성명': '홍길동', '등록번호':'560702-1******'}},
                                        '등급수정':{'1':{'년월일': '1983.05.16', '토지등급':'75'},
                                                    '2': {'년월일': '1984.07.01', '토지등급':'200'}},
                                        '개별공시지가' : {'1': {'기준일':'2009.01.01', '지가(원)/m2' : '3010000'}}},
                                        'DS' : b'₩xaf₩xf7₩xb6₩x1c₩xac₩tK₩x907₩xbe₩xb4₩x9b₩x06]sM₩x9b₩xOb₩xd7₩xef₩x8f₩xecL₩t₩xbf₩xb7₩xO1₩
```

2) Buliding registration

```
building_register = {'서류명' : '일반건축물대장','내용':{'고유번호':'1156011000-1-006000','명칭' :'한화금융센터',
                                                  '건축물 정보' : {'대지위치' : '서울특별시 영등포구 여의도동', '지번':'60 외1필지', '도로명주
                                                                  '대지면적(m2)':'21390', '연면적(m2)':'166,429.08', '지역':'상업지역', '지구':'주
                                                                  '용적률산정용역면적(m2)':'115,689.59', '주구조':'철골철근콘크리트', '주용도':'업
                                                                  '건폐율(%)':'49.52', '용적률(%)' : '540.66', '높이(m)':'249.58', '지붕':'스라브
                                                '건축물 현황' : {'구분':'주1', '층별':'지3', '구조' : '철골철근콘크리트조', '층면적(m2)':'1
                                                                '용도 및 면적(m2)': {'주차장':'10992.23', '기계,전기실':'3,629.25', '전시장' : '
                                                                                    '쓰레기처리장':'94.26', '휴게음식점':'63'}},
                                                 'DS' : b'₩x12g₩xacw₩xcd₩xd6₩xe8 ₩xc4Z₩r₩x94pw₩xf7₩xbO₩xfcEVK₩xd6₩xc80₩xa5₩xd84₩xac₩xc6₩xd8)

                                    }}
```

3) Resident registration

```
resident_register = {'서류명':'주민등록표(등본)', '내용':{'세대주 성명':'홍길동', '세대구성': {'1':{'세대주와의 관계':'본인', '성명': '홍
                                                                                  '2':{'세대주와의 관계':'부', '성명': '홍가나', '주민등록번호':'500
                                                                                  '3':{'세대주와의 관계':'모', '성명': '김다라', '주민등록번호':'500
                                                                'DS' : b"₩xac₩x91₩n6{₩xcfq₩xac₩xd9₩xae₩xfb5^₩xee₩x14₩x92E₩xef₩xaa₩xde₩xe6|₩xabY₩x9f₩
```

4) Certificate of local tax payments

```
localtax = {'서류명':'지방세 납세증명서', '내용':{'발급번호':'4825-231-7715-871',
                                            '납세자 인적사항':{'상호(법인명)':'한라시멘트주식회사',
                                                              '사업자등록번호':'123-45-67890',
                                                              '성명(대표자)':'홍길동',
                                                              '주민등록번호':'800101-1234567',
                                                              '주소(본점)':'강원도 강릉시 옥제면'},
                                          '증명서의 사용목적':'기타',
                                          '증명서 유효기간':'2018년 1월1일',
                                          '징수유예 또는 체납처분 유예의 내역': '해당없음', '물적납세의무 체납내역' :'해당없음'},
                                        'DS' : b'₩x9f₩x8f₩2H₩xO2₩x97:₩xc5J₩x94p₩xaf₩x1b₩xf4₩n₩x6r₩xdd|₩xc2₩x14₩x9er₩x9b₩xc4₩xc2T₩xfc
```

5) Land(forest) registration

Image to String

