FOUR-CHAINS

Crypto Blockchain

New generation blockchain based on smart contracts

White Paper

Author: *Yujeong Lee, Minji Yoon, Minji Shim, Junghoon Lee*

Team:	Cryptographic Research Team
Department:	Research and Development Center
Date:	July 24, 2023

Contents

1 Introduction

In this document, we would like to introduce 'Crypto Blockchain' and 'Strongbox', which we have researched and developed ourselves. First, the Crypto Blockchain deals with mathematical approaches and various cryptography, so it was set up under the name **Crypto**, and unlike the existing blockchain, it has the following characteristics.

- Block inscribe methodology
- New consensus algorithm using ECDLP
- Block broadcast methodology on the network
- Individual identification using isomorphic structure
- Smart contract based service

From transaction details to create blocks, various encryptions were applied to secure both security strength and lightness. In order to apply various cryptography, methodological research on writing blocks was carried out in principle and included in this document. In addition, as a result of investigating and researching existing consensus algorithms, it was determined that there were some unfortunate parts such as a model that creates a gap between rich and poor through computing power, a model that may cause problems mathematically, and a meaningless formal model, as well as difficult parts to set up a suitable model for the Crypto Blockchain being developed by this research team. Considering that it is very suitable for developing a consensus algorithm based on good problems that mathematical researchers are dealing with, but considering that it is difficult to look at application cases, it was decided that this research team should directly research and develop a suitable consensus algorithm through a mathematical approach. The research team used the following criteria to research and develop the consensus algorithm.

- Difficulty can be adjusted
- The problem of having a running time suitable for the company's blockchain because there is an appropriate level of difficulty suitable for the environment
- A problem that matches our concept, that is, a problem that requires mathematical knowledge related to cryptography
- Problem solving is not proportional to computing power

The mathematical material chosen by this research team to develop the consensus algorithm was by far a mathematical problem related to elliptic curves, and among them, **'Elliptic Curve Discrete Logarithm Problem'** was judged to be the most appropriate. Difficulty can be adjusted while generating an elliptic curve, and it is very suitable for the running time to solve this problem. In addition, a strong research-based consensus algorithm was developed and applied, considering that solving a problem is not proportional to computing power in that finding a solution can vary depending on the selection of a primitive root (generating source) while performing simple iterative calculations through random selection of primitive roots. In addition, a personal identification system was established through a cryptographic approach, and details

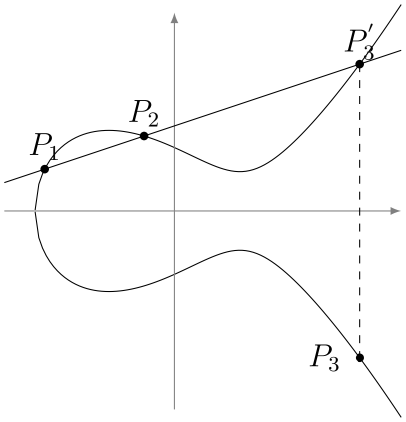


Figure 1: Adding points on an Elliptic Curve

were included in the text. Finally, the crypto blockchain developed by the company constitutes a blockchain structure based on smart contracts and aims to service it.
To this end, the function of personal information wallet and functional elements that can write smart contracts have been attached to Strongbox to prepare a stepping stone for servicing. In order to deal with the contents of the main text, information related to elliptic curves was referred to [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12].

2 Background

2.1 Elliptic Curve Group structure

Consider an elliptic curve $\mathbb{E}$ defined over a field $\mathbb{Q}$ of rational numbers. Consider the two points $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ on an elliptic curve $\mathbb{E}$ given by $y^2 = x^3 + Ax + B$ with $P_1, P_2 \neq \infty$. We can obtain a third point P_3 on $\mathbb{E}$ as follows. First, we draw the line L through P_1 and P_2. (If $P_1 = P_2$, the line L is the tangent line at P_1.) The line L intersects $\mathbb{E}$ in a point $P_3^{'}$. Reflect $P_3^{'}$ across the x-axis to get $P_3 = (x_3, y_3)$. Finally we define the group law as follows:

$$P_1 + P_2 = P_3 \tag{1}$$

1. If $x_1 \neq x_2$, then $x_3 = m^2 - x_1 - x_2$, $y_3 = m(x_1 - x_3) - y_1$, where $m = \frac{y_2 - y_1}{x_2 - x_1}$.

2. If $x_1 = x_2$ but $y_1 \neq y_2$, then $P_1 + P_2 = \infty$.

3. If $P_1 = P_2$ and $y_1 \neq 0$, then $x_3 = m^2 - 2x_1$, $y_3 = m(x_1 - x_3) - y_1$, where $m = \frac{3x_1^2 + A}{2y_1}$.

4. If $P_1 = P_2$ and $y_1 = 0$, then $P_1 + P_2 = \infty$.

Moreover a line through ∞ and P is vertical. It intersects $\mathbb{E}$ in $P = (x, y)$ and $(x, -y)$. Reflecting the point of P across the x-axis, we get the point $(x, -y)$. Therefore, we can get

$P+\infty = P$. Also, a line through (x, y) and $(x, -y)$ is vertical, so the point of intersection with $\mathbb{E}$ is ∞. Therefore, we get $(x, y) + (x, -y) = \infty$. Since ∞ is a role of identity on $\mathbb{E}$, we define $-(x, y) = (x, -y)$ that is, $(x, -y)$ is inverse of (x, y).
We now consider the Elliptic Curves over $\mathbb{F}_p$. Let $\mathbb{F}_p$ be a prime finite field so that p is an odd prime number, and let $a, b \in \mathbb{F}_p$ satisfy $4A^3 + 27B^2 \not\equiv 0 \pmod{p}$. Then an elliptic curve $\mathbb{E}(\mathbb{F}_p)$ over $\mathbb{F}_p$ defined by the parameters $A, B \in \mathbb{F}_p$ consists of the set of solutions or points $P = (x, y)$ for $A, B \in \mathbb{F}_p$ to the equation:

$$y^2 \equiv x^3 + Ax + B \pmod{p} \tag{2}$$

Together with an extra point $\mathcal{O}$ called the point at infinity. The equation $y^2 \equiv x^3 + Ax + B \pmod{p}$ is called the defining equation of $\mathbb{E}(\mathbb{F}_p)$. For a given point $P = (x_p, y_p)$, x_p is called the x-coordinate of P, and y_p is called the y-coordinate of P. The number of points on $\mathbb{E}(\mathbb{F}_p)$ is denoted by $\#\mathbb{E}(\mathbb{F}_p)$. **'Hasse Theorem'** introduce how to calculate $\#\mathbb{E}(\mathbb{F}_p)$ as follows:
If $\mathbb{E}(\mathbb{F}_p)$ is elliptic curve over finite field $\mathbb{F}_p$ then

$$p + 1 - 2\sqrt{p} \leq \#\mathbb{E}(\mathbb{F}_p) \leq p + 1 + 2\sqrt{p} \tag{3}$$

The group law over $\mathbb{E}(\mathbb{F}_p)$ is as follows:

1. Rule to add the point at infinity to itself:

$$\mathcal{O} + \mathcal{O} = \mathcal{O} \tag{4}$$

2. Rule to add the point at infinity to any other point:

$$(x, y) + \mathcal{O} = \mathcal{O} + (x, y) = (x, y) \text{ for all } (x, y) \in \mathbb{E}(\mathbb{F}_p) \tag{5}$$

3. Rule to add two point with the same x-coordinates when the points are either distinct or have y-coordinates 0:

$$(x, y) + (x, -y) = \mathcal{O} \text{ for all } (x, y) \in \mathbb{E}(\mathbb{F}_p) \tag{6}$$

4. Rule to add two point with different x-coordinates: Let $(x_1, y_1) \in \mathbb{E}(\mathbb{F}_p)$ and $(x_2, y_2) \in \mathbb{E}(\mathbb{F}_p)$ be two points such that $x_1 \neq x_2$. Then $(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$, where:

$$x_3 \equiv \lambda^2 - x_1 - x_2,\ y_3 \equiv \lambda(x_1 - x_3) - y_1,\ and\ \lambda \equiv \frac{y_2 - y_1}{x_2 - x_1} \pmod{p} \tag{7}$$

5. Rule to add a point to itself(double a point): Let $(x_1, y_1) \in \mathbb{E}(\mathbb{F}_p)$ be a point with $y_1 \neq 0$.

$$x_3 \equiv \lambda^2 - 2x_2,\ y_3 \equiv \lambda(x_1 - x_3) - y_1,\ and\ \lambda \equiv \frac{3x_1^2 + A}{2y_1} \pmod{p} \tag{8}$$

The set of points on $\mathbb{E}(\mathbb{F}_p)$ forms a group under this addition rule. Furthermore, the group is abelian group. (meaning that $P_1 + P_2 = P_2 + P_1$ for all point $P_1, P_2 \in \mathbb{E}(\mathbb{F}_p)$). Given an integer k and a point $P \in \mathbb{E}(\mathbb{F}_p)$, scalar multiplication is the process of adding P to itself k times. The result of this scalar multiplication is denoted kP.

2.2 Elliptic Curve Discrete Logarithm Problem, ECDLP

Let $\mathbb{E}(\mathbb{Z}_p)$ be an elliptic curve in the finite field $\mathbb{Z}_p$ and say $P \in \mathbb{E}(\mathbb{Z}_p)$. If $Q \in \mathbb{E}(\mathbb{Z}_p)$ is a point obtained by calculating P multiple times, the problem of finding m that satisfies the following is called **'Elliptic Curve Discrete Logarithm Problem, ECDLP'**.

$$Q = mP \tag{9}$$

At this time, if expressed through a Discrete Logarithm, it is as follows.

$$m = Log_P(Q) \tag{10}$$

The Elliptic Curve Discrete Logarithm Problem(ECDLP) is being used in attacks on elliptic curve cryptosystems(ECC). In fact, in the process of implementing the elliptic curve cryptography and the encryption process, text information is converted through ASCII code and $P \in \mathbb{E}$ is calculated as much as ASCII code value.

