ISSN: 2182-2069 / E-ISSN: 2182-2077

Decentralized Fair Data Trading Scheme based on mCL-ME Primitive

Su Jin Shin[1], Youngho Park[2], and Sang Uk Shin[3*]

[1]Department of Information Security, Graduate School, Pukyong National University, Busan, Republic of Korea. inuin9014@pukyong.ac.kr, https://orcid.org/0009-0009-7206-9684

[2]Electronics and Information Communications Research Center, Pukyong National University, Busan, Republic of Korea. pyhoya@pknu.ac.kr, https://orcid.org/0000-0002-4734-6274

[3*]Division of Computer Engineering and Artificial Intelligence, Pukyong National University, Busan, Republic of Korea. shinsu@pknu.ac.kr, https://orcid.org/0000-0002-7048-582X

Received: July 26, 2024; Revised: September 6, 2024; Accepted: October 4, 2024; Published: November 30, 2024

Abstract

The Internet of Things (IoT) is now used by people, homes, transportation, hospitals, and more, and the data collected from IoT devices is more diverse and vaster. Online data trading or sharing systems are evolving to use this data to create new value or utilize it for financial gain. Traditional online data trading systems rely on trusted third parties to ensure fair trading. However, these centralized trading systems suffer from various problems such as single point of failure and data leakage. Therefore, in this paper, we propose a blockchain-based fair data trading model that does not rely on a fixed trust factor and supports bilateral access rights management. The proposed model is designed in a hybrid way that combines on-chain and off-chain processes to minimize the storage space and performance limitations of the blockchain. Instead of relying on a fixed trust factor such as a trusted third party, we randomly select arbiter nodes during each data trading process, which provides higher decentralization than existing approaches. It also provides data confidentiality by ensuring bilateral access control and provides security in the dispute resolution process. Finally, smart contracts in the proposed model perform only simple computations, resulting in lower overhead and higher efficiency than existing models.

Keywords: Data Trading, Fairness, Bilateral Access Control, Blockchain, Matchmaking Encryption.

1 Introduction

The Internet of Things (IoT) is the interaction of multiple smart devices, such as smartphones, wearables, and smart home devices, to send and receive services or data. IoT devices are being used in a variety of fields, including human, homes, transportation, hospitals, and more (Wei et al., 2021; Asokan et al., 1997). As a result, the data collected from IoT devices is more diverse and vaster than ever before, and in today's world where information is power, data is no longer just information, but a resource with asset value. Individuals and enterprises have begun to actively use online data trading or sharing systems to create new value from data or utilize it for financial gain. However, these systems face challenges such

Journal of Internet Services and Information Security (JISIS), volume: 14, number: 4 (November), pp. 569-589.
DOI: *10.58346/JISIS.2024.I4.036*

*Corresponding author: Division of Computer Engineering and Artificial Intelligence, Pukyong National University, Busan, Republic of Korea.

as reliability and data confidentiality. First of all, since online data trading systems involve entities that do not fully trust each other, data confidentiality and fair trade are important issues (Dai et al., 2019). Fair trade means that data providers want to be paid fairly for the data they provide, and data requesters want to receive legitimate and correct data for what they request and pay for. It also means that if the right data is not provided and the right payment is not made, no one gets what they want. However, in online data trading systems, it is difficult to achieve fair trading due to the lack of trust between participants. Therefore, traditional online data trading systems rely on trusted third parties to ensure fair trading. However, these centralized trading systems suffer from problems such as single points of failure and data leakage (Muralidharan, 2024).

Blockchain can be thought of as a distributed ledger that ensures transparency and trust (Bajoudah et al., 2019; Ziga Kodric & Jelovcan, 2021). Blockchain is decentralized database that is free from centralization, and verification and authentication are performed using a consensus mechanism among participants in the blockchain network. It is also characterized by the fact that once data is recorded on the blockchain, it is almost impossible to change the recorded data. By utilizing these features, blockchain can replace the role of trusted third parties that exist in traditional online data trading systems. While researches on decentralized and fair data trading protocols using blockchain technology are ongoing, most existing research still relies on a fixed trusted entity to resolve disputes, manage the keys of the cryptographic scheme applied, or create system-wide public parameters. The application of blockchain also offers advantages such as transparency, immutability, and traceability. However, blockchain has limitations in terms of data storage capacity and authentication of trading participants to each other. When data exceeds a certain size, the blockchain cannot store it entirely on the blockchain itself, so external storage must be used. After storing the original or encrypted data in the external storage, the data provider performs a data trading, and the data requestor receives the requested data from the external storage. At this point, data buyers and requesters may want to specify attributes for who is accessing the data and who is giving the data, respectively. This is a necessary part of secure and fair data trading, and it is necessary to design the system accordingly.

Therefore, in this paper, we propose a blockchain-based fair data trading model that does not rely on fixed trust factors and supports bilateral access rights management. The proposed model is designed in a hybrid way that combines on-chain and off-chain processes to minimize the storage space and performance limitations of the blockchain. In particular, it improves efficiency by reducing storage and complex operations performed by smart contracts on the on-chain. The main contributions of this paper are as follows:

- The proposed scheme provides a higher level of decentralization than existing methods by allowing randomly selected blockchain nodes to be involved in the generation of system-wide public parameters and keys during each data trading process, rather than relying on fixed trust entities.
- The proposed scheme ensures data confidentiality and bilateral access control. The Data Owner (DO) is able to specify access controls for counterparties to ensure the confidentiality of the trading data. The Data Requester (DR) is able to simultaneously determine whether the trading data comes from a legitimate source.
- The proposed protocol supports decentralized and fair exchange. Data owners are guaranteed to get paid for providing data, and data requesters who pay data owners are guaranteed to get accurate information about the data, i.e., the authenticity of the trading data is guaranteed. The DO-DR exchange process is guaranteed to be performed in a secure, fair, and decentralized manner.