```
# Use base64 module to make image string
import base64
# Open and read image file
image=open('land.png', 'rb')
image_read = image.read()
# Encode the image file
image_64_encode = base64.encodebytes(image_read)
print(image_64_encode)
```


OnL841JE5adQx1Lh9K2I7X\0jN3I4KlupGvFxM/8s/DXvrsz4vuvI+E\nFq6ivouTfYdzpNaubZIMhgVL096VYzVHIQGeM3hqRentfhcJw/vpQjcdSke0NNtFhhQbI8008t9H\n3S
XZhEGw1PKusybeN+uhfD5Vuovu2s5zvbpLmfB+IPtq+vZ9qZf\yeFLt2XGCOM92nBmvOYYPhau\nR9b09uhYnhSID9/5SX5jZwcr4q\XPN06fEk34TmkKfUmffU4jR04k\+/JhUv1
R3sbL3v4StvXPQq\nacntBHFTdrHULjsvZadVo6QvTXjxZys191G2BJSgf6G+A4fGO3XAcf7wyX17e+d+6bsTusXXboe9\nQPLYfvoJep00fKIHbncQjLDAmU30pjsESjJ6DI1D73
oOzB9dsKk\mcT42jL7w4UKT7nK9H5U6\Im\nv5fDKVSbV26/KNUwsJBp053iDmM4c7bAv/rUunmZc\Bw/hpH7osCDdo40anwfn5N3Ab6st6B0fSu\niOnuj7LhTyN3fujis3XuFy1
d4Kyh6/Vqj64\jbtTv2SdnHU7dfTw4VijDBMdhwMQcUvOAh21a\+Y\nR4yrrzVaXhzChV76F/f4Pkw2vNmhEM16TtpVZ1cnxf\1LjD06QfuR/JNeA5flomRPKq70+2oQxA7\nHbVC
+KrxN77apUOMJMZtqu/ZV+hBxvaofm+5i65A+HTXQmeTcpYXBcv133bvEF2LcXX1ffQk2F33\ncVtHCIz02dG9z9zFN37v1LfTVrjtJeI+8PcPm5iE9VazCVxL83rchfouz09ease
8169c73Hda\Vo\n5i7+/XJhjE1cgNwuDX+qxN8P/Pn56V6ADy2ysUuDMflyR\P5U5zH690Gmz4jwVoafIjUOrO3wUUH\nGKfEv+8ZSoDP7LIQJsZv1LfwHpN7e3ZDnM66DfevhBjq
zJOG61zGC3ypMHL41XLZhUVo8Vxx\1eM\n66pvkvvtZYv8ix+e2qTxM3QpTcuOqoHGi5FA17pV4tk9o7AYD6C++T4wR5sul/sOXtxCjuPnuuhZ\nhozxuibSRqM6wNMTCZcxIiHIa
6fvzznMUkt98+UB3ZD\d33q55kzZP\KNnu\IU07iq200fAxEnFQ\n7DLDGPBTso4Zj6e+SyeHtoYHw4U/L03tA6xnn8dRMYx3bcaJxfA978U4+tSU5f\E5zITUsbr2hrP\ngGnJy6
AY9etQ1/iPT1LnHBH2/S\85IDIzLLym1hRhZJ+yM7tO6pmmjBGnI\kFz3mOOP/Ej6Xp4Dr\n1N3G\afDwsyopSaPkdBpOj1uMdhme39/jjtqNJXrXDd8m\d/qXB\2zJsF46qpYRJ6
HGEfC\5OX8p\nvN1S+OoOqXCnrs\kmQI2x+nCGkuYLCHHEfKVTB7wYEuURFOu3vtEGc9D793KeOZN6KsaS5iEONjP\ny3ppbtsFziRq7fiaiO0sIjyjs7\menUdEroVHPYk702tvY
RJiIP93zrgFkTr+pXrhZhNLRG5PI1n\nkxZXXOIbZpGSvXmH71aG4\nhSnsh2sG2xxE2PiHrO\b6tcIg5Es3a7Hr7rNRZgzCTghZPQo53JXH\nipSsDnZwHGEtDZ7t+JokV5iwdyj
T7mR44D5ISCISYk2Gc09qYivK9xvfH46qtnAjHX1XsUIxG\7Q\nnRDisZx\sCS+7LM5TYV3a9u4e4/H13StvhVvZNGjDdShTNIOZ06r1QTLI44z4MKF6L2+xs3Nbm27\n/MLBn7A5
iumibLijDiVkkcJt015s07Z3wjZdIsIEL\C50A2pwQD45/iI86uqeHxNsfUBS1IdLOMA\nZLi8L8X53sUr67Xvt2\q25UC7bOfqCX9FS9P+Vx7\OF7u47JwLOsRv\A8vpUpRAyuQz
325uEXHbh\njeHK+y3wiHHOAx67y67VE5VfwjuzDLe8ic/LRQqB39JvUCe3BroOmUA+IV6IrILXA+o2QZXNSieU\n4cbVN/BSQ+4isP5Mz9BpyDy4BSvbGveAdoxfSw5nkOG\Qyx6
kcLTI8KqL2QVtQVFTbu3kPVUeR7w\n8sOX9oSn7jHXDRqkX50+v871K9G5776J1xBgDGzwccaAB/OAZtO+\hs7j7o74Egdd8hDOvV7Ki5k\nACqEzKNVth7xyIijC1M3cnRH1z4pp
g7bdIOMGANb+9ItcLICpjDu9fQafn/ozjNFk94Ak1v7\Qna\npOjcHtCXeVs4Z3BTmI46U\JxBMAY8GMydZ2dDm22s16oox3zqS8gdpYIAZYEZ/KqpefhzNZkeGk4\nIRY+AH6J7A
um7URy3PhgxmztYaIpJaa+Q/VJTTn82KYezM5KOJ\bfzyXamOTq3b\L1HfUdGswieO\nSAF\iOZStx6yNJYc1D1FTrwA2PCqU5iqTOI7s7Ptcvz7Q40n\5a5KuqrZwAj71GCrtD8s
wOatTk/\nudhOLRDOgEPedg3Ddy7JNUN330zzRw4I1tzDO7P2ZeT4\3t/VyNbJ/\tCde1KZZ\ywCpIFhzz+DG\nmpiFo/kTZ5HNQ\ND7pIOfbPy9+/fX89pOHI+r23v6pobQEKMAR


```
land_image ={'서류명' : '지적도 등본', '내용' : {'발급번호':'20091019-0002-02001',
                                  '토지소재': '충청남도 홍성군 홍등면 운임리', '지번':'368-21', '축척': '1/1200',
                                 '지적도' : b'iVBORw0KGgoAAAANSUhEUgAAAoAAAAKKCAIAAAAiNHDSAAA8IUIEQVR42u3d\ZbjuJJuYXoszePM\n5Qy85I
                                  'DS' : b'o\x06\x16\xfe\x94[\xfe*\x8e\xa5\x14\xa2G9\x1c\x94A ,m\xa4\xdf<\xda\xeb\x1f\xf4\xc2]c\xc
```

String to Image

```
# Decode the image file encoded
image_64_decode = base64.decodestring(image_64_encode)
# Save the image as file
image_result = open('land_decode.png', 'wb')
image_result.write(image_64_decode)
print(image_result)
```

<_io.BufferedWriter name='land_decode.png'>

2. Code for model

1. Stage 1 : Issue

1) Key generate

```
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP
from Crypto import Random
import json
```

```
# Generate RSA key randomly
rsa = RSA.generate()

# Generate Issuer's public key and private key
private_pem_issuer=rsa.exportKey(format='PEM', passphrase='password')
with open('private_issuer.pem', 'wb') as f:
    f.write(private_pem_issuer)
public_pem_issuer=rsa.publickey().exportKey()
with open('public_issuer.pem', 'wb') as f:
    f.write(public_pem_issuer)


# Generate Sender's public key and private key
private_pem_owner=rsa.exportKey(format='PEM', passphrase='password')
with open('private_owner.pem', 'wb') as f:
    f.write(private_pem_owner)
public_pem_owner=rsa.publickey().exportKey()
with open('public_owner.pem', 'wb') as f:
    f.write(public_pem_owner)