2.3 Elliptic Curve Homomorphic Cryptosystem, ECHC

We would like to use **"Elliptic curve Group structure"**. Considering ECC, first, the plain text m is expressed as one point M , Q is the receiver's public key, and k is the encrypted result kQ calculated when randomly selected on an integer. The sender can send the points $C_1 = kP, C_2 = M + kQ$ to the receiver using their private key and calculate as follows.

$$dC_1 = d(kP) + k(dP) = kQ \tag{11}$$

And $M = C_2 = kQ$ can be obtained. According to Weil's pairing, the group structure of the elliptic curve $\mathbb{E}(\mathbb{F}_p)$ is always either a cyclic group or isomorphic to the direct product of two cyclic groups. If the structure of $\mathbb{E}(\mathbb{F}_p)$ is isomorphic to the cyclic group, $\mathbb{Z}_1$ can be used to express the direct product of two cyclic groups. Also, consider the isomorphic function σ defined as:

$$\sigma : \mathbb{E}(\mathbb{F}_p) \longrightarrow \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \tag{12}$$

We name this isomorphic function **'EI function'**.

We developed a new cryptographic scheme using Weil's paring and denote the cryptosystem as $\mathbf{E}$. The cryptographic scheme $\mathbf{E}$ is composed of four algorithms $\mathbf{KeyGen}_E$, $\mathbf{Enc}_E$, $\mathbf{Dec}_E$, and the isomorphic expression $\mathbf{EI}_E$ with public key $\mathbf{pk}$. Also, $\mathbf{EI}_E$ is an expression for the EI function σ.

■ **List of other notations**

1. $\mathcal{P}$: a set of plain texts.

2. $\mathcal{C}$: a set of cipher texts.

3. $\mathcal{R}$:=$\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$.

4. $\mathcal{C}_{\mathcal{R}}$: a set of cipher texts on $\mathcal{R}$.

5. $\mathbf{rep}_E$: message $\mathbf{m} \in \mathbf{P}$ expressed as a point $\mathbf{M}$ on elliptic curve $\mathbb{E}$.

6. C_1=k$\sigma(\mathbf{P})$, C_2=$\sigma(\mathbf{M} + k\mathbf{Q})$
7. $\widetilde{C}_1$ = σ^{-1} (C_1), $\widetilde{C}_2$ = σ^{-1} (C_2), $\widetilde{C}$=$\left(\widetilde{C}_1, \widetilde{C}_2\right)$ are cipher texts ordered pairing.
8. Σ_p : a set of **EI function** over $\mathbb{E}(\mathbb{F}_p)$.

If, for a key pair (pk, sk) output by $\mathbf{KeyGen}_E$, the plaintext is $M \in \mathcal{P}$ and the ciphertext is $C \in \mathcal{C}$, it means the following case:

$$C \leftarrow \mathbf{Enc}_E(M), \psi \leftarrow \mathbf{EI}_E(C), \mathbf{EI}_E(M) = \mathbf{Dec}_E(\psi)$$

We will refer to the aforementioned cryptographic system as **Elliptic Curve Homomorphic cryptosystem**, short for **ECHC**.

2.3.1 Key generation scheme of elliptic curve homomorphic encryption system(ECHC key generation scheme)

Since the elliptic curve homomorphic encryption system (ECHC) is also constructed on the elliptic curve, the elliptic curve homomorphic encryption system (ECHC) is constructed in the same way as the key generation of the elliptic curve cryptosystem (ECC).

Algorithm 1 Key generation algorithm of elliptic curve homomorphic encryption system(ECHC key generation algorithm)

```
    Input: Domain parameters of elliptic curve homomorphic encryption system $(\mathbb{E}, n, p, P)$
    Output: Public key $Q$ and private key $d$
Choose $d \in [1, n-1]$
Calculate $Q = dP$
Return $(Q, d)$
```

Through this key generation, the private key d can be obtained, and it becomes an important part of defense against attacks. d is a prime number less than p, generated by a random number generator. P is already known, and Q is released after key generation. However, it is very difficult to infer the value of d from only P and Q. This is called the elliptic curve discrete logarithm problem (ECDLP). Because of these characteristics, it has been used as a public key cryptography.

2.3.2 Encryption scheme of elliptic curve homomorphic encryption system(ECHC encryption scheme)

Encrypt message m using generator P. In a cyclic group, any element of the group can be created. Then $Q = dP$ is calculated for the randomly selected $d \in \{1, \ldots, n-1\}$, and the result of the cryptographic algorithm is the ciphertext pair $\widetilde{C} = (\sigma^{-1}(C_1), \sigma^{-1}(C_2))$. There are algorithms for elliptic curve homomorphic encryption systems. Actually, $\widetilde{C} = (\sigma^{-1}(C_1), \sigma^{-1}(C_2))$ is expressed as $\widetilde{C} = (\widetilde{C}_1, \widetilde{C}_2)$, and maintained as $\widetilde{C}_i = \sigma^{-1}(C_i)$ $(i = 1, 2)$.

Algorithm 2 Encryption algorithm of elliptic curve homomorphic encryption system(ECHC encryption algorithm)

```
**Input : Domain parameters of elliptic curve homomorphic encryption system** $(\mathbb{E}, n, p, P)$**, Public key** $Q$**, plaintext** $m$
**Output : ciphertext** $(\widetilde{C}_1, \widetilde{C}_2)$
Express the message $m$ as a point $M$ on the elliptic curve $\mathbb{E}(\mathbb{F}_p)$.
Choose $k \in [1, n-1]$
Calculate $C_1 = k\sigma(P)$
Calculate $C_2 = \sigma(M + kQ) = \sigma(M) + k\sigma(Q)$
Return $(\sigma^{-1}(C_1), \sigma^{-1}(C_2)) = (\widetilde{C}_1, \widetilde{C}_2)$
```

In this encryption process, it is not necessary to actually present the EI function, but it is presented through the EI function for the isomorphic relationship.

2.3.3 Decryption scheme of elliptic curve homomorphic encryption system(ECHC decryption scheme))

Like the key generation process, the decryption process of the elliptic curve homomorphic encryption system is configured similarly to the decoding process of the elliptic curve cryptosystem (ECC). The decryption process is performed using the private key sk. Here, sk is the secret key obtained through the key generation process (Algorithm 1).

Algorithm 3 Decryption algorithm of elliptic curve homomorphic encryption system(ECHC decryption algorithm)

```
**Input: Domain parameters of elliptic curve homomorphic encryption system** $(\mathbb{E}, n, p, P)$**, Public key** $Q$**, ciphertext** $(C_1, C_2)$
**Output: plaintext** $m$
Calculate $\sigma(M) = C_2 - dC_1$
Get $m$ from $\sigma(M)$
Return $m$
```

It should not be confused that it is the ciphertext of $\mathcal{R}$ and not the ciphertext of the elliptic curve.

2.3.4 Properties of elliptic curve homomorphic encryption system(Properties of ECHC)

Finally, the elliptic curve homomorphic encryption system (ECHC) have isomorphic properties. These properties are constructed using the EI function, whose expression is $\mathbf{EI}_E$.

$$\mathbf{EI}_E(m_1 * m_2) = \sigma(m_1 * m_2) = \sigma(m_1) + \sigma(m_2) = \mathbf{EI}_E(m_1) + \mathbf{EI}_E(m_2) \tag{13}$$

Roughly expressed, $\sigma = \mathbf{EI}_E$, but strictly expressed as:

$$\mathbf{EI}_E(\widetilde{C}) = \psi(\widetilde{C}_1, \widetilde{C}_2), \tag{14}$$

$\psi : \mathbb{E}^2 \longrightarrow \mathcal{R}^2$ is an idea that satisfies $\psi(\widetilde{C}_1, \widetilde{C}_2) = (\sigma(\widetilde{C}_1), \sigma(\widetilde{C}_2))$. Through this property, it has an isomorphic structure on an elliptic curve. In fact, we use Weil's pairing to construct these EI functions. Therefore, the elliptic curve-based homomorphic encryption developed by Fourchains is a homomorphic-based encryption using Weil's pairing. Strictly speaking, it is an encryption scheme different from the existing homomorphic encryption structure, and we call it **Pseudo Homomorphic Cryptosystem**.

2.4 Mathematical Approach to Blockchain Definition

Our research team aims to research and develop next-generation blockchains for storing and sharing data and to develop an electronic authentication system using them. In order to research and develop next-generation blockchains, we would like to define concepts not covered in general blockchain research, and to this end, we plan to generalize the concept of blockchain through a mathematical approach by proceeding with the axiomatization of blockchain.

Definition 1. ***Node***
Let's define a single actor in the system as ***node*** *and denote it as $n_i (i = 0, 1, 2, ..., k)$. In addition, if the set of nodes is shortened and named as a node set and expressed as $\mathcal{N}$, the following is established.*

$$\mathcal{N} = \{n_0, n_1, ..., n_k\} \tag{15}$$

Specifically, on a computer network, computers are nodes, and in the traditional client-server model, both clients and servers can be viewed as nodes. Conversely, these nodes can be seen as constructing a network or block chain structure as components, and a specific model must be defined to define the block chain structure.

Definition 2. ***Message delivery model(Message passing model)***
We study a distributed system consisting of a set of nodes $\mathcal{N} = \{n_0, n_1, ..., n_k\}$ and assume the following condition.