- The proposed scheme ensures timeliness.

This paper is organized as follows: Section 2 describes the cryptographic techniques used to achieve fair data trading. Section 3 briefly specifies the components of the proposed model and explains the behavior of the model in each step. In Section 4, the security analysis and the comparison with existing schemes in terms of overhead and gas consumption are presented. Finally, Section 5 concludes the paper with a brief summary of the proposed model.

2 Cryptographic Primitives

This section describes the cryptographic schemes used in the proposed model. The proposed model uses modified certificateless matchmaking encryption (mCL-ME) and considers the following cryptographic primitives:

- Cryptographically secure hash function $H : \{0,1\}^* \rightarrow \{0,1\}^\lambda$, where λ is the security parameter (Al-Kuwari et al., 2011).
- Symmetric cryptographic algorithm that is semantically secure against chosen ciphertext attack consisting of $(\mathrm{SymKGen}, \mathrm{SymEnc}, \mathrm{SymDec})$(Bellare et al., 1997).
- Asymmetric cryptographic algorithm that are semantically secure against attacks on chosen ciphertext attack consisting of $(\mathrm{AsymKGen}, \mathrm{AsymEnc}, \mathrm{AsymDec})$ (Goldwasser & Micali, 1984; Naor & Yung, 1990).
- Secure digital signature scheme in existential unforgeability under chosen message attack consisting of polynomial time algorithms $(\mathrm{SignKGen}, \mathrm{Sign}, \mathrm{Verify})$ (Goldwasser et al., 1988).

All-or-Nothing Transform

All-or-Nothing Transform (AONT) introduced by Rivest (Zhang et al., 2004) is a block cipher technique that does not change the key length. Additionally, each bit of the encrypted resource depends on every bit of the plaintext resource, meaning complete interdependence (Abbadini et al., 2024). Complete interdependence means that if even one block of the ciphertext is missing, decryption of the encrypted resource is impossible. AONT encryption mode operates as follows:

- Divide data M into block-sized units: $M = m_1, \dots, m_n$.
- Randomly select a 256-bit r.
- For $i = 1, \dots, \mathrm{n}$, $c_i = m_i \oplus G(r||i)$, where $G()$ is a cryptographically secure pseudo-random function.
- $c_0 = r \oplus H(c_1|| \dots ||c_n)$, where $H()$ is a cryptographically secure hash function.
- $(Stub||Package) = (c_0\big|\big|c_1\big|\big| \dots ||c_n)$, where Stub can be set to the first few blocks, such as $Stub = c_0||c_1$ and $Package = c_2|| \dots ||c_n$.

The inversion of AONT encryption mode operates as follows:

- $(Stub, Package) = (c_0\big|\big|c_1\big|\big| \dots ||c_n)$.
- $r = c_0 \oplus H(c_1|| \dots ||c_n)$.
- For $i = 1, \dots, \mathrm{n}$, $m_i = c_i \oplus G(r||i)$.

The proposed model assumes that a cryptographically secure hash function such as SHA-2 or SHA-3 that supports at least 256-bit output is used to preprocess the data owner's data during the AONT's preprocessing process.

mCL-ME

The proposed model uses mCL-ME, a modified version of the existing CL-ME (Chen et al., 2021). The CL-ME technique is a cryptographic primitive consisting of five polynomial-time algorithms: $\mathsf{Setup}, \mathsf{RKGen}, \mathsf{SKGen}, \mathsf{Enc}, \mathsf{Dec}$. The mCL-ME method used in the proposed method uses most of the existing algorithms of the CL-ME method, with the exception of the Enc algorithm, which is modified to compute an additional value that any entity (including the receiver, which is the buyer of the data) can use to verify the source and validity of the encrypted data. The mCL-ME technique also provides the Verify algorithm. This algorithm allows any entity, including blockchain nodes, to verify the provenance and validity of encrypted data. For more details on the CL-ME technique, we can refer to (Chen et al., 2021).

The existing CL-ME technique requires a single, fixed Key Generation Center (KGC) that is responsible for initializing system parameters and generating partial secret keys for the DO and the DR. However, in the proposed model, the blockchain node (BN) performs the role of KGC, and it is possible for the DO to select a different BN for each trading data. This can solve the problem of a single fixed KGC always remaining online, and also solve the Single Point of Failure (SPOF) problem that can occur through Denial-of-Service (DoS) attacks on the KGC. Additionally, the proposed technique adds the Verify algorithm to the existing CL-ME technique to verify the validity of trading data. The specific mCL-ME techniques are as follows:

- $mCLME_{Setup}(1^{\lambda}) \rightarrow (mpk, msk)$: This algorithm is implemented by KGC (or BN in the proposed protocol). It accepts the security parameter λ as input and generates the master public key denoted as mpk and master private key referred to as msk.
  - e: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$: It is symmetric pairing. Here $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups with the order q and the generator P.
  - $H_1, H_2: \{0,1\}^* \rightarrow \mathbb{G}$, $\widehat{H}: \mathbb{G}_T \rightarrow \{0,1\}^l$ are different cryptographic hash functions.
  - ϕ: $\{0,1\}^* \rightarrow \{0,1\}^l$ is a padding function that can be computed in polynomial time. This requires that for every m ∈ $\{0,1\}^*$, m can be determined to be padded exactly in polynomial time and that ϕ(m) can be efficiently inverted.
    1. a) It takes the security parameter λ as input.
    2. b) Choose two random numbers $r, s \in \mathbb{Z}_q^*$ and calculate $P_0 = r \cdot$ P.
    3. c) Output the master public key mpk and master private key msk.
       - mpk = $(\mathrm{e}, \mathbb{G}, \mathbb{G}_T, P, H_1, H_2, \widehat{H}, \phi, P_0)$
       - msk = (r, s)
- $mCLME_{SKGen}(msk, \sigma) \rightarrow (ek_{\sigma}, P_{\sigma})$: This algorithm is executed jointly by the sender with identity σ and KGC. It is provided with the master private key denoted as msk and the sender's identity represented by σ. In response, it generates the encryption private key ek_{σ} and the encryption public key P_{σ} associated with the sender.
  - It takes KGC's master private key msk and sender identity σ as input.

  - KGC first calculates the partial key $ek_\sigma^1 = s \cdot H_2(\sigma)$.
  - A sender with identity σ randomly selects $ek_\sigma^2 \in \mathbb{Z}_q^*$ and then calculates $P_\sigma = ek_\sigma^2 \cdot P$.
  - Output the encryption private key $ek_\sigma = (ek_\sigma^1, ek_\sigma^2)$ and the encryption public key P_σ.

- $mCLME_{RKGen}(msk, \rho) \to (dk_\rho, P_\rho)$: This algorithm is executed jointly by the receiver with identity ρ and KGC. It receives the master private key msk and the receiver identity ρ as input, and output the decryption private key $dk_\text{ρ}$ and decryption public key $P_\text{ρ}$ for the receiver.
  - It takes KGC's master private key msk and receiver identity ρ as input.
  - KGC first calculates the partial keys $dk_\rho^1 = r \cdot H_1(\rho)$, $dk_\rho^2 = s \cdot H_1(\rho)$.
  - The receiver with identity ρ randomly selects $dk_\rho^3 \in \mathbb{Z}_q^*$ and then calculates $P_\rho = dk_\rho^3 \cdot P$.
  - Output the decryption private key $dk_\rho = (dk_\rho^1, dk_\rho^2, dk_\rho^3)$ and decryption public key P_ρ for the receiver.

- $mCLME_{Enc}(mpk, ek_\sigma, rcv, P_{rcv}, m) \to C$: This algorithm is run by the sender, who inputs the encryption private key ek_σ, the identity of the receiver rcv, the associated decryption public key P_{rcv}, and the message m intended for transmission. The output of this process is the ciphertext C.
  - It receives as input the encryption private key ek_σ, the receiver's identity $rcv = \rho$, the corresponding decryption public key $P_{rcv} = P_\rho$, and he message $m \in \{0,1\}^n$ to be shares.
  - To encrypt, do the following:
    i. Select two random numbers $t, u \in \mathbb{Z}_q^*$ and calculate $T = t \cdot P$, $U = u \cdot P$.
    ii. Calculate $k_R = e(H_1(\rho) + P_\rho, u \cdot P_0)$, $k_S = e(H_1, T + ek_\sigma^1 + ek_\sigma^2 \cdot P_\rho)$.
    iii. Calculate $V = \phi(m) \oplus \widehat{H}(k_R) \oplus \widehat{H}(k_S)$.
    iv. The value w_1 for verifying the sender and ciphertext is calculated as follows: $w_1 = t \cdot H_2(\sigma) + ek_\sigma^2 \cdot H_1(V)$.
    v. Output ciphertext $C = (T, U, V, w_1)$.

- $mCLME_{Dec}\left(mpk, dk_\rho, snd, P_{snd}, C\right) \to m \ or \ \perp$: This algorithm is implemented by the recipient, who inputs the decryption private key dk_ρ, the identity of the sender snd, the associated encryption public key P_{snd}, and the ciphertext C. Message m is output only if the ciphertext C generated by the sender's identity snd is related to the receiver's identity ρ. Otherwise, an error ⊥ is output.
  - It takes as input the decryption private key dk_ρ, the sender identity $snd = \sigma$, the corresponding encryption public key $P_{snd} = P_\sigma$, and the ciphertext C.
  - To decrypt, do the following:
    i. Parse the ciphertext C into (T, U, V, w_1).
    ii. Calculate $k_R = e\left(dk_\rho^1 + dk_\rho^3 \cdot P_0, U\right)$ and $k_S = e(H_1(\rho), T)e\left(dk_\rho^2, H_2(\sigma)\right)e\left(H_1(\rho), dk_\rho^3 \cdot P_\sigma\right)$.
    iii. Calculate $\phi(m) = V \oplus \widehat{H}(k_R) \oplus \widehat{H}(k_S)$.
    iv. If the padding is valid, it returns m. Otherwise, output ⊥.