# Generate Receiver's public key and private key
private_pem_receiver=rsa.exportKey(format='PEM', passphrase='password')
with open('private_receiver.pem', 'wb') as f:
    f.write(private_pem_receiver)
public_pem_receiver=rsa.publickey().exportKey()
with open('public_receiver.pem', 'wb') as f:
    f.write(public_pem_receiver)
```

```
# Print Sender's private key
file=open('private_owner.pem','r')
print(file.readlines())
# Print Sender's public key
file2=open('public_owner.pem', 'r')
print(file2.readlines())
```


['-----BEGIN RSA PRIVATE KEY-----\n', 'Proc-Type: 4,ENCRYPTED\n', 'DEK-Info: DES-EDE3-CBC,D72C83007IAF942FE\n', '\n', 'CJSkhqRleEfq57gusVj
EPgtir3BOBh7k1j1lugmit3QB2E7DXwjRLUQCVY9EhGV2\n', 'AlPW/z4rCVPpHmFiOe+9wurJ1S8dbSINYGyz8DFDdtFfOuQKu9v9AsVP0SSWQ6tU\n', 'oCF5TUVIXDJs8UOl
5DnDqruJxsmjxVmaD8cOjLuiDmXTnSieXeepqVqRkdwhFgTM\n', 'AO+S8zGCTBF5aE/U8wqIREyHGqKM68I5o5f9VOyuMSmkj2XxgOMm1C5TUmoNfIrk\n', 'FUqNhjnz3RTjY
ou5iKWdSp8lwUvsv7UPouy5tauOrMaQFnzDXTNKnB74noBky5/T\n', 'DHhMvMCk9sbaHDPxwg021LiGLKUbbkxXV6gH/V9x4z7zJdbpzWzmo4BFA5iJpVFT\n', '6JEXhS73bm
rAFmhdt9IlhZOMWHhbTNGRxXOcnUf74Sg3fJ24rTulAIUBAF+ps\nbPWn', '6i4FtARTQLDgXUTNSsgtiP4l+vAGDKoPw9U1gPwl\nHJyAW+bVALDOX9fk39IzyI\n', 'ZJlJv8q
zDkmWca8zayldVCoNPgSAblXsPKQvi+LTTUb8EJkg6w1SeYXb9Zzul7d3\n', 'Xyvnan+mN496EolmlLRGuKTqfYoKMD42LAUSP1vjfXujeSZk1Cp6uy7uxGrKsz7I\n', 'PAEK
x94A3GWeM7NV7byAOJlH6Bpic9IT4lPfTpye+YbwdDVqZthplgHdjYHHC76X\n', 'hdgOAjLckqKBcKqlshvyFor4GtECqJOGLzpEUbgzhdgN/soXCQXtXNH5OsE5N5Kg\n', 'Z
p3crlUh6VM2aYVX7uEvoJlAjJG+6n92RjPJiihLrBD9qWOGXGBKkV+orkp6M8s\n', 'TcojmGnEUtvkxH8zsQpqFSgKOyd+j7ORV/dAtSNHsg2SplfyDisPXitps\nu8CKl\n',
'5LHVZ1EA52fhdxma8INGGiIKZqcW/png4PZLjPv9v+3Xcrt1lTOVlB2MoOg54kNt\n', 'eVZWxJX4BMZ8UW3V4iFVpeE79DKlellNdd7VWZhXjo2ENHolukvuZbfwZp8/dPW
\n', 'XibTSLrmArqplQ9sxlcQbnlpqw5dHdkufDmFZyddkPJqbEQ1v9tLwUJXzkFgtTDl\n', 'rO/wH386hsINo78u0c9FN7l8CY7H49ZhkDC44QQ/n6vOlajT2upL47Hi8Gktc
8WE\n', '/ltWcXgiT7+f3Swb/lWExbUOfdsVvOWuJSIO/GFi4Kz6dI3Zg2K3bp+ZdnFAQh3J\n', '6HQLO+okE1P+aJiFRyqMTxCR2a+pc+rPiEo5g0XxY3sucWNWMsicNFgIS3
zf7/kp\n', '85gzVme7VL37LelFEGS3HpiD/tmWBjo59FMroipjSfrMQOkJ9MHUpjnUrAnYvvFW\n', '4t5QOP8vdmf13d8W4g37o/YO4o+isetOlcWupxTBqJ5XcJyaeJT3+In
NdD37/+9z\n', 'BnGSPbrgBtlnX3OM6+oqhTDQTrryWIUD6EzJkx3RR6b+8bQGM2B/Xd329Nbccwec\n', 'DlkEEd28+5ig/AHHeYHM67mQDVqxBxoHqMlnk96mDIPSuPlurDmd
CTdMac3Xmak7\n', '9WdrKTY8/kAzsA4ngRskx3QsVO2S7gWGNSdCrTq5xNOppGB9ELRdJG/fAVwzcgc7\n', 'GvELh7hlNyfU8usPUaJq+DZL+gamj18hNKHYf5WDy3tXFu+2V
WnnBYROXSX5mAge\n', 'aO5icG+OUNEALJH7TQTu2qCY27OEs//7l+RzuTvyolfEQum2sWEJG+zJgeF7Qxic\n', 'HLHQNvmX/NBHyRt6bkadXgRNwa53mPoFllZoJoviTM4irH
pWOo5Nnd0ifbWHV6LH\n', 'wQn/Z/lidWegDEzL2e6yO2eUg2r/b3veLiOEVqbgivE4hCly7qXdY2VbM9kEnYTu\n', 'Oc/y7E6Z8OBLL7MJ3M9ZEDxKNWLrFwr8yvLLm/kdeoE
Ly+h9cEQONBtoo2u8szUt\n', 'zL1IhtbCCiOytcc7qmp5y6grslMSDv4U/NQSJwgkk+MsdZfSKMT1oC3+bpshOlA2\n', 'e1d/cTOV7fblsFXuij7O8auKjlcRzE/EsWU+c6je
kkbjTDHGp3Ol7GjBp8ISPD8i\n', 'NbmR8cwoJ/umky3c6JiRolP3snifGFtLEszS1ay/uv8vQQgwxZOST5Q4kLH1Ys8e\n', 'dRaKg9fwjVNpM3Q+DFcKTTSbwHWZwm2jshg8
Zu4KsceT4pUabCizD7\nJ2cNAOr\n', '92QqQYLK7Pks/OJlwL4V5yhz19QHSBaOP/d0/kOwmQmIqZChOPokqsuBNCt3lMpG\n', 'enSW\nJxlGE94lKXthlolP//QZybn+i8iJ2
iSiOvOoNN79O+vzaln2Eu1TB1RXle8\n', 'RlDNJTiqPi/s8a5Nvu7mwiGF5Bbuv4evdAZolOWplvYDVc+V9xfJGFbq7poCV7hv\n', 'pxc57d3QKTuThz5bp+8eq5ixeckAt2D

2) Encryption