- *Each node can process local computation*
- *Each node can send messages to all other nodes*

We want to set the minimum number of nodes in the distributed system to two. Specifically, a 'server' node that manipulates data (save, update, etc.) and a 'client' node are the minimum setting. Also, in order to clarify the configuration between the 'server' node and the 'client' node, we plan to develop the contents based on the following algorithm.

Algorithm 4 Naive client-server algorithm

```
The client sends one command at a time to the server.
```

Definition 3. ***Message loss model***
The message delivery model in which a message may be lost is called ***message loss model*** *and the following situation can be considered.*

- *No particular message can be guaranteed to reach its receiver safely.*
- *A message is received through message corruption, but there is a possibility that the content may be modulated.*

Algorithm 5 Client-server algorithm with receipt confirmation

```
The client sends one command at a time to the server.
The server acknowledges each command.
If the client does not receive the receipt confirmation within a certain time, the command
    is sent again.
```

Sending commands one at a time means that when a client sends a command c, it does not send a new command $\widetilde{c}$ until it receives an receipt confirmation, and the client node sends each command to all server nodes. If the command is sent and all receipt confirmations are received from each server node, the command is considered to have been executed successfully.
Now, to explain the consensus algorithm, let's assume that all existing nodes can send messages to all other nodes, and that the transmitted messages are always received because there is a reliable link.

Definition 4. ***Consensus***
Assuming that a maximum of f collisions can occur with a total of m nodes, $m - f$ nodes are called 'correct', and n_i starts with the input value v_i. These nodes must then choose one of those values, and must satisfy the following requirements:

- *Agreement : All correct nodes choose the same value.*
- *Termination : The correct nodes terminate in a finite amount of time.*
- *Validity : The determined value must be the input value of a certain node.*

All blockchains contain a consensus algorithm, which defines how to reach consensus on block creation and transaction processing in a distributed network. In addition, in the blockchain, consensus algorithm is used to maintain the distributed layer state that network participants agree on, to make important decisions, and to verify the validity of transactions. Commonly used consensus algorithms include:

- Proof of Work, PoW: Many blockchains, such as Bitcoin and Ethereum, use proof-of-work algorithms, which give miners the right to create blocks by solving mathematical problems and meeting certain conditions.
- Proof of Stake, PoS: Some blockchains use a proof-of-stake algorithm, which grants block creation authority based on the amount and time of cryptocurrency held by network participants.
- Hybrid PoW/PoS: It is known that some blockchains are using hybrid algorithms combining proof-of-work and proof-of-stake, and Ethereum, for example, will switch to Ethereum 2.0 and use a consensus algorithm that combines proof-of-work and proof-of-stake.

In addition to the three above, there are various consensus algorithms, including Delegated Proof of Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT), and Permissioned Consensus. In fact, all blockchains use consensus algorithms to define how network participants reach consensus and maintain the state of the blockchain, and each blockchain has specific characteristics and security levels depending on the consensus algorithm used. Considering this, it is of great significance to include the consensus algorithm as a functional element of the blockchain. Therefore, in this study, the set of consensus algorithms is denoted as Con, and the selection of various consensus algorithms is expressed as a component of the set.

Definition 5. ***Network partition***
Network partition is defined as a failure in which a network is divided into at least two and cannot communicate with each other.

There are various factors that cause division, and representative examples are as follows.

- Physical disconnection
- Software error
- Incompatible protocol

Definition 6. ***Consistency***
All nodes in the system agree on the current state of the system.

Definition 7. ***Availability***
The system is operational and can process incoming requests immediately.

Definition 8. ***Partition tolerance***
It is the ability of a distributed system to continue to function correctly in the face of a network partition.

In general, if no new update occurs in the shared state between nodes, the system may ultimately be viewed as a resting state. In other words, it is said that the nodes are in a state in which messages no longer need to be exchanged, and the shared state becomes consistent. This state is called **'eventual consistency'**. Eventual consistency is a form of consistency and is considered a form of weak consistency. Also, eventual consistency ensures that the state is eventually agreed upon, but temporarily nodes may disagree.

Theorem 1. ***CAP theorem***
It is impossible for a distributed system to provide consistency, availability, and fragmentation tolerance all at the same time, and it can meet two of these attributes but not all three.

The concept of blockchain to be dealt with in this study is intended to be established based on a network with eventual consistency, and the various definitions of the content that will be developed later are also intended to be set as the same premise.

Definition 9. ***Transaction***
A transaction is a unit that transmits or records data in a block chain, and a block chain collects these transactions into a structure called a block and connects them to form an overall transaction record. In this study, the ith transaction is denoted as tx_i, and the set of transactions is denoted as Tx. However, note that Tx is a set of transactions and should be distinguished from blocks.

Now, we want to lay the foundation for the axiom of blockchain by defining a block through these transactions.

Definition 10. ***Block***
A block is a data structure for transmitting gradual changes to the local state of a node. A block consists of a list of transactions, references to previous blocks, and nonces. Specifically, a block consists of several components as follows:

- *Block Header: A block header is a part that contains the metadata of a block. Typically, this part contains information such as the version of the block, hash of the previous block, timestamp, and difficulty. The block header serves to determine the identity of the block and its location within the blockchain.*
- *Transaction Data: Transaction data is the actual transaction information included in the block, and this data consists of a list of transactions included in the block, and each transaction includes information such as sender, receiver, transferred asset or data.*
- *Block Hash: A block hash is a unique identifier of a block and is created by hashing the block header and all contents of the transaction data. The block hash is then used to verify the connection and integrity of blocks.*
- *Previous Block Hash: The hash of the previous block represents the connection to the previous block. Through this, the blockchain maintains continuity and blocks are connected sequentially.*
- *Nonce: The nonce is a value used in the Proof of Work algorithm, which is repeatedly changed to meet certain conditions to calculate the hash of a block.*

A tree of blocks can be constructed through references to previous blocks, and the root of this tree is called **genesis block**.

3 Mathematical structure of block chain using isomorphic relationship in network structure

3.1 Mathematical Approach to Blockchain and Axiom

Blockchain is a distributed database technology that distributes and stores transaction records among multiple computer nodes and manages them. Each transaction is tied into a data structure called a block, and these blocks are linked in chronological order to form a chain. This decentralized structure enables reliable transaction records to be maintained and shared without a central administrator. In fact, we plan to proceed with axiomatization to mathematically approach the block chain, and it was determined that the technical definition has limitations as a procedure for this. In order to solve this problem, we will look at the comprehensive characteristics of blockchain and give a mathematical definition through it. The common characteristics of universal blockchain are as follows.

- Distributed Ledger: Blockchain stores transaction information distributed across multiple computer nodes. Because of this, unlike a centralized database, there is no single point of attack, and it is difficult to forge or change data.
- Transparency: In the blockchain, all participants have access to transaction information, and anyone can verify transaction records. This increases reliability and transparency.

- Security: Blockchain uses encryption technology to protect the contents of transactions. This enhances data safety and security.
- Decentralization: A blockchain is a decentralized network that can operate without a central administrator. Thus, transactions can proceed without the involvement of intermediaries or central authorities.

3.2 Axioms and abstractions about blockchain

Now, in order to construct the axioms of blockchain and define a general blockchain through this, we will recall the functional elements of blockchain. Assuming that k is the number of nodes and t is the number of blocks, Let the set of nodes $\mathcal{N} = \{n_0, n_1, ..., n_k\}$, the set of transactions $Tx = \{tx_1, tx_2, ...\}$, the set of consensus algorithms $Con = \{c_0, c_1, ...\}$, the set of blocks $LetmathcalB = \{B_0, B_1, ..., B_t\}$.

Definition 11. ***Blockchain***
For a suitable $c_j \in Con$, the network relation $\mathcal{BC}(\mathcal{N}, c_j, Tx, \mathcal{B})$ that satisfies the following condition is called ***Blockchain****.*

(a) (Distributed Ledger) Distributed and stored transactions across multiple computer nodes.

(b) (Transparency, Security) Transactions are encrypted and stored and shared on all Tx.

(c) (Decentralization) It is a distributed network structure that operates without a central administrator.

Through the concept of a block chain that satisfies these axioms, we intend to proceed with a mathematical approach, and through this, we intend to create a more general block chain structure.

3.3 Definition of Sub Blockchain

The final goal of this study is to establish a concept for handling various data at once, and for this, Sub Blockchain is defined as follows.

Definition 12. ***Sub Blockchain***
For a subset $\mathcal{N}'$ of a set of nodes $\mathcal{N} = \{n_0, n_1, ..., n_k\}$, if the relationship $\mathcal{BC}(\mathcal{N}', c_j, \widetilde{Tx}, \widetilde{\mathcal{B}})$ again forms a blockchain structure, let's define it as ***Sub Blockchain*** *of $\mathcal{BC}(\mathcal{N}, c_j, Tx, \mathcal{B})$.*

Additionally, two blockchain structures whose network structure is graphically isomorphic and have the same consensus algorithm are called **isomorphic** and are denoted as follows.

$$\mathcal{BC}(\mathcal{N}, c_j, Tx, \mathcal{B}) \cong \mathcal{BC}(\mathcal{N}', c_j, Tx', \mathcal{B}') \tag{16}$$

4 Text data compression method on network through encryption method

4.1 Encoding

Converts text data to be compressed and encrypted into ASCII code. The ASCII code used at this time corresponds to the 94th decimal, and the numbers corresponding to the number of digits of the work, such as 0, 1, $\ldots$, and 9, are expressed as 00, 01, $\ldots$, and 09. Let the number of bits of the original text data be $\widetilde{m}$. Then, the number of bits of the converted ASCII code becomes $2\widetilde{m}$, and let's express the listed ASCII code values as one number m.
ASCII code value : When $a_1, a_2, \ldots, a_{\widetilde{m}}$,

$$m := a_1 \oplus a_2 \oplus \cdots \oplus a_{\widetilde{m}} = a_1 a_2 \ldots a_{\widetilde{m}} \tag{17}$$

As a notation for this situation, let's do the following:

$$\operatorname{repn}(\mathit{text\ data}) = m \tag{18}$$

Through this process, text data was encoded into numerical values in the form of integers, and through this conversion process, text data will be encrypted with points on the elliptical curve.