- $mCLME_{Verify}(mpk, C, snd, P_{snd}) \rightarrow 1\ or\ \perp$: The DO's identity and ciphertext are input, and it is verified whether the source of the ciphertext is a legal DO or trading data between a legal DO and the DR and whether it is a valid ciphertext.
  - Verifies whether C is a valid ciphertext generated by the sender's identity $snd = \sigma$.
  - It receives the sender's identity $snd = \sigma$,the corresponding encryption public key $P_{snd} = P_{\sigma}$, and ciphertext C as input,
  - To verify, do the following:
    1. Parse the ciphertext C into (T, U, V, w_1).
    2. Verify that the following holds: $e(w_1, P) \overset{?}{\Leftrightarrow} e(H_2(\sigma), T)e(H_1(V), P_{\sigma})$.
    3. If established, it can be confirmed that it is a valid ciphertext C generated by the sender's identity $snd = \sigma$. Otherwise, it returns the error symbol $\perp$.

The $Verify$ algorithm $mCLME_{Verify}()$ can be used by any entity, including blockchain nodes (including receivers who purchase data), to verify the origin and validity of encrypted data.

The security of the mCL-ME technique inherits the security of the existing CL-ME technique under the standard Bilinear Diffie-Hellman (BDH) assumption. The mCL-ME scheme ensures security against indistinguishability under a chosen-plaintext attack (IND-CPA) with respect to the privacy, as well as safeguarding against existential unforgeability under a chosen message attack (EU-CMA) in relation to the authenticity. The formal proof of the specific security follows the security proof of the CL-ME method (Chen et al., 2021), and the correctness of the mCL-ME method can also be found in (Chen et al., 2021). The correctness of the $mCLME_{Verify}$ algorithm added by the mCL-ME technique can be verified by the following:

$$\begin{aligned} e(w_1, P) &= e(t \cdot H_2(\sigma) + ek_{\sigma}^2 \cdot H_1(V), P) \\ &= e(t \cdot H_2(\sigma), P) \cdot e(ek_{\sigma}^2 \cdot H_1(V), P) \\ &= e(H_2(\sigma), t \cdot P)e(H_1(V), ek_{\sigma}^2 \cdot P) \\ &= e(H_2(\sigma), T)e(H_1(V), P_{\sigma}) \end{aligned}$$

3 Proposed Model

In this section, we propose a fair data trading scheme that supports blockchain-based bilateral access rights management. First, we describe the assumptions and threat model, then outline the components of the proposed scheme, and finally, explain the specific protocol behavior.

Threat Model and Assumptions

In the proposed model, the attacker is assumed to be a static corruption (Abe & Ohkubo, 2012) that can only corrupt an entity before the protocol execution process, and we consider only the stand-alone (Lindell, 2009) model. We also assume that the adversary has computationally limited capabilities. To describe the adversary's ability to control communication, the proposed model assumes a synchronized network model of authenticated point-to-point channels, i.e., for any message passed between honest parties, the adversary can only delay it by a known amount Δ in advance, and cannot delete, reroute, or modify it. Finally, we will assume the presence of a global clock within the system, without loss of generality (Kosba et al., 2016).

The proposed model is designed based on these attacker models, as well as the following assumptions:

- It is assumed that the participating entities, the DO and the DR, do not trust each other. This means that each participating entity can perform actions to deceive the other for its own benefit. For example, a DO might receive a legitimate payment for data from a DR, but fail to deliver the data or deliver false or incorrect data. Conversely, the DR might attempt to obtain the data it wants without paying for it.
- The blockchain used in the system is assumed to inherit the security of the underlying blockchain platform, i.e., it assumes the basic security of the underlying blockchain, such as integrity and tamper immunity.
- It is assumed that the participating blockchain consensus nodes are not fully trusted entities. Therefore, when selecting blockchain nodes to act as trading arbitrators, it is important to randomize the selection to ensure that no blockchain node can intentionally participate in a particular trading to cheat.

System Architecture and Main Design Goals

This section describes the components of the proposed scheme. Figure 1 shows the overall flow of the proposed model.

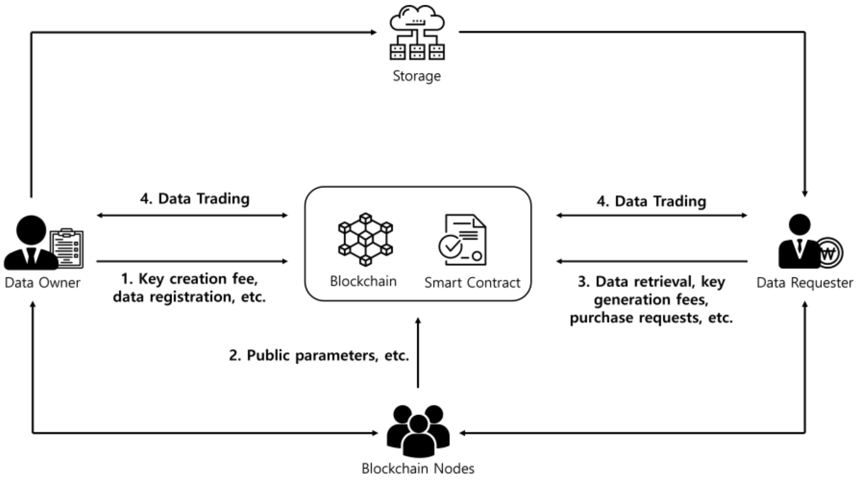

Storage
4. Data Trading
4. Data Trading
Data Owner
1. Key creation fee, data registration, etc.
Blockchain
Smart Contract
3. Data retrieval, key generation fees, purchase requests, etc.
Data Requester
2. Public parameters, etc.
Blockchain Nodes