```
# Open Owner's public key and private key
owner_public_key_file =  open('public_owner.pem', 'r')
owner_private_key_file =  open('private_owner.pem', 'r')

# Import the key from Owner's public and private key file
owner_public_key = RSA.importKey(owner_public_key_file.read())
owner_private_key = RSA.importKey(owner_private_key_file.read(), passphrase = 'password')

# Make document in dictionary to json
resident_register['DS'] = resident_register['DS'].decode('ISO-8859-1')
document = resident_register
document_dic = json.dumps(document)

# Encrypt document with owner's public key
cipher = PKCS1_OAEP.new(owner_public_key)
encrypted = cipher.encrypt(document_dic.encode('utf-8'))

# Make encrypted document decoded to put it into BigchainDB
encrypted_real = encrypted.decode('ISO-8859-1')
```

3) Putting the data into BigchainDB

```
from bigchaindb_driver import BigchainDB
# Use the virtual network environment of BigchainDB
tokens = {}
tokens['app_id'] = '6c5569868'
tokens['app_key'] = '0ac6d515cb5b96f13c22b9347752311b'
bdb = BigchainDB('https://test.bigchaindb.com', headers=tokens)
```

```
from bigchaindb_driver.crypto import generate_keypair
# Generate Sender's keypair in BigchainDB
issuer = generate_keypair()
```

```
# Create Sender's data in BigchainDB
data_owner = {'data' : {'content': encrypted_real}}
prepared_creation_tx = bdb.transactions.prepare(
    operation='CREATE',
    signers=issuer.public_key, asset=data_owner)
```