4.2 Encryption Process

If you select the elliptic curve $\mathbb{E}(\mathbb{F}_p)$ on the finite field $\mathbb{F}_p$ having natural number n greater than m as order, there is an isomorphism $\sigma : \mathbb{E}(\mathbb{F}_p) \longrightarrow \mathbb{Z}_n$ that satisfies the following condition, and this function is called the EI function. Let's denote the inverse of this function as ρ.

$$\rho := \sigma^{-1} : \mathbb{Z}_n \longrightarrow \mathbb{E}(\mathbb{F}_p) \text{ such that } \rho(1) = P \tag{19}$$

In this case, P selects one of the generators of the elliptic curve $\mathbb{E}(\mathbb{F}_p)$. Then a suitable integer k exists and satisfies the following.

$$\rho(m) = kP \tag{20}$$

At this time, encryption is performed using the points P and n on the elliptic curve as the secret key. Through this method, the encoded text data was encrypted as a point on an elliptic curve. This encryption is going to be carried out as an encryption process with a secret key and a symmetric key structure, and the best performance is achieved by transmitting the minimum amount of information on the network. We will try to prove this theoretically and experimentally later.

4.3 Compressed data and transmission methods

Let's define the function $\phi : \mathbb{E}(\mathbb{F}_p) \longrightarrow \mathbb{Z}$ as follows.

$$\phi(kP) = k \tag{21}$$

Let's call the **data compression process** a combination of the encryption process and the process of matching the element k of the integer space $\mathbb{Z}$ through the function ϕ. The k value obtained through encryption is transmitted over the network. Even if it is leaked through an attack on the network, the information on the original data cannot be leaked because the structure of the elliptic curve $\mathbb{E}(\mathbb{F}_p)$ and the information of the point P on the elliptic curve $\mathbb{E}(\mathbb{F}_p)$

cannot be known. At this time, the information about the point P must be shared in advance to proceed with the decryption process. To this end, we plan to construct a network method through encryption through a classical method.

4.4 Decryption Process

After receiving the k value transmitted on the network, kP is calculated by calculating the private key P k times, and through this, you can find the encrypted data. For a functional expression, the decryption process is expressed through the inverse function ϕ^{-1} of ϕ as follows.

$$\phi^{-1}(k) = kP \tag{22}$$

Through this decoding process, it is converted into a point on the elliptic curve, and the point on the elliptic curve is converted into the original data form through a decoding process. In general, encryption using elliptic curves takes the form of public key structure encryption, but in this case, a special type of encryption method including compression and data transmission process is used, so it is different from the general situation and uses a symmetric key structure. Be aware. /so it should be recognized that symmetric key structures are used differently from general situations.

4.5 Decoding Process

If the number of digits of m obtained through the decryption process is odd, 0 is added to the front digit, and if it is even, it is left as it is, divided into two digits, and converted into text corresponding to each ASCII value.

$$\rho^{-1}(kP) = \sigma(kP) = m \tag{23}$$

Through this process, original text data can be obtained without any loss.

$$\begin{tikzcd}
 & \text{Data compression} & \text{Network} \\
\mathbb{Z}_n \arrow[r, "\rho: = \sigma^{-1}"] & \mathbb{E}(\mathbb{F}_p) \arrow[r, "\phi"] & \mathbb{Z} \\
\rotatebox{90}{$\in$} & \rotatebox{90}{$\in$} & \rotatebox{90}{$\in$} \\
m \arrow[r, mapsto] & \rho(m) = k\mathrm{P} \arrow[r, mapsto] & k \arrow[ddd, "\text{Send}"] \\
 & & \\
 & & \\
\mathbb{Z}_n & \mathbb{E}(\mathbb{F}_p) \arrow[l, "\rho^{-1} = \sigma"'] & \mathbb{Z} \arrow[l, "\phi^{-1}"'] \\
\rotatebox{90}{$\in$} & \rotatebox{90}{$\in$} & \rotatebox{90}{$\in$} \\
m = \rho^{-1}(k\mathrm{P}) & k\mathrm{P} \arrow[l, mapsto] & k \arrow[l, mapsto]
\end{tikzcd}$$

Figure 2: Schematic diagram of data compression process

The figure below (Figure 3) is a schematic diagram of the entire data compression process. Let's define the data compression process as a process corresponding to k on an integer through an encoding process and an encryption process. That is, expressed through a function:

$$\phi \circ \rho \circ \mathrm{repn}(text\ data) = k \tag{24}$$

We integrally define this situation as **data compression via encryption process**. After that, we will prove the security strength of data encryption and performance indicators such as the compression rate for data compression and the loss rate through data compression. Data compression was performed through the ECHC encryption process, and based on this fact, the security strength of data compression can be measured using the security strength of the ECHC cryptographic scheme. So, the index of security strength has a similar indicator to the security strength of the encryption process. Now let's develop a theory to measure an indicator of security strength.

4.6 Performance evaluation

4.6.1 Security level

The ECHC encryption scheme proceeds with encryption on an elliptic curve, and in order to use this fact, the partial exponential algorithm must first be used. According to the partial exponential algorithm, the running time of the algorithm is $L[x, c, \alpha]$ and $L[x, c, \alpha]$ is defined as follows.

$$L[x, c, \alpha] = O\left(exp\left((c + o(1))(lnx)^{\alpha}(lnlnx)^{1-\alpha}\right)\right), \tag{25}$$

At this time, x is the size of the input space, c is a constant, and $0 < \alpha < 1$.
To prove the running time of the encryption process on the elliptic curve, the running time of the optimal algorithm for $\mathbb{F}_p$ is $L[p^m, c, 1/2]$ is well known and we want to take advantage of this fact.
First, $L[p^m, c, 1/2]$ is expressed as follows.

$$L[p^m, c, 1/2] = O\left(exp\left((c + o(1))(lnp^m)^{1/2}(lnlnp^m)^{1/2}\right)\right) \tag{26}$$

Also, if $q = p^m$ and $\mathbb{E}(\mathbb{F}_q)$ is a supersingular curve, the reduction of the elliptic curve logarithmic problem in the elliptic curve $\mathbb{E}(\mathbb{F}_q)$ for the discrete logarithmic problem in the finite field $\mathbb{F}_{q^k}$ is a stochastic polynomial time (in lnq) contraction. So we can get the following result.

On the supersingular elliptic curve $\mathbb{E}(\mathbb{F}_q)$, let the points P and $R = lP$ of order n be the points on $\mathbb{E}(\mathbb{F}_q)$. If q is in the form of a prime number or a power $q = p^m$ of a prime number p, l can be determined by probabilistic subexponential time. Considering this situation, the time complexity of the encryption process on the elliptic curve is $L[p^m, c, 1/2]$. The number of steps that are calculated before stopping the algorithm is as follows.

$$\leq \frac{n_1}{\phi(n_1)} \frac{n_2}{\phi(n_2)} = O((lnlnN)^2) = O((lnlnq)^2). \tag{27}$$

4.6.2 Compressibility

The data compression method developed by the author reduced the text data to the $Q = \rho(m) = kP$ point of the elliptic curve by including the location information, and compressed the information of this point into the integer information k. At this time, since the network type is configured with the secret key structure, information about the point P does not need to be included in the transmission process. So, the process of data compression is defined as follows.

$$m \longrightarrow Q = \rho(m) = kP \longrightarrow k \tag{28}$$

Now, to calculate the general compression ratio p_c, if the number of digits of k is l, it is as follows.

$$p_c = \frac{l}{\widetilde{m}} \tag{29}$$

Using the structure of the group, by selecting the point P, you can select k with the number of digits l exceeding $\frac{n}{4}$, and the following holds.

$$n \geq 2\widetilde{m} \implies \frac{n}{2} \geq \widetilde{m} \tag{30}$$

Therefore, it can be seen that the following is established.

$$\frac{1}{\widetilde{m}} \geq \frac{2}{n} \implies \frac{l}{\widetilde{m}} \geq \frac{2l}{n} \tag{31}$$

Now we can consider the compression ratio as:

$$p_c = \frac{l}{\widetilde{m}} \geq \frac{2l}{n} \geq \frac{2}{n} \times \frac{n}{4} = \frac{1}{2} \tag{32}$$

As a result, the compression ratio p_c is more than 50%.

4.6.3 Loss rate

The content of this text is a state in which a data compression method has been developed through a method different from a general compression method, that is, a method of compressing in the form of deleting location information or some data. The data compression method developed by the author described above reduced the text data to the $Q = \rho(m) = kP$ point of the elliptic curve, including the location information, and compressed the information at this point into the integer type information k. Now, the original data can be transmitted without missing information through a decryption process using the private key structure. In conclusion, it has a structure that does not cause data loss during data compression. As a result, the loss rate is zero.

5 Financial transaction methodology using elliptic curve group structure

5.1 Trading methodology using isomorphic structure

In this study, the process of remittance in financial transactions was set to a limited scope as a standard, and the entire process of financial transactions using account encryption was shortened and named Transaction, which will be used throughout the text.

Definition 13. ***Transaction Requester***
The transaction requester (receiver) is a party requesting a transfer from the other party to the transaction in order to receive the remittance, and the transaction requester will be denoted by R.

In order to set the subject of the transaction, the transaction requester was defined in advance, and the counterparty of the transaction requester is set to set all the subject of the transaction in detail.