Figure 1: Overview of Proposed Model

- Data Owner (DO): The DO wants to trade its own data using flexible access control policies. To do this, the DO participates in blockchains and peer-to-peer (P2P) networks. The DO is assumed to have its ID and a key pair. The DO also wants the arbitration of a fair exchange to a third party by promising to pay a Fee_{bn} in return for successful data trading. Finally, the DO uses a one-time key pair (PK_{DO},SK_{DO}) for each data trading.
- Data Requester (DR): The DR wants to buy the data from the DO who owns the data it wants. To do so, the DR must also participate in the blockchain and the P2P network, assuming they have their ID and a key pair. The DR pays the DO the amount $Value_M$ for the data provided. The DR also uses a one-time key pair (PK_{DR},SK_{DR}) for each data trading.
- Blockchain/Smart Contract (SC): It is assumed to be designed based on a blockchain infrastructure that is secure, reliable, and supports SC. In addition, a P2P network is used to

facilitate communication between participating entities. The SC is a form of stateful ideal functionality (He et al., 2021) that exposes all internal states to all components, including attackers. The SC can transparently mimic smart contracts in real life by specifying an immutability condition in advance to trade "coins" on the cryptocurrency ledger. Moreover, practical implementations of smart contracts can be executed via blockchain platforms, such as Ethereum. In this paper, the notation for the SC adheres to a pseudo-code format of (Kosba et al., 2016).

- Blockchain Node (BN): For an entity that participates in consensus on a blockchain infrastructure, the selection of a BN depends on the blockchain infrastructure used. If a BN behaves unfairly, it will forfeit the Deposit it made to participate as an arbiter node. It is assumed that BN also has its own ID and key pair. The BN, functioning as an arbitrator node, employs a one-time key pair for each instance of data trading.
- Storage: Distributed storage, such as Cloud Storage or IPFS (Inter Planetary File System), can be used. Storage must ensure reliability and availability in terms of data management. Given that a storage is not fully trusted, data owners encrypt their data to maintain confidentiality prior to outsourcing it to the sotrage.

The main design goals of the proposed scheme are as follows:

- It should provide a higher level of decentralization than existing methods.
- Data confidentiality and bilateral access control against non-adaptive Probabilistic Polynomial Time (PPT) adversary must be ensured in a synchronous authenticated network and stand-along model.
- In a synchronous authenticated network and stand-along model, it is essential that it upholds the fairness requirement for all parties involved, even in the event that either the DO or the DR is compromised by a non-adaptive PPT adversary.
- Timeliness must be guaranteed in a synchronous authenticated network and stand-along model if the DO or the DR is honest.

Data Trading Protocol

This section describes the proposed trading protocol in detail. The proposed model applies AONT and mCL-ME as main cryptographic primitives, and consists of the following steps: Prepare, Publish, Request, Trading, Download & Validate, and Dispute. Table 1 summarizes the notation used in the proposed protocol.

The proposed scheme guarantees fairness requirement for all parties involved, even in the event that either the DO or the DR is compromised by a non-adaptive PPT adversary. In the proposed model, the SC is stateful idealized functions that have access to the blockchain to support fair data transactions and are described according to the customary pseudo-code notation for the SC (Kosba et al., 2016; He et al., 2021). This means that a SC with Turing completeness is a stateful program that can transparently handle predefined functionalities. Furthermore, the SC has access to cryptocurrency blockchains that perform conditional payments based on their own internal stored state. The proposed scheme's SC is designed to execute only lightweight operations, including comparison, allocation, storage, addition, and subtraction, rather than performing complex and recource-intensive computations. The more intricate cryptographic operations are carried out locally by the participating entities. While blockchain platforms like Ethereum offer support for Turing completeness is smart contracts, the implementation of intricate cryptographic operations, such as zero-knowledge proofs, in smart contracts remains a considerable challenge due to the associated processing time and costs. Figure 2 shows a formalization of the behavior of smart contracts in the proposed model. The following is a step-by-step description of how the proposed model works.

Table 1: Notation of the Proposed Model

Symbols	**Details**	**Symbols**	**Details**
P_0	Master public key	PP	Public parameter
msk	Master private key	Fee_{BN}	Key generation fee
$Deposit_{XX}$	XX's deposit	ID_{XX}	XX's identity
T_{XX}	Timestamp	$Sign_{XX}$	XX's signature
σ	DO's identity attribute	ρ	DR's identity attribute
ek_σ	Encryption private key	P_σ	Encryption public key
dk_ρ	Decryption private key	P_ρ	Decryption public key
r	DO's data trading count value	M	Data
$Keyword_M$	Keyword required when retrieving data	$Terms_M$	Terms required when trading data
$Addr_{\sigma-M}$	Stored data address	$Price_M$	M's price

(1) Prepare & Setup Phase

In this step, the DO first deploys the SC and the BN releases the public parameter PP, including the master public key P_0, to the blockchain. The BN performs the $mCLME_{Setup}$ algorithm of the mCL-ME technique to generate the public parameter PP and the master public key P_0, while keeping the master private key msk secure. The BN's identity, key generation fee Fee_{BN}, deposit $Deposit_{BN}$, and the validity period T_{PP} of the transaction are posted to the blockchain along with the public parameter PP and the signature value $Sign_{PP}$. Here, $Deposit_{BN}$ is the amount of money that will be forfeited if the BN is found to be cheating.