```
# Complete Create transaction
fulfilled_creation_tx = bdb.transactions.fulfill(prepared_creation_tx, private_keys = issuer.private_key)
print(fulfilled_creation_tx)
# Send Creation transacton to Bigchaindb Server
sent_creation_tx = bdb.transactions.send_commit(fulfilled_creation_tx)
```


{'inputs': [{'owners_before': ['3TpRAVQqBwxTAzHdRXPm9rCVa3Yh6BmzAHuqhJnvo4xP'], 'fulfillment': 'pGSAICSWFst9zAr6WgSL4QaNFX4NtAlGTg-h2Nzm9MJ8PUbYgUD8SBNIrUGHsSCJSiQaUzIxyeSKcSJjoOk#yxgpAHiwNHvTr4HTYpA2I7u1f6N8cWIbnngLR-Vp7Yh_KQaYo8rQB', 'fulfills': None}], 'outputs': [{'amount': '1', 'condition': {'details': {'public_key': '3TpRAVQqBwxTAzHdRXPm9rCVa3Yh6BmzAHuqhJnvo4xP', 'type': 'ed25519-sha-256'}, 'uri': 'ni:///sha-256;MIFcKMOckDeTZ87QmlmEyKVDvtS8JEWkDqS2hS-xrzE?fpt=ed25519-sha-256&cost=131072'}, 'public_keys': ['3TpRAVQqBwxTAzHdRXPm9rCVa3Yh6BmzAHuqhJnvo4xP']}], 'asset': {'data': {'content': '¥c1cP#¥9dJ8&¥9l0_S¥x1d)ë1&¤¥9c¥x81¥%92¥8c€¥95¥%5ã ú,+è[~þ wh¥x84¥x8d<=|EQª d¥x8e§)ôàZþ ã í7¥9cúciQ¨ ÿw¥x84 j ¥xO3¥x94ÑÓ¥x8dà¥x84-¥r@¥xDesx1d]Q#ø ¨ ¥r6° 5¥x92uh½¥na.¥xO3¥x81R<¥x8eí~¥x93D¥x87¥x1bifw¥x1e¥x9d@@° ® ÷ VQ¤x84,ö¥xO3=¥x8O1¥x1d¥x94ïö¥x16 1ëš° ò Ð¥x10]lÒà!7º 8-Fjt¥9aAtE<ì¥nú¥x9eµ¥xOc¥xOb' ² ß V¥x1OIqC¤ ¿ ¥xDe¥x84}öL¥x19b<A)Ê0ù&¥x99¥x9O0¥x15^ôY~V¥x90E¥xOI þ ÿ¥x8aÀ¥x1fÙ}¬uêM?~(ck¥x120à ±à¥xOb1O7¥W¥x86ö¥x91'¥ö¥x1d þ ãE>¥xOOIP¥x1aÆ ¥xaO¥x80X¤¥x8bO¥x1açD¥x18-¥xad_zéé~ÿ?¥xaW¥x1cO¥x92)¥xOb^¥x1O¥x96Jk6Á¥x15EE n;^¥xO2(¥x94¥x05º l~ù>ÿ¥x97¥x1O¥x9ftN¥W¤Ø¥x98<}04;¶ W¥x8e0¥x07¥x81¥x87C3¥x02¥xOcT¥x86¥xd&¥x85¥x07ã¥x93+6Ex?¥x9fÃTà£¥x9cEkÉ;&,¥x83U¥x1f¥x04ëry:V¥x038ÃÁ ÝJäz¥x15¥x93&ã¥x16¥xO2¥x80:¥x93x÷è ' y¥x85!Aö¥x9e¥x84E/kt þ ¥xO5ís÷ ¥x08þ ¥hQ~Eö¥x1bhûU+|ZDéÝ,v£ ¥xO3P¨ ¬¥x98: 9¥x88¥x87ª $%¥xOci=hã¥x01Z¥x92¥x97¥xaQ&Ð m${0¥x16º k¥W¥x90¥xOcc!¥x1f¥xOI£r½,lØ L@¥x9dÉ!¥x94- ø ÿ¥x9e®¥xGd$e¥x88Þ ¥xOI¥Ã¥x11ª Ú?¥x8dQt/¥x83[à`_¥x8Oc¥x9c_¥n&{b]+³ ?5¥x88×¥x7f¥x96¥W ±¥y97¥x12®jh¥x04w8² ¥x9E¥xO8Y?ìîîÃ÷¥x85¥x83¥x90.löHoDE× Ñ! µ ¥x1c' ¤ p¥x94¥xOf¥x89¥x18!¥xOfkaÀ¤¥x1es;O9¥x1c¥+¥xOc¥x1d%Z½$?%¤ãk90Uo0X¤¨ Ã:¥xOe8JIQ¥xObvE@TÁ;Á¥x88¥x9e¥x8b±¥x96<+^Ú¥x8f+0¥xadî-¥x1eisFÃPb¥x88¥x1cç¥x1a¥x99¥xO6;(¥rÍø@¥x99¥x1d¥x7fiD¥x80îβ §ö¥w¥xOQç^¤ Lk-öyφf}è¢X¥x9a¥x17_¥xOOy¥n T¥x16¥x1c»ÒL8)C9I 9d¥x84U¥x9eO¥x07 · Úl i½¥x81® ' ¸ ø¥x19aB ' Ã¥x9f2Ei-þúfA¥x9eDÃ¸ /UCOR<¥x1eØ ¶ i¥x1cJíáj¾wñ¥WUwñNE${x¥x1Oäö¤¥xO6dSU¥x96Z ½¤ w~ˆi>¥x9dE¥x975÷vÀ ª x¥x89¥x8d+¥x91Ed[a¥xO3Z¥x8dÁV1¥xO2¥x9Z¥x9f¥x97>2 ' ¥x17¥x85¥x8ai3ñ.¥x9aot¥x8fbé¥x191i¥x13nÁMü¥x98U¥x7fi¥x91@¥xOb¥x99¿ ¸ ¥xOf¥x18§ wö¨ Bä ã :Á¥x94¥x11¥x95Þ ¥t¥x11jÈçí-¡ vç$¾¥x89Kð¥xO2ã ¥x9bu@¥x1b:L¥x81Rp E~l¥x1c¥x9dàoñL/¨ ¾n¥x8a¥x83¥x1O!pb¥OeMf¥x99J¥x9c¥x89¥x83÷ß · ¥x8d ø ¥x8a¥x83¥x8e¥x85ö ± ¥x92¥x14¥x91¥xaO61¥xaO¥x1aj^Æ }Jáµ)ýúE0' u@) ¨ ¥x13t ¡ öOu ¶ Pœ i¥x7f~ù¥x85¥f`@ ¥xOc¥xO5O0¥x80¥xO2@ ¥x15:× öe¥x97EB¥xOOœ ¬FV¥x11¥xOb@Ð X¥x07¥rÁE±¥x9fO8¥x7fp¥x95a¥Ñ'k¥x9f1¥f`Ðþ Z@O4sPŸ§ ¬y0{¥x9dœ œ r¥x8Od¥x831<¥x9B¥x13l¥x12qn¥x92¥x88¥x93A¥x85Fl±áSPF¥f`F¥W¤x85÷¥x9f §ΧÃβ Á(ú¥x91ã¥nu§ <ª à¡ç8¥x84¥x17ßSi¥x97ã6¥x16¨ 5¥xO1¥xaO2µ¡ÿ¥x12E¥x94u · Ye¥x9e÷+, .ú¥x99HUp¥xad¥x05jµk¥x99,}¨ ¥x12riL¥x97¥x19-[¥x8dQçEyñ¥x83¥x1fO¥x12gi2jÃçû§ RØ º L¥x91¥xDeÑ9 · 02OCp/n¥x1f<æ¥x8d÷¥xO7Öd.Ã6÷¤0f¥n¥x9c0¥x92à' ² ¥x81¥n¥x89¥x894A¥x8aA¥xaOQ]mlPÃ:@lO¶nówEQª dwl¥x1e¥tV«Z¥x17{Ê!