Definition 14. ***Transaction Receiver, Sender***
Transaction receiver and sender are parties that make remittances, and in this study, they will be called transaction receiver or sender for convenience, and will be denoted as P.

The transaction in this study is a network-based system, and an engine for encryption operation and user management is required to build a complete system.

Definition 15. ***Management Engine***
The integrated management engine (Management Engine) aims to ensure that the entire transaction proceeds without problems, and is actually responsible for managing other elements so that the network-based system can operate smoothly. In this text, we will call it engine for short and denote it as M. The specific functions of the engine are as follows.

- *Cryptographic operation*
- *Crypto information sharing*
- *Cryptographic system initialization*
- *Managing data transfer between R and P*

As suggested earlier, the transaction system is network-based, and the institutions that directly conduct transactions are the Open Banking Center and banks. Therefore, I would like to specifically define the two institutions.

Definition 16. ***Open Banking Center***
The Open Banking Center is an element that connects the bank and the transaction receiver in order for the transaction receiver to proceed with the remittance, and serves to request the bank to transfer the remittance request received from the transaction receiver. The Open Banking Center will be denoted with an O.

Finally, we want to define a bank that directly performs wire transfers. Here, account transfer will be shortened to transfer.

Definition 17. ***Bank***
The bank serves to perform the transfer requested from the open banking center (O) to the transaction requester (R), and will be denoted as B.

From now on, we will explain the overall process of the system by presenting a specific algorithm based on the elements defined above. The encryption process on the elliptic curve $\mathbb{E}$ is expressed as $\widetilde{C} = \widetilde{Enc}(m)$ using the notation of the elliptic curve-based homomorphic encryption, which is the basis of this study, and if σ is the EI function between $\mathbb{E}$ and $\mathcal{R}$, the encryption process on $\mathcal{R}$ is expressed as $C = \sigma(\widetilde{C})$ and shortened as $C = Enc(m)$. The algorithm to be introduced next intends to proceed with the premise that M transmits parameter information based on ECHC encryption to R and P. In addition, an additional security environment was established by inserting a transaction permission certification procedure similar to digital signature in the middle of the algorithm.

Algorithm 6 Encryption Remittance Process Algorithm Using Homomorphic Encryption

```
Input : Account number $m$
Output : Account transfer completed
R calculates $\widetilde{C} = \widetilde{Enc}(m)$
R transmits $\widetilde{C} = \widetilde{Enc}(m)$ to M
M transmits $C = Enc(m)$ to P
P calculates $m = Dec(C)$
Determining that P is $C = \sigma(\widetilde{C})$, accept the transaction, otherwise reject the transaction
When the transaction is accepted, P transfers $m$ to O
O requests transfer to B
B transfers to R
```

Figure 4 shows the schematic representation of Algorithm 6 introduced above for better understanding. It was written in terms of functionality without including the meaning of short-term financial transactions that the company wants to develop. We want to create a system that has the meaning of short-term financial transactions by adding a 'function to delete data after a certain period of time'. The goal of the 'deleting data after a certain period of time' function is to improve various problems that may be caused by maintaining the state of holding the account data that has been traded. Problems in the situation of having account data are as follows.

- Data leak through P's management server attack
- Possibility of data leakage if P is a malicious user
- Possibility of leakage through new attack methods

In order to improve these various security problems, it was found to be meaningful for the function of deleting account data after a certain period of time, and here, the meaning of a certain period is to be given in terms of both time and algorithm stage. In terms of time, the standard of 5 minutes was set, and in terms of the algorithm stage, the time of transfer completion was set as the standard, and various figures are planned to be applied. The meaning of specific figures will be demonstrated in the system implementation stage, and will be specified through related papers.

Now, we would like to introduce a new algorithm that adds the meaning of one-off. In this study, we aim to build a system based on the two algorithms. In particular, the system to be developed based on the current research is intended to have high significance by setting a high importance stake for the algorithm to be introduced next.

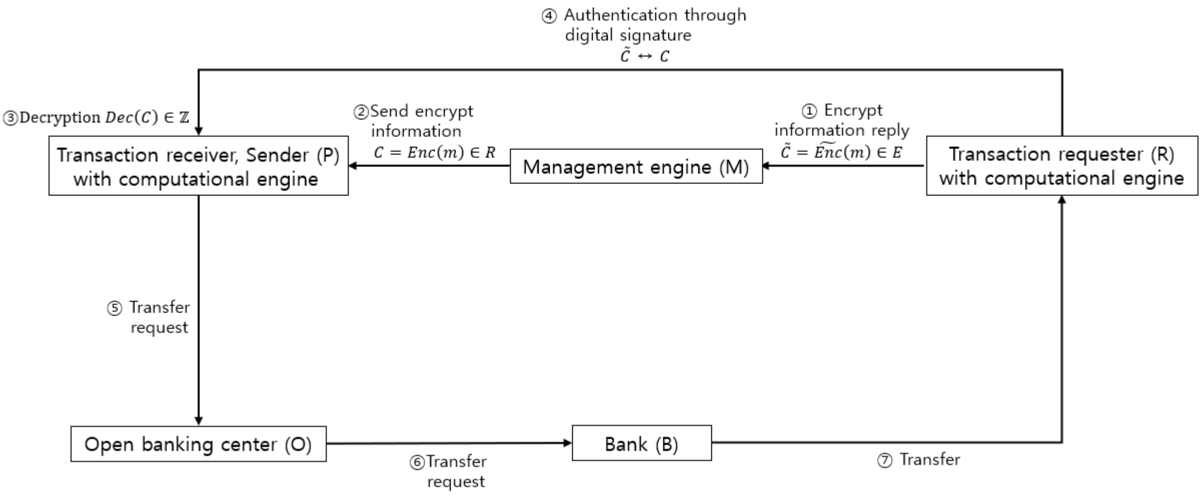

④ Authentication through digital signature $\tilde{C} \leftrightarrow C$
③Decryption $Dec(C) \in \mathbb{Z}$
②Send encrypt information $C = Enc(m) \in R$
① Encrypt information reply $\tilde{C} = \widetilde{Enc}(m) \in E$
Transaction receiver, Sender (P) with computational engine
Management engine (M)
Transaction requester (R) with computational engine
⑤ Transfer request
Open banking center (O)
⑥Transfer request
Bank (B)
⑦ Transfer

Figure 3: Network-based financial transaction system using elliptic curve-based homomorphic encryption

Algorithm 7 Algorithm for remittance process using homomorphic encryption

```
Input : Account number $m$
Output : Account transfer completed
R calculates $\widetilde{C} = \widetilde{Enc}(m)$, Measurement of transaction start time (if transaction is not completed after n minutes, transaction is terminated and account number data is deleted)
R transmits $\widetilde{C} = \widetilde{Enc}(m)$ to M
M transmits $C = Enc(m)$ to P
P calculates $m = Dec(C)$
Determining that P is $C = \sigma(\widetilde{C})$, accept the transaction, otherwise reject the transaction
When the transaction is accepted, P transfers $m$ to O
O requests transfer to B
B transfers to R, deletes account number data
```

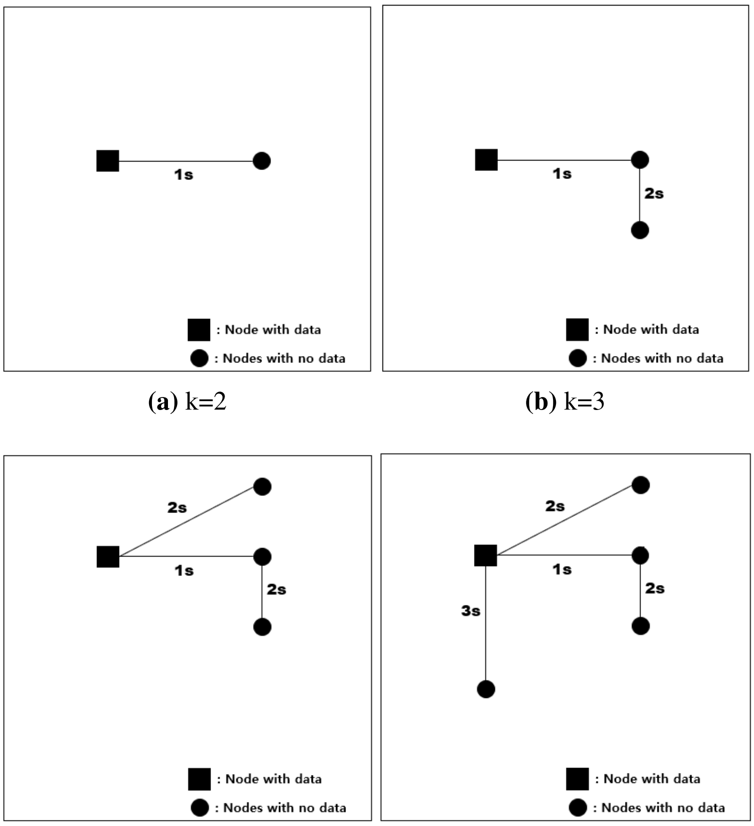
6 Block Broadcast Methodology on Blockchain Network

6.1 Algorithm for data transmission between nodes using a graph

In the initial state, we tried to find the information to be the basis through observation, and observed while increasing the number of nodes k discretely. In all graphs, square boxes are set as nodes with data, and circles are set as nodes without data. Additionally, in all graph types, the number of nodes that hold data is set to one. Due to the characteristics of the blockchain structure, the minimum number of nodes is $k = 2$ and the number of steps is 1, so the total time required is $1S$. These situations are broken down by case and summarized as follows. This situation is divided by case and summarized as follows.