When the SC receives information from the BN regarding the public parameter, it first checks the BN's account balance. If the BN's account balance is greater than or equal to $Deposit_{BN}$, it checks if the time of the timestamp is within T_{PP} and then sends $\{SetupTx\} \coloneqq \{Setup, ID_{BN}, PP, P_0, Fee_{BN}, Deposit_{BN}, T_{PP}, Sign_{PP}\}$ to all nodes.

(2) PP Selection and DO Key Generation Phase

Based on Algorand (Chen & Micali, 2016), an unpredictable leader selection technique, the DO randomly selects an arbiter node and public parameters for use in the trading. The DO selects the parameters of the node with the smallest value of $.H(\sigma, r, H(M), Sign_{PP})$ among the public parameters. Where r is the DO's data trading count value and $.H(x)$ is the hash value of x expressed as a 256-bit binary number in the interval [0,1]. These values are published on the blockchain during the publication phase of trading data, and any participant can verify the selection of the arbiter node by calculating the $.H(\sigma, r, H(M), Sign_{PP})$ value.

The DO utilizes the information from the selected BN to perform $mCLME_{SKGen}$ algorithm. In addition, the DO interacts with the selected BN to generate the following key pairs for the encryption of the trading data.

$$\text{Encryption private key } ek_\sigma = (ek_\sigma^1, ek_\sigma^2)$$

$$\text{Encryption public key } P_\sigma = ek_\sigma^2 \cdot P$$

The DO sends $\{DOpayTx\} \coloneqq \{DOpay, \sigma, ID_{BN}, Fee_{BN}\}$ to the SC to pay the fee for key generation, where σ is the DO's identity attribute and Fee_{BN} is the key generation fee to be paid. When the SC receives information from the DO about the payment of the key generation fee, it checks the DO's account balance. If the DO's account balance is equal to or greater than the Fee_{BN}, it subtracts the Fee_{BN} from the DO's account balance and then the Fee_{BN} is frozen (this is paid to BN upon normal completion of the trading or resolution of the dispute). However, if the DO's account balance is less than the fee, the trading is terminated.

(3) Data Upload and Trading Data Publishing Phase

The DO converts data M to be traded into $M' = (Stub||Package)$ pair through AONT processing. The DO stores $(package, Sign_{pkg})$ to the external storage and obtains the stored address $Add_{\sigma-M}$. Here, $Sign_{pkg} = Sign(Package, SK_{DO})$ is the signature for $Package$. The DO transmits $\{SellTx\} \coloneqq \{Sell, \sigma, H(M), H'(M), Keyword_M, Terms_M, T_S, Price_M, Deposit_{DO}, P_\sigma, TxID_{PP}\}$ including metadata for data M to the SC. The SC receives information about trading data from the DO and operates as follows: First, after checking whether it was received within T_S, set the state variable Σ to 'registered'. Then, $\{SellTx\}$ is sent to all nodes. If the DO's account balance is less than $Deposit_{DO}$, the trading is terminated.

(4) Data Retrieval and DR Key Generation Phase

The DR retrieves the desired data through searching the metadata posted on the blockchain and then performs $mCLME_{RKGen}$ algorithm using the information of the BN selected for the trading. Based on $mCLME_{RKGen}$, interaction with the BN is performed to generate the following key pairs for decrypting the trading data.

$$\text{Decryption private key } dk_\rho = \left(dk_\rho^1, dk_\rho^2, dk_\rho^3\right)$$

$$\text{Decryption public key } P_\rho = dk_\rho^3 \cdot P$$

The DR sends $\{DRpayTx\} \coloneqq \{DRPay, \rho, ID_{BN}, Fee_{BN}\}$ to SC to pay the fee for key generation. The ρ is the identity of DR and Fee_{BN} is the key generation fee to be paid. When the SC receives information about the key generation fee from the DR, it checks the DR's account balance. If the DR's account balance is more than Fee_{BN}, the Fee_{BN} is deducted from the DR's account balance and Fee_{BN} is frozen. However, if DR's account balance is less than Fee_{BN}, the trading is terminated.

(5) Buy Request Phase

The DR transmits $\{BuyTx\} \coloneqq \left\{Buy, \rho, H(M), Price_M, Deposit_{DR}, P_\rho, T_B\right\}$ to the SC for trading the retrieved data. When the SC receives information about a purchase request from the DR, it operates as follows: First, check whether the state variable Σ is 'registered' and whether the trading request came within T_S. Additionally, the DR's account balance is checked and if it is less then $(Price_M + Deposit_{DR})$, the trading is terminated. If it is not less than $(Price_M + Deposit_{DR})$, the state variable Σ is set to 'buy-req' and $\{BuyTx\}$ is sent to all nodes.