#º ¨ H:¥x8dfG¤¥x12Æ ic&#?Fì · á¥x05X¥x87oZl¥n¥y¥x95¥x81+æàZ¥x8c}6O¥'p¥x94¥x18× D6¼ú;!¥r[¶ +Óyzaç¥xOco¨ ¥t£q}qv¥x11¥x94Xëöñ¿ ç¥xOb±¥x1'¥x1'¥x88¥x1d!á¥x1On]lÊ' ¥xO2o0¥xBba¥x14Blùœ£ß h¥x9f5-$º Wt!¥W¤ó¥xadæ]E¶x1E¥x8a¥x15z¥x1c}¥xOeÉ · ®ç¥xO2lO0n¤¥x83ª Á¥x95¥xOE¥xBöÁ.¥x13T¥x13O+Aç{¥x19Á' âÉ§ ¥x9c¥x900/¥x1dûA.¶g° _µlÞ 9eÈ0ç¦R§ Úvöþ :O;¨ æ¥x93r¥x1c¥x90ñ(¥x87L¥x9fB8ú{u¥x11 n¥x1eO(úji¸ bç%x9eOFçX^É¶x19¥x97¥xO8¡ Oouz¶x9bn' ú&¥x9b¥x15<ì¥x1a!/¥x9a1ö¥nz&° ö!l¥x97ÈûFA£¥x89,O0¥x166ñT.ã æ Ú¥x8e=¥x93× öF¥x04aT_¥x1eö× %£X¥x93b° bôd¥xaOU¥x17ù¿ FM¥x81w¥x8f¥x1b¥xDe`f@ 6u · Dà+¥x87¨¥x9f¥xO1¥xaO¡ ¥xO1µâ=¨¥x93IOÁó² i¥x9f¨O¥tcX¦ í&~ó¥x12à¥¥Caj!ø=¥x16¥¥r¥x15ã¥x1O0÷Æ pú¥xO6ã:Þ Ä@cE+¥xaO¥x9e¥xO8r-]Ai¥x9Oþ DíÉæXûfM¥x97Ð 7la¥x98æ ± ¶ §¥xaOR¤ ¥xOOœ i¥x11oOp¥' ® ¶ ¿ qúVîµ¥x9O¥x8c.¥xBciöE¥xöU¥x18¥n¥x85ë¥¼ó~sEMp¡ ¥x9bÉ@ AuP§ @¥x8fá¥x1f8¥xaOriSØ#w¥x8aX] íéaÄ¥x8fml[æ|+»öY ' ;æ|ê¥x8cU¥x9em||œ 2þ ¥x17¥x04¥xOO¥x15afu¥xO3[¥x9l8E<Ú¥xOb9@ r¥x88÷yd`i¥x8OI¥x908{m÷wa?SœE¥x9a¥x1O¥x19° ¥x9eK,É¥xO4¥x9O¥x05OÜ@,¥x93¥x95× ¼@ +¥x18à¥xBb¥x9a¥x92Ri¥x7fµmÚKõª ùÀ¥x7fÑø µ¥x9a¥xlf¨ ÒH&(ë¥x16þ |¶tqîrÜà' ¥x89¥x1c¥xO2¥x93¥oï=º tþ ° Y±j_ë#'¥x8e÷+ÀM,|,r¥x96@ $¥xO2¥x98¥x87(1Æ îëB"° ú3¥x9béyM]× ÞOKíóaOæ.¥x96uëã+xT÷ ¥x12h"¥ó¥MP¥xOfuëiS¥x1dC¥x9cò¥xBbK÷e¥x9b'¥xad´ ?ç¥x90¥nwsÐ S¥x87u¡óü¥x9d¥x86Þ x¥x18u`¥x11ï¿@¥x15+w¥x1a¥x8fat¶>· C%× ¥x9eOu¥x19ehg4v+H¥xO8¥x7f¥x1b¥x87¥x1ea¶¥x89{îL?² t¥xO1 úîñSÐ÷w¥9Oç#ëçkEF¥xBe¥x1OE¥W · ñ-Ý+D6{HT÷ß¥x17¥x1a¥x9búU¥x05îò1¥xO8' ¥x7fÉ{¥x04¥x9e¥x1O¥x04¥xOe¥x92O¥xBeáßë_l+¼÷ ¥x94çî ª ¥x16uf,<¥x8O· ¥xldôOehed¥¥f' ¥xO3¥xBeG² ç¥xaOw¥x98æ î¡ ¤ ¥x95¥x9c_.Á¿îíxã° }Vx<¥x83ò¶ O¥x19u6_p¥xBb¥x10Á° MÓ÷ µqe<¥xO3¥x1dç¥x1aiL`J¦ '7ã¡¥x17ã4¥x87-t¥x89¥x9bŷ¥xOOdB¥x1b!¥xaOQAaÄ¥x8c:d[á<ö¥x9c¥x13ø òY¥x8530pQ$¡ µmoÏî° ºa¥x81äÿéjGñçî¨ ¥x83E¥x11¥x850¥x12à¥x9O¥x87ö¥íR0.`f¥x1eGíÕy¥x91ã @Gnv ø®ù:¸ ,-¥xOf¥xO6u4zH8=îáJ¥x89O¥x17(° ¥x90Ái¥xDeáE¥x1cT · Yh¥¥¥x13lø ¥x8OD¥x87ã¥xOej¥x18¥xadSaGblñø ÿ¥x9c¥xOD-¥x97¥xada7ð {¥'¥x05¥x8O¥ã ã?ÈO¥x87¥xOcmÄ¥x1e9X¥xO2Gã¥xO6½¥x14ø-ú[Jb¥xadφß ^ØG>Ø0îº¿ ° EB¥x93¥x1f¥x8b¥xO3E!¥x1a¥x86¥x8e§ <÷ s0¥xOf N¥x87· ¥tëT¥r.¥x85yZ¥x99Oq,ÆÏ3¥W¤xO2x¥h¥x9a¥x8eüÙ · çhPu-4S¥x14¥x9dµ¥x10¥xOOEN×¥9btW¤xad¥x11>(1É½ME,öU¥x7fç æ ¡ >ú¡¥xO8Æ¥x9df¡¥x8dëó¥x82ÚX¨ ~r¥x13ã£ßI V° +·¥x16W¤99Huµ¾ãi¥xBO¥0|Á ' ¥x9e· ŸGã± ¬i ' ¡¥x82¥xad¥xOc¥xO6¥Þ Ú² èÓª¥x14¥xO2Á^Av¥V9Ab æ¥x13¥xOIã¥x850¥' ¥x96¥x96ÿ¥x840¥x14<Æûr² ðé¥xaOEnhÄtL¤ _¥x18÷<¥x82gÁ² ¥x91Ô÷¥x17b¥x88ø i¥x1c3Ÿ³ L0¥x9fúk' ª ¥x8d³ i£Gµ#x1O§¥xO2<e3#¨^¥x9b¿:÷é¥¥O¤x89ä¥x8fÁ¥m¥x9b¥x11ó ª éCGÀ¤x1d¥x82ç¥xO7ê¥x98¥x96`=p ë§kL¥x04i¥xOfç~æ À¥¨§Ÿ¥x1c¥x1dîT¥xO3Œ¥x87¥x8O-pá?¥x19¥x7f5¥x9cv¼ Ø lÿczP8P6½+¥xO1×¥'ù¼¥xO4¥x16TnéN`dW\>æ 6i¥'Þ æ}(ö¥¥÷ . Ð 5Z¥x9eiO¥x07Ø¥x12÷Üöèà¼¥x06 rÅ ¶¥x16¥x95¥wÁø ¥t &o¸ ÿ¥x88¥xOOUR{¥x95¥x9d6e`ì~¥x8e¸ +sZ)e÷¥x1c÷ ¥x1e¥x8fó¢÷¥x92¥xOb¥x7fS¥x95Á?÷ ó¥x98E¥rX½¥x16@Dov!F¥xad¥x88ý®Oø° %1E¥x89Ø¥x9e¥x91ôß FD/¥x8fy+ltýn¥/¸Ú¸ ¥x994¥x8c>¶ ¶¥nÁÐ3¥x8d§æ Ä¥¥>SÙÄ¥x83ûµß¥x12,¥x8ag¥x95Ø&ah¥x87¥xadîBÁ}æ¥x86tú¿ ¥x140Ð ð wNi6iEXgç i¥x8fã Ú¥x6bD² ún¥x19¥x7fñÁïú¥x86]æ¥x8f æ¥x95´ -¥x95¥xld[ÚTãÄOi¥x8aOw¥x9ae¥x8a¥x1eô¥x13dáLQ¢_¥x9e_¥xOI¤¥x93¥x92' moSZ¥x13=f¥xO8jzdiQU¥rqPçwEã¤ ·}}, 'operation': 'CREATE', 'metadata': None, 'id': '2572377c42cbac289e53c3e0a561135344b7a3032466cde7f2b953f5fb61a1cf6c', 'version': '2.0'}

2. Stage 2 : Exchange

1) Encrypt the data with Receiver's key