- Case 1 (2 nodes) The minimum number of steps is 1 and the total time taken is $1S$.
- Case 2 (3 nodes) The minimum number of steps is 2 and the total time taken is $2S$.
- Case 3 (4 nodes) The minimum number of steps is 2 and the total time taken is $2S$.

- Case 4 (5 nodes) The minimum number of steps is 3 and the total time taken is $3S$.



(a) k=2

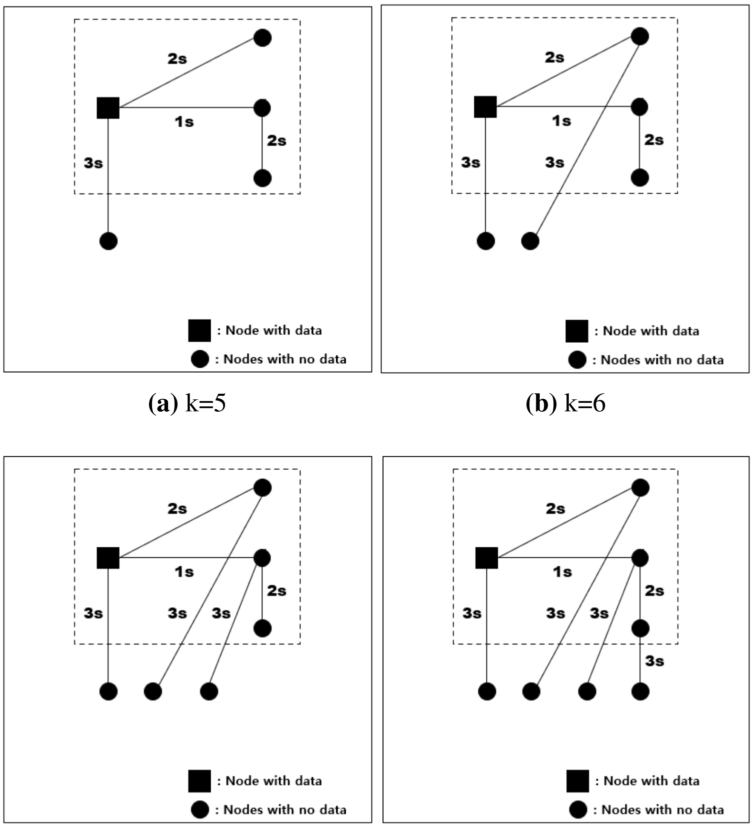
(b) k=3

(c) k=4

(d) k=5

Figure 4: The number of shortest steps for each number($k = 2, 3, 4, 5$)

Based on the observed data, it was determined that it was important to create an environment in which more efficient calculations could be performed by grouping, and it was reasonable to set the number of nodes to be grouped to 4. Additional observations were made to prove this.



(a) k=5

(b) k=6

(c) k=7

(d) k=8

Figure 5: The number of shortest steps for each number($k = 5, 6, 7, 8$)

- Case 1 (5 nodes) The minimum number of steps is 3 and the total time taken is $3S$.
- Case 2 (6 nodes) The minimum number of steps is 3 and the total time taken is $3S$.
- Case 3 (7 nodes) The minimum number of steps is 3 and the total time taken is $3S$.
- Case 4 (8 nodes) The minimum number of steps is 3 and the total time taken is $3S$.

Specifically, based on grouped nodes, the minimum number of steps is 2, and since there are a total of 4 grouped nodes, data can be transmitted to up to 4 people. Considering this, it can be determined that the total time taken for up to 8 nodes is $3S$, and when nodes are added, the total time taken is also added. When presenting a specific algorithm, let's consider the algorithm in the form of treating the first 4 grouped nodes as the total time taken of $2S$, and adding 1 step by matching the additional $1\sim$ 4 nodes one by one. Applying this situation inductively, we obtain the following theorem:

Theorem 2. ***Grouping-related theorems***
If you group 1 node that has data and 3 nodes that do not, you can calculate the shortest total time taken.

6.2 Calculation of the shortest time of the data transmission algorithm

By generalizing and calculating the above situation through the grouping-related theorem, the following result can be obtained.

index	1	2	3	$\cdots$	k
step	3	4	5	$\cdots$	$k+2$
the maximum number of processable nodes	8	16	24	$\cdots$	$2 \times (4 \times k)$
number of groups	2	4	6	$\cdots$	$2k$

Table 1: The maximum number of nodes and the number of groups that can be processed at the kth

As a result, in the kth case, the number of steps is $k + 2$, and the maximum number of nodes that can be processed is $8k$ with the $(k-1)$th case fixed.

7 Blockchain Consensus Algorithm Using Elliptic Curve Discrete Logarithmic Problem (ECDLP)

7.1 Consensus Algorithm Using Elliptic Curve Discrete Logarithm Problem

Assume that the set of elliptic curves to define the elliptic curve discrete logarithmic problem is denoted by $\mathcal{E} = \{\mathbb{E}_1, \mathbb{E}_2, ...\}$, and each elliptic curve $\mathbb{E}_i$ $(i = 1, 2, ...)$ contains information about the elliptic curve, such as the Weierstrass equation and order. All elliptic curves are defined on the finite field $\mathbb{Z}_p$ for a prime number p, and at this time, the decimal p is set to a fixed value. In addition, we will write the set of generators of the elliptic curve $\mathbb{E}_i$ as $\mathcal{P}_{\mathbb{E}_i}$, and an arbitrary elliptic point on the elliptic curve $\mathbb{E}_i$ as $Q = (x, y)$.
In addition, the concept of blockchain is essential to define the consensus algorithm, and we would like to list definitions of mathematical materials related to blockchain. Assuming that k is the number of nodes and t is the number of blocks, let the set of nodes $\mathcal{N} = \{n_0, n_1, ..., n_k\}$, the set of transactions $Tx = \{tx_1, tx_2, ...\}$, the set of consensus algorithms $Con = \{c_0, c_1, ...\}$, and the set of blocks $\mathcal{B} = \{B_0, B_1, ..., B_t\}$.

Definition 18. ***Blockchain***
For some $c_j \in Con$, a network relation $\mathcal{BC}(\mathcal{N}, c_j, Tx, \mathcal{B})$ that satisfies the following condition is called ***Blockchain****.*

(a) (Distributed Ledger) Distributed and stored transactions across multiple computer nodes.

(b) (Transparency, security) Transactions are encrypted and stored and shared in all Tx.

(c) (Decentralization) A distributed network structure that operates without a central administrator.

Based on these notations, **'Consensus Algorithm Using Elliptic Curve Discrete Logarithm Problem'** is defined as solving the problem of calculating the discrete logarithmic value m for an arbitrary elliptic point $Q = (x, y)$ based on the randomly selected generator P from the generator set $\mathcal{P}_{\mathbb{E}_i}$ of the elliptic curve $\mathbb{E}_i$ $(i = 1, 2, ...)$. In other words, it can be seen that the node that quickly solves the discrete logarithm problem on the randomly selected elliptic curve becomes the miner, and the consensus algorithm is defined. Solving the problem based on the randomly selected generation source P for each node presents a situation where various running times can be randomly obtained. Through this principle, it is possible to solve the criteria for computing power presented in the introduction by creating an environment in which a proportional relationship between computing power and problem solving is not established. In addition, it is possible to set the difficulty through the aspect that the time complexity of the operation can be adjusted according to the elliptic curve, and the criteria for the difficulty setting that this study wanted to consider can also be solved. From now on, We would like to present a specific process for the consensus algorithm using the elliptic curve discrete logarithmic problem defined above, Afterwards, the contents of this text will proceed on the assumption that the elliptic curves $\mathbb{E}_i,\ (i = 1, 2, ...)$ are randomly selected from the set of elliptic curves $\mathcal{E} = \{\mathbb{E}_1, \mathbb{E}_2, ...\}$.

Algorithm 8 Consensus Algorithm Using Elliptic Curve Discrete Logarithmic Problem (ECDLP)

```
Input: Set of nodes $\mathcal{N} = \{n_0, n_1, ..., n_k\}$, random elliptic point $Q$ of elliptic curve $\mathbb{E}_i$
Output: A specific node $n_i$ selected as a miner
Select $\mathbb{E}_i \in_R \mathcal{E}$ and broadcast it to the set of nodes $\mathcal{N}$
Select $P \in_R \mathcal{P}_{\mathbb{E}_i}$
Calculate $m$ that satisfies $Q = mP$
If the elliptic curve discrete logarithm value $m$ matches, select as miner
```

7.2 Transaction authentication procedure algorithm

In this study, the principle of the Elliptic Curve Digital Signature Algorithm (ECDSA) was used as an idea and included in the entire process of the consensus algorithm in order to additionally proceed with the authentication process that each node is participating in the transaction. In this regard, let's set the notation for the necessary mathematical elements the same as those defined in the previous consensus algorithm. Therefore, by using the Weierstrass equation that defines the elliptic curve $\mathbb{E}_i$, let's define the process of authenticating participation in a transaction by calculating the y-coordinate only with the information of the x-coordinate of the specific elliptic point $Q = (x, y)$ of the elliptic curve $\mathbb{E}_i$ **'transaction participation authentication algorithm'**. The following algorithm presents a specific process for the 'trade participation authentication algorithm', and proceeds on the premise of setting a specific elliptic point Q.