(6) Trading Phase

When the DO receives the trading request $\{BuyTx\}$, it checks whether the state variable Σ is 'buy-req' and checks whether the trading request came within T_S. Then, the $mCLME_{Enc}$ encryption algorithm is performed by applying the identity ρ of the DR. It computes the ciphertext $C = eStub = (T, U, V, w_1) = mCLME_{Enc}(P_0, ek_\sigma, \rho, P_\rho, \left(Stub\middle||Addr_{\sigma-M}\right))$ that encrypts $(Stub||Addr_{\sigma-M})$ by applying $DO's$ encryption private key ek_σ, DR's identity attribute ρ, and DR's corresponding decryption public key P_ρ. Additionally, the signature $Sign_{Stub} \leftarrow$

$Sign((Stub||Addr_{\sigma-M}), SK_{DO})$ for validation is calculated.

Then, the DO transmits the trading data $\{TradeTx\} \coloneqq \{Trade, \sigma, \rho, ID_{BN}, H(M), eStub, Sign_{Stub}, T_T\}$ to the SC. At this time, if the size of the $eStub$ is relatively large, it is possible to directly deliver the $eStub$ to the DR off-chain and store only the hash value of this information in on-chain. When the SC receives information about the trading from the DO, it operates as follows: Check whether the state variable Σ is 'buy-req' and check whether the time for receiving trading-related data is within T_B. If it is received after T_B, change the state variable Σ to 'cancelled' and cancel the trading. If received within T_B, set statues Σ to 'traded' after checking information such as T_T. Finally, $\{TradeTx\}$ is sent to all nodes.

(7) Data Download and Validation Phase

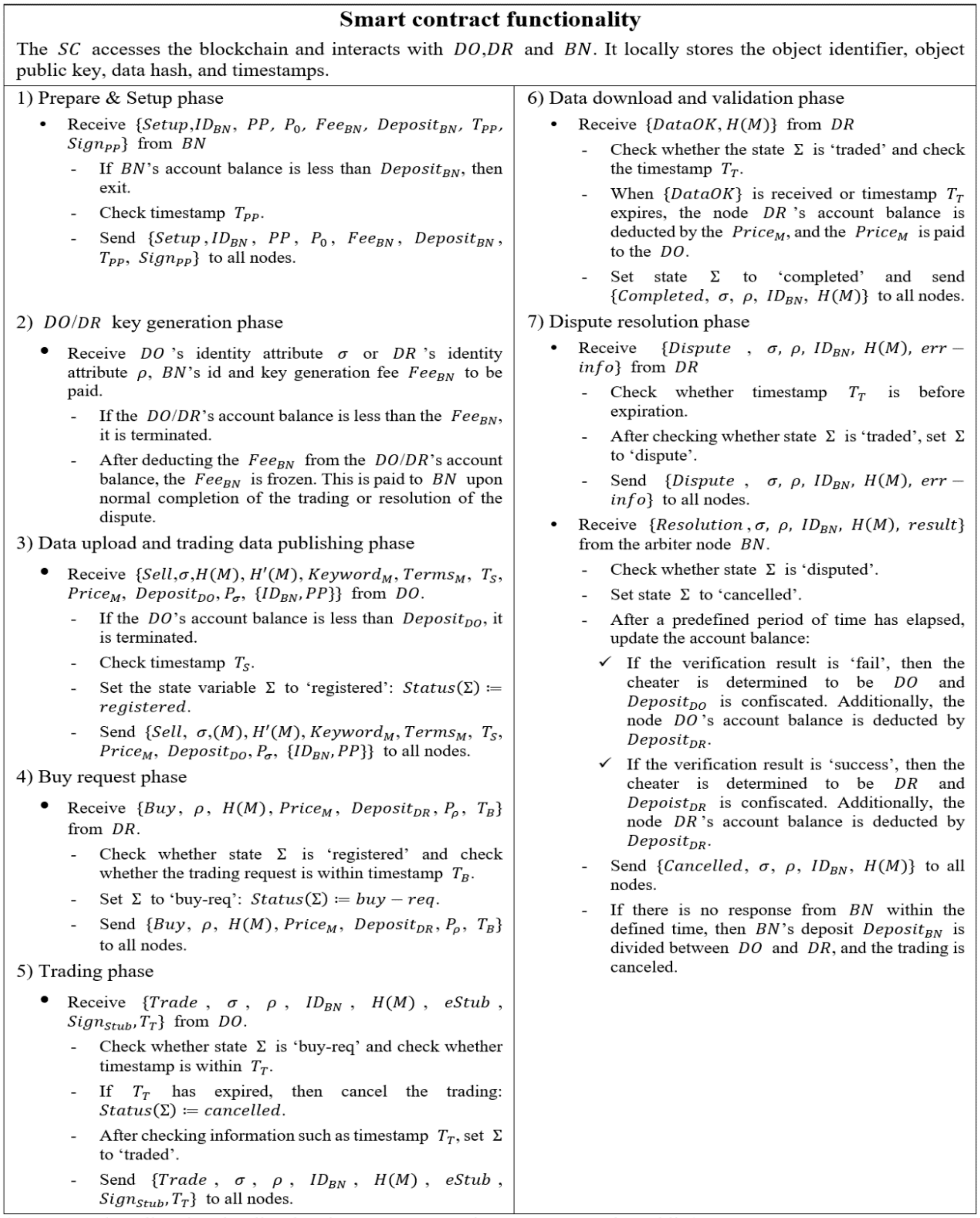
When DR receives $\{TradeTx\}$, it first checks whether the reception time is within T_B and checks whether the state variable Σ is 'traded'. Then, the $mCLME_{Dec}$ algorithm is performed. It obtains $(Stub'||Addr_{\sigma-M})$ by decrypting $eStub$ using the DR's decryption private key dk_ρ, the DO's identity σ, and public key P_σ corresponding to the DO. Afterward, $(Package', Sign'_{pkg})$ is downloaded from the $Addr_{\sigma-M}$ address of the external storage, and it verifies the signature $Sign'_{pkg}$ and $M' = (Stub', Package')$'s signature $Sign(M')$. In addition, by performing the inverse transformation of an AONT mode for $(Stub', Package')$, the trading data M is obtained, and then $H(M)$ is verified. If there is a problem with validity, an objection can be raised within T_T, and in this case, the dispute resolution step will proceed. If the validity verification of the restored data M is successful, $\{DataOKTx\} \coloneqq \{DataOK, H(M)\}$ is transmitted to the SC. When the SC receives information regarding the trading completion from the DR, it performs the following: First, check whether the state Σ is 'traded' and then check whether the received time is within T_T. If related information is received after T_T, $Price_M$ is deducted from the node DR's account balance and $Price_M$ is paid to the DO. Then, set the state Σ to 'completed' and send $\{CompletedTx\}$ to all nodes.