```
# Pull data in bigchaindb
encrypted_1 = '₩x1cR₩x9dJ8&₩x91ô_S₩x1d)ælš0₩x9c₩x81₩x92₩x8c4₩x95E₩x95ä û,+è[~þ wh₩x84₩x8d<=|ÉQ¤ d₩x8e§ }ôAZþ ä í7₩x9cûcìQ¯ ÿw₩x84¡₩xC3₩x9
encrypted_2 = encrypted_1.encode('ISO-8859-1')

# Decrypt the data with private key of host
private_key = PKCS1_OAEP.new(owner_private_key)
message = private_key.decrypt(encrypted_2)


owner_public_key_file.close()
owner_private_key_file.close()

# Open the receiver's public key and private key
receiver_public_key_file =  open('public_receiver.pem', 'r')
receiver_private_key_file =  open('private_receiver.pem', 'r')

receiver_public_key = RSA.importKey(receiver_public_key_file.read())
receiver_private_key = RSA.importKey(receiver_private_key_file.read(), passphrase = 'password')

# Encrypt data with receiver's public key
re_publickey = PKCS1_OAEP.new(receiver_public_key)
encrypted_receiver = re_publickey.encrypt(message)
```

2) Blockchain for Exchange

```
from hashlib import sha256
import json
import time

from flask import Flask, request
import requests
```

```
class Block:
    def __init__(self, index, data, timestamp, previous_hash):
        self.index = []
        self.data = data
        self.timestamp = timestamp
        self.previous_hash = previous_hash
        self.nonce = 0

    # Compute block's hash
    def compute_hash(self):
        block_string = json.dumps(self.__dict__, sort_keys=True)
        '''
        hexdigest : Change hash to string
        encode: Change json to python
        '''

        return sha256(block_string.encode()).hexdigest()

```

```
class Blockchain:
    def __init__(self) :
        # Transactions not confirmed
        self.unconfirmed_transactions = []
        self.chain = []
        self.create_genesis_block()

    def create_genesis_block(self):
        genesis_block = Block(0, [], time.time(), "0")
        genesis_block.hash = genesis_block.compute_hash()
        self.chain.append(genesis_block)

    def last_block(self):
        return self.chain[-1]

    # Difficulty of mining
    difficulty = 5

    # Proof of Work
    def proof_of_work(self, block):

        block.nonce = 0
        computed_hash  = block.compute_hash()


        while not computed_hash.startswith('0' * Blockchain.difficulty):
            block.nonce +=1
            computed_hash = block.compute_hash()

        return computed_hash

    # Add block to Blockchain
    def add_block(self, block, proof):
        previous_hash = self.last_block.hash

        # Return false if previous block's hash is not the same with last block's hash
        if previous_hash != block.previous_hash:
            return False

        # Return false if block is invalid
        if not self.is_valid_proof(block, proof):
            return False

        block.hash = proof
        self.chain.append(block)
        return True
```

```

    # Check whether block is valid or not
    def is_valid_proof(self, block, block_hash):
        return (block_hash.startswith('0' * Blockchain.difficulty) and
                block_hash == block.compute_hash())

    # Add transactions not confirmed to list
    def add_new_transaction(self, transaction):
        self.unconfirmed_transactions.append(transaction)


    # Mining function
    def mine(self) :
        if not self.unconfrimed_transactions:
            return False

        last_block  = self.last_block

        new_block = Block(index = last_block.index +1,
                          transactions = self.unconfirmed_transactions,
                          timestamp = time.time(),
                          previous_hash=last_block.hash)

      # Add block after PoW
        proof = self.proof_of_work(new_block)
        self.add_block(new_block, proof)

     # Make list for unconfirmed transaction empty
        self.unconfirmed_transactions = []
        return new_block.index

```

3) Application for Blockchain

```python
app = Flask(__name__)


blockchain = Blockchain()


peers = set()