7.3 Elliptic Curve Discrete Logarithm Proof (ECDL Proof)

We intend to define the final consensus algorithm including transaction authentication by using the previously defined 'agreement algorithm using elliptic curve discrete logarithmic problem (ECDLP)' and 'transaction participation authentication algorithm'. Different from the order

Algorithm 9 Transaction authentication procedure algorithm

```
Input: Elliptic curve $\mathbb{E}_i$ $(i = 1, 2, ...)$, $x$-coordinate of a specific elliptic point $Q$ of elliptic curve $\mathbb{E}_i$
Output: True/False for transaction participation authentication
Calculation of the $y$-coordinate of a specific elliptic point $Q$ of the elliptic curve $\mathbb{E}_i$
If $(x, y) = Q$, evaluates to 'true'
If $(x, y) \neq Q$, evaluates to 'false'
True/false return for transaction participation authentication
```

introduced above, the algorithm for the transaction authentication procedure is conducted in advance to specifically find the coordinates of a specific elliptic point Q, and based on these coordinates, the discrete logarithmic problem of the elliptic curve is solved to find an integer m that satisfies the following.

$$Q = mP \tag{33}$$

In this way, the node that solves the elliptic curve discrete logarithmic problem the fastest is selected as a miner, and the genesis block is built up. This research team defined this comprehensive consensus algorithm as **'Elliptic Curve Discrete Logarithm Proof (ECDL Proof))'**. In the following Algorithm 10, we would like to introduce the comprehensive process of ECDL proof in detail.

Algorithm 10 Elliptic Curve Discrete Logarithm Proof (ECDL Proof))

```
Input: The set of nodes $\mathcal{N} = \{n_0, n_1, ..., n_k\}$
Output: A specific node $n_i$ selected as a miner
Select $\mathbb{E}_i \in_R \mathcal{E}$
Select $Q \in_R \mathbb{E}_i$
Broadcast randomly selected $\mathbb{E}_i, Q$ to the set of nodes $\mathcal{N}$
Calculate the $y$-coordinate of the elliptic point $Q$ of the elliptic curve $\mathbb{E}_i$
If $(x, y) = Q$, evaluates to 'true'
If $(x, y) \neq Q$, evaluates to 'false'
True/false return for transaction participation authentication
If the decision on transaction participation authentication is 'true', select $P \in_R \mathcal{P}_{\mathbb{E}_i}$
Calculate $m$ that satisfies $Q = mP$
Determining the match for the elliptic curve discrete log value $m$
If the value of $m$ matches, it is determined that the problem has been solved and selected as a miner.
```

8 Encryption-based personal identification system using isomorphic structure and block writing methodology

8.1 Individual identification using isomorphic structure of elliptic curve

In order to present a methodology for individual identification by utilizing the isomorphic structure of elliptic curves, the area is divided into layers and the theory is developed. First, let's classify it into three layers, the Real Layer, the Serial Number Layer, and the Elliptic Curve Point Layer. Here, the elliptic curve point layer is a layer that we want to share publicly, so we will also call it **Data Sharing Layer**. If there is no additional situation, let's judge it as a data sharing layer.

Here, the real layer is a layer composed of addresses for each individual user. Since we have a research goal to present a methodology that does not actually share addresses, we want to conduct research in a situation where the data of the actual floor is not disclosed. The serial number layer is a layer in which serial numbers composed of large-sized integers that do not contain address information are matched and stored for each individual user. Clearly, the serial number is not an element defined on the integer set $\mathbb{Z}$, but an element defined on the class $\mathbb{Z}_n$ classified as a congruent expression $\equiv_n$Here, n is an integer that satisfies the group isomorphism of the following elliptic curve $\mathbb{E}$.. Here, n is an integer that satisfies the group isomorphism of the following elliptic curve $\mathbb{E}$.

$$\mathbb{E} \cong \mathbb{Z}_n \tag{34}$$

It is a mathematically well-known fact that this relationship is formed, and it can be proved using Weil's pairing. A practical relationship can be implemented through the EI function σ presented in basic knowledge, and through this, the elliptic point P_i corresponding to the serial number SN_i can be directly matched.

$$SN_i = \sigma(P_i) \tag{35}$$

Here, the information of the sender and the receiver of the transaction is written based on the ellipse point, and the EI function σ serves as a kind of secret key. When it is necessary to know the actual transaction details, the transaction details written based on the elliptic points are matched with the serial number data through the EI function to convert the transaction details, and finally, the actual transaction details can be known through the data of the actual layer.

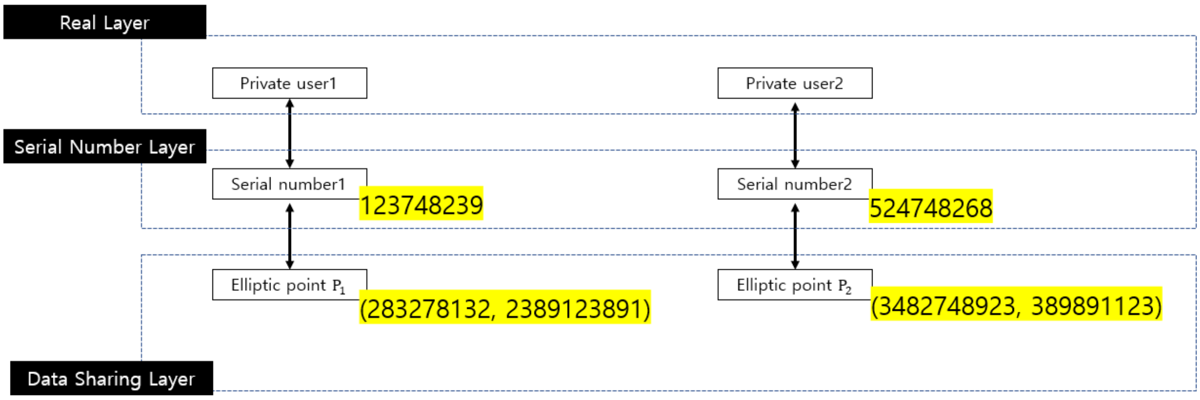


Figure 6: Layer-by-layer area of the personal identification system

8.2 Implementation and encryption method for each type of data written in the block

The data written in the block will be different for each block chain, but the data to be handled in the crypto block chain can be classified as follows based on one transaction.

- Timestamp data
- Sender data
- Receiver data
- Distributed data information (similar to transaction volume)
- The x-axis coordinate of the elliptic point Q in the consensus algorithm

In the case of timestamp data, it was determined that the previously used hash encryption method was most appropriate in terms of diversity, In the case of sender data and receiver data, as described in **3.1. Individual identification using isomorphic structure of elliptic curve**, elliptic curve cryptosystem (ECC) was used to convert to elliptic points and encrypt them. Lastly, the distributed data information is to represent the information on the amount actually traded, and it is the data that encrypts the feature vector $T_i = (t_1, t_2, ..., t_m)(i = 1, 2, 3, ...)$ of the image data stored in the Crypto Blockchain, and is expressed as follows. Here, the encryption process is the process of applying the elliptic curve cipher Enc and using the EI function σ as follows.

$$\widetilde{T_i} = \sigma \circ Enc(T_i) = \sigma(T_i P) = \sigma((t_1, t_2, ..., t_m)P) = \sigma((t_1 P, t_2 P, ..., t_m P)) \tag{36}$$

The above situation is summarized and re-presented as follows.

- **Timestamp data** : $Hash(timestamp)$
- **Sender data** : Elliptic point (x_1, y_1)
- **Receiver data** : Elliptic point (x_2, y_2)
- **Distributed data information** (similar to trading volume) : $\sigma \circ Enc(T_i)$
- **The x-axis coordinate of the ellipse point Q in the consensus algorithm**

To match the name of the crypto-blockchain, various encryption processes are applied to each type of data, and the meaning of this diversity can be translated into the meaning of providing a safe protection environment from attackers.

8.3 Comprehensive block writing methodology

Now, let's set a specific ellipse point Q to include the ECDL proof presented in the background knowledge, and let's say the following holds.

$$Q = (\widetilde{x}, \widetilde{y}) \tag{37}$$

Now, let's define the text data to be stacked in the block considering the following order.

$$Hash(timestamp) \oplus (x_1, y_1) \oplus (x_2, y_2) \oplus \sigma \circ Enc(T_i) \oplus \widetilde{x} \tag{38}$$

Now, let's say $TD = Hash(timestamp) \oplus (x_1, y_1) \oplus (x_2, y_2) \oplus \sigma \circ Enc(T_i) \oplus \widetilde{x}$ and introduce the procedure for building blocks on the network.

Algorithm 11 The process of building blocks on the network

```
New transaction broadcast to all nodes randomly selected $\mathbb{E}_i, Q$ to the set of nodes $\mathcal{N}$
Calculate the $y$-coordinate of the elliptic point $Q$ of the elliptic curve $\mathbb{E}_i$
If $(x, y) = Q$, evaluates to 'true'
If $(x, y) \neq Q$, evaluates to 'false'
True/false return for transaction participation authentication
If the decision on transaction participation authentication is 'true', select $P \in_R \mathcal{P}_{\mathbb{E}_i}$
Determining the match for the elliptic curve discrete log value $m$
If the $m$ values match, select a miner to determine that the problem has been resolved
Selected miners store $TD$ in genesis block
Selected miners broadcast blocks
The node approves and stores the block
```

9 Personal information wallet platform Strongbox

9.1 Data storage and control through crypto-blockchain

Crypto blockchain was researched and developed with the purpose of establishing an environment suitable for utilizing the relationship of a mathematical structure called isomorphic. Here, the meaning of isomorphic can be interpreted as a relationship that has the same mathematical structure even if it has a different form, and to utilize this meaning of isomorphic is as follows.

- To handle different types of data with the same meaning at once
- Block chain structure suitable for applying elliptic curve-based homomorphic encryption

For this research purpose, the crypto block chain was developed, and the functional elements of the block chain that are most important to this study are as follows.

- Establishment of sub-blockchain concept through axiomization of blockchain
- Research on the relationship (isomorphic relationship) of sub-blockchains
- Building an environment that can handle various types of data at once through the sub-blockchain concept

The role of crypto-blockchain in Strongbox is to handle various data at once. Specifically, by studying the isomorphic relationship of the sub-blockchains that make up the crypto-blockchain, sub-blockchains that form a homomorphic relationship are treated the same, and sub-blockchains that do not have a different role have been developed. The data types handled by the platform of this study can be organized as follows.