@app.route('/new_transaction', methods=['POST'])
# Accept new transaction
def new_transaction():
    tx_data = request.get_json()
    required_fields = ["author", "content"]

    for field in required_fields:
        if not tx_data.get(field):
            return "Invalid transaction data", 404

    tx_data["timestamp"] = time.time()

    blockchain.add_new_transaction(tx_data)

    return "Success", 201



@app.route('/chain', methods=['GET'])
# Get the longest chain
def get_chain():

    consensus()
    chain_data = []
    for block in blockchain.chain:
        chain_data.append(block.__dict__)
    return json.dumps({"length": len(chain_data),
                       "chain": chain_data})


```

```python

@app.route('/mine', methods=['GET'])
# Mining unconfirmed transaction and add to blockchain
def mine_unconfirmed_transactions():
    result = blockchain.mine()
    if not result:
        return "No transactions to mine"
    return "Block #{} is mined.".format(result)



@app.route('/add_nodes', methods=['POST'])
# Register new node to network
def register_new_peers():
    nodes = request.get_json()
    if not nodes:
        return "Invalid data", 400
    for node in nodes:
        peers.add(node)

    return "Success", 201



@app.route('/add_block', methods=['POST'])
# Collect transaction and add block to chain
def validate_and_add_block():
    block_data = request.get_json()
    block = Block(block_data["index"],
                  block_data["transactions"],
                  block_data["timestamp",
                  block_data["previous_hash"]])

    proof = block_data['hash']
    added = blockchain.add_block(block, proof)

    if not added:
        return "The block was discarded by the node", 400

    return "Block added to the chain", 201



@app.route('/pending_tx')
# Before making block, define the transaction
def get_pending_tx():
    return json.dumps(blockchain.unconfirmed_transactions)
```

```
88
89
90  # Peer check whether mined blockc is the longest(Consensus)
91  def consensus():
92
93      global blockchain
94
95      longest_chain = None
96      current_len = len(blockchain.chain)
97
98      for node in peers:
99          response = requests.get('http://{}/chain'.format(node))
100         length = response.json()['length']
101         chain = response.json()['chain']
102         if length > current_len and blockchain.check_chain_validity(chain):
103             current_len = length
104             longest_chain = chain
105
106     if longest_chain:
107         blockchain = longest_chain
108         return True
109
110     return False
111
112
113 # Announce the new block to others
114 def announce_new_block(block):
115
116     for peer in peers:
117         url = "http://{}/add_block".format(peer)
118         requests.post(url, data=json.dumps(block.__dict__, sort_keys=True))
119
120
121 app.run(debug=True, port=8000)
```

4) Decrypt the data with Receiver's key

```
1  # Encrypted document in bytes
2  encrypted_3 = b'Gx₩xda₩xec₩xad₩x86H₩xa6SDi₩xd8R₩x14₩x82m;$|₩xb3₩xb6₩'K₩xc5₩x82₩xc2₩xf2~₩xb7₩x8c₩x1a₩xb4!₩x9d₩xea₩xb1₩x98₩xca₩xe2₩xce₩xb8
3
4  # Decrypt with receiver's private key
5  cipher_receiver = PKCS1_OAEP.new(receiver_private_key)
6  decrypted = cipher_receiver.decrypt(encrypted_3)
7
8  # Decode with the UTF-8 to see the actual documents
9  received_document = decrypted.decode('utf-8')
10 received_document_1=json.loads(received_document)
11 received_document_1['DS'] =received_document_1['DS'].encode('ISO-8859-1')
12 print(received_document_1)
13
```


{'DS': b"₩xac₩x91₩n6{₩xcfq₩xac₩xd9₩xae₩xfb5^₩xee₩x14₩x92B₩xef₩xaa₩xde₩xe6|₩xabV₩x9f₩tY₩xaal₩x85go₩x1e|₩xf8₩xfa₩xa3₩xcf8₩xbe₩₩xcO₩xa7M@₩x1c₩x9e17₩x1a@₩xf5{/l₩x1dA₩xb0<₩xcb₩x7f₩xcc'₩xd3₩x81₩xab₩xb2₩xb8₩xa3#|T₩xa9₩x17₩₩₩x04₩x9cI7k₩x0cP₩x9a /₩dc₩xb3₩x16₩xcf₩x9e₩xf00A₩xbeT₩x1c|₩xfBM₩xaO₩xf8a₩x9cEE₩x9d₩x19₩x8a₩xee₩xaa<'e₩xb9-₩xfa₩xd0g;₩tt₩xd₩xe3D=₩xd2₩x99 w(₩xde₩x17L₩xfc₩x1b₩xaO₩x6b₩xfa₩₩xd0₩xaa₩xb4[₩x68₩xe9₩xe2u₩xbf₩x10₩x1c/~/₩xd3₩x94₩xO3₩x1f₩x1e₩xee₩xf7₩x98₩x125₩x1f₩xbO9+=₩xba₩xb6₩xab₩x8a ₩xd1₩x16₩xa5₩x15t.₩xb₩xbfV-,O₩xOf}₩x8b:#₩xd2₩xe2X₩x8e₩xd2₩xO7₩xbO₩x84₩xa5<G₩x85X((₩xb5,₩rmm₩t!₩xe3₩xdO§c₩xO3₩xea:₩x95₩xOc₩x9fm₩x8dn₩x90₩xbO#₩xc2lGu)]₩x9O₩x89₩x12₩x1a₩xeel₩x9c₩t₩xdch₩xce₩xOe?₩x12₩x89;₩x95;₩xca₩xe2", '서류명': '주민등록표(등본)', '내용': {'세대구성': {'3': {'세대주와의 관계': '모', '주민등록번호': '500101-2234567', '등록상태': '거주자', '성명': '김다라'}, '1': {'세대주와의 관계': '본인', '주민등록번호': '800101-1234567', '등록상태': '거주자', '성명': '홍길동'}, '2': {'세대주와의 관계': '부', '주민등록번호': '500101-1234567', '등록상태': '거주자', '성명': '홍가나'}}, '세대주 성명': '홍길동'}}