- **Type 1** Personal information
- **Type 2** electronic signature
- **Type 3** handwritten signature image data
- **Type 4** digital stamp image data

- **Type 5** fingerprint image data
- **Type 6** Smart contract transaction history data

Type 1 personal information is not data that constitutes a sub-blockchain, and data is not stored and shared in a locally encrypted state.Type 2 electronic signatures are used for personal identification and are not data that constitutes a sub-blockchain. Types 3, 4, and 5 are data forms that form an isomorphic structure, and are in a state where they do not form an isomorphic structure with type 6 smart contract transaction history data. Therefore, types 3, 4, and 5 are configured based on a private blockchain structure, and type 6 smart contract transaction history data is configured based on a public blockchain structure.

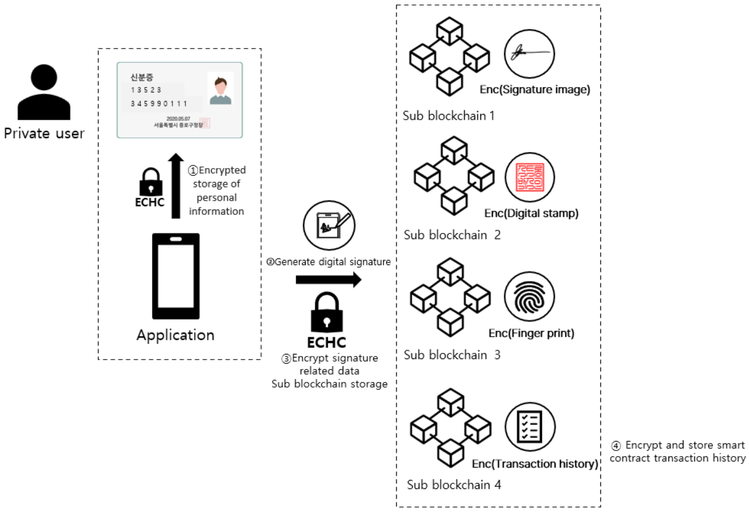


Figure 7: How to encrypt and store personal information and authentication data

9.2 Fingerprint authentication deep learning model through feature extraction

This study developed an algorithm for personal identification based on type 5 fingerprint image data, and considering the isomorphic structure, type 3 and 4 data can also be studied in the same way as this study. In order to proceed with fingerprint authentication, the fingerprint image is acquired from the device and matched with the fingerprint data stored in the sub-blockchain. Matching between these fingerprint data is difficult to proceed without decryption in the state of encrypted data using elliptic curve-based homomorphic encryption. To solve this problem, this study presents the following methodology. First, the process for storing fingerprint data in the sub-blockchain is as follows.

1. Deep learning learning based on secured fingerprint dataset
2. Fingerprint image feature extraction and vectorization
3. Encrypt the image vector and store it in the sub-blockchain

The process of an individual user's authentication through fingerprint recognition and the methodology of personal identification through a digital signature algorithm are as follows. Here, the digital signature algorithm was studied based on the EHDSA algorithm, and the EHDSA algorithm can perform matching for personal identification in a safe and secure environment by utilizing an asymmetric encryption form.

1. Acquire fingerprint image through device
2. Acquired fingerprint image feature extraction and vectorization
3. Retrieve data from sub-blockchain
4. Measure cosine similarity based on encrypted data
5. Identification of the same fingerprint image
6. Personal identification through stored electronic signatures

In this study, the reason for not proceeding based on the digital stamp image and the handwritten signature image is a situation that exists separately. In the case of a digital stamp image, it has a formal image form to measure the similarity of the image, so the accuracy aspect clearly has values of 0 and 1, and it was judged to be unsuitable for having the meaning of a research unit. On the other hand, in the case of the handwritten signature image, the accuracy aspect is relatively low and there is a part that is unsatisfactory to use as the main, so it is judged that it is reasonable to apply it as an auxiliary means.

9.3 Blockchain-based personal information inscribe algorithm

It is judged that it is misleading to express the function of the platform by sharing personal information, so rather than sharing personal information, We would like to express it as **'Individual users use the platform to enter personal information through authentication'**. The specific methodology proceeds through the following process.

1. Customers participate in the blockchain as a client node through membership registration
2. Customer creates smart contract
3. An individual user proceeds with fingerprint authentication through the device
4. Retrieve acquired fingerprint image data from sub-blockchain
5. Personal identification through digital signature algorithm
6. Request to write personal information to the Asset Node corresponding to the individual user
7. Inscribe by directly transmitting personal information in an encrypted state through a smart contract platform
8. Encrypt smart contract transaction details and store them in type 6 sub-blockchain

9. Send the written contract to the customer
10. Personal information used is deleted

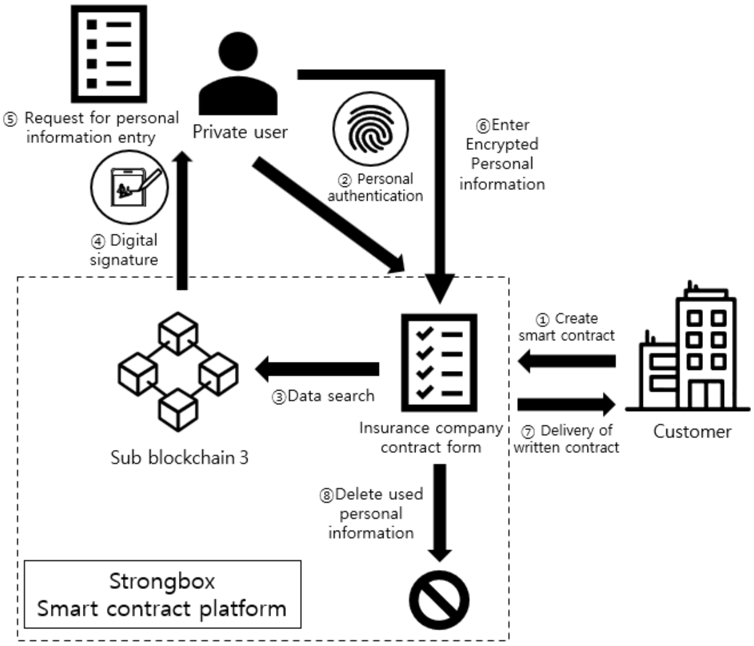

⑤ Request for personal information entry
Private user
② Personal authentication
⑥Enter Encrypted Personal information
④ Digital signature
① Create smart contract
③Data search
Insurance company contract form
⑦ Delivery of written contract
Customer
Sub blockchain 3
⑧Delete used personal information
Strongbox
Smart contract platform

Figure 8: Blockchain-based smart contract inscribe methodology

In order to achieve the final goal of this study, to develop a personal information sharing platform free from legal regulations related to personal information, it has the following characteristics.

- Stores sensitive personal information locally and holds only individual users
- Personal information is not stored and shared by the blockchain
- After receiving a request to fill in, personal information is also deleted after using the data on the platform.
- We do not provide personal information to our customers
- Possibility of data leakage can be blocked at the source because it is stored in an additionally encrypted state
- Processing without decryption process by utilizing the characteristics of homomorphic encryption in the state of entering personal information
- In situations where individual users fill in their personal information, personal information can be entered with simple authentication using fingerprints, digital stamps, signatures, etc.

9.4 Building an application using Flask

It was developed so that individual users who participate in the blockchain as Asset Nodes can use the application to proceed with the authentication process through fingerprint recognition. In order to build an environment for developing this application, the following functional elements were considered.

- Network environment to connect with blockchain network
- Environment for using elliptic curve-based homomorphic encryption
- Environment for applying EHDSA
- Deep learning environment for fingerprint recognition
- Web-based smart contract platform environment

Therefore, the development language that the company judged to be suitable for development was Python, and Flask was used to build a web and network environment based on Python. For application, which is the final result, the web using Flask was implemented as a model suitable for mobile using React-Native.

References

[1] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Commun. ACM*, vol. 21, no. 2, p. 120–126, feb 1978. [Online]. Available: https://doi.org/10.1145/359340.359342

[2] A. Menezes, T. Okamoto, and S. Vanstone, “Reducing elliptic curve logarithms to logarithms in a finite field,” *IEEE Transactions on Information Theory*, vol. 39, no. 5, pp. 1639–1646, 1993.

[3] J. P. Buhler and P. Stevenhagen, *Algorithmic number theory*. Cambridge University Press, 2008.

[4] K. Butt, “Elliptic curves and mordell-weil theorem,” *Ubicación: http://math. uchicago. edu/˜ may/REU2016/REUPapers/Butt. pdf*, 2016.

[5] W. Date and W. Note, “Archived nist technical series publication,” *NIST Special Publication*, vol. 800, p. 60, 1992.

[6] D. Johnson, A. Menezes, and S. Vanstone, “The elliptic curve digital signature algorithm (ecdsa),” *International journal of information security*, vol. 1, no. 1, pp. 36–63, 2001.

[7] H. W. Lenstra Jr, “Factoring integers with elliptic curves,” *Annals of mathematics*, pp. 649–673, 1987.

[8] A. K. Lenstra, “Key lengths,” Wiley, Tech. Rep., 2006.

[9] A. Menezes, “Elliptic curve public key cryptosystems,” in *The Kluwer international series in engineering and computer science*, 1997.

[10] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[11] L. D. Singh and K. M. Singh, “Implementation of text encryption using elliptic curve cryptography,” *Procedia Computer Science*, vol. 54, pp. 73–82, 2015.

[12] L. C. Washington, *Elliptic curves: number theory and cryptography*. CRC press, 2008.