(8) Dispute Resolution Phase

If the honest DR obtains invalid data, dispute resolution steps are taken. The dispute resolution mechanism verifies whether data that matches the published hash value or has a valid signature value has been decrypted. If a problem occurs in the validity of the restored data M, the DR requests the SC and the arbiter node BN to resolve the dispute within T_T. When the SC receives $\{DisputeTx\} \coloneqq \{Dispute, \sigma, \rho, ID_{BN}, H(M), err - info\}$, it does the following: First, it checks whether T_T has expired and the state variable Σ is 'traded', then set the state to 'dispute'. And then $\{DisputeTx\}$ is sent to all nodes.

When the arbiter node A receives $\{DisputeTx\}$, it performs the following: Verify whether the state variable Σ is 'dispute' and check $err - info$. Then, verification is performed by executing mCL-ME's $Verify$ algorithm. The verification process retrieves $eStub$ information on-chain and then performs the $mCLME_{Verify}$ algorithm using σ, ρ and w_1 values, where σ and ρ are the identity information of the DO and the DR, respectively. If validation is successful, 1 will be returns as a result. If validation fails, then ⊥ will be returned as a result. When validation is completed, $\{ResolTx\} \coloneqq \{Resolution, \sigma, \rho, ID_{BN}, H(M), result\}$ specifying the verification result is sent to the SC.

When the SC receives the $\{ResolTx\}$ message from the arbiter node, it performs the following: After checking whether the state variable Σ is 'disputed', change it to 'cancelled'. The account balance is updated after a predefined time has elapsed. If the verification result is a failure, the cheater is determined to be a DO, $Deposit_{DO}$ is confiscated, and the node DO's account balance is deducted by $Deposit_{DO}$. However, if the verification result is successful, it is determined that the cheater is $DR, Deposit_{DR}$ is confiscated, and the account balance of the node DR is deducted by $Deposit_{DR}$. Additionally, if there is no response from BN within the predefined time, BN's deposit $Deposit_{BN}$ is confiscated and distributed to the DO and the DR and the transaction is cancelled. Finally, $\{cancellTx\} \coloneqq \{Cancelled, \sigma, \rho, ID_{BN}, H(M)\}$ is sent to all nodes.



Smart contract functionality

The SC accesses the blockchain and interacts with DO,DR and BN. It locally stores the object identifier, object public key, data hash, and timestamps.

1) Prepare & Setup phase

- Receive $\{Setup, ID_{BN},\ PP,\ P_0,\ Fee_{BN},\ Deposit_{BN},\ T_{PP},\ Sign_{PP}\}$ from BN
    - If BN's account balance is less than $Deposit_{BN}$, then exit.
    - Check timestamp T_{PP}.
    - Send $\{Setup, ID_{BN},\ PP,\ P_0,\ Fee_{BN},\ Deposit_{BN},\ T_{PP},\ Sign_{PP}\}$ to all nodes.

2) DO/DR key generation phase

- Receive DO's identity attribute σ or DR's identity attribute ρ, BN's id and key generation fee Fee_{BN} to be paid.
    - If the DO/DR's account balance is less than the Fee_{BN}, it is terminated.
    - After deducting the Fee_{BN} from the DO/DR's account balance, the Fee_{BN} is frozen. This is paid to BN upon normal completion of the trading or resolution of the dispute.

3) Data upload and trading data publishing phase

- Receive $\{Sell, \sigma, H(M), H'(M), Keyword_M, Terms_M,\ T_S,\ Price_M,\ Deposit_{DO}, P_\sigma,\ \{ID_{BN}, PP\}\}$ from DO.
    - If the DO's account balance is less than $Deposit_{DO}$, it is terminated.
    - Check timestamp T_S.
    - Set the state variable Σ to 'registered': $Status(\Sigma) \coloneqq registered$.
    - Send $\{Sell,\ \sigma, (M), H'(M), Keyword_M, Terms_M,\ T_S,\ Price_M,\ Deposit_{DO}, P_\sigma,\ \{ID_{BN}, PP\}\}$ to all nodes.

4) Buy request phase

- Receive $\{Buy,\ \rho,\ H(M), Price_M,\ Deposit_{DR}, P_\rho,\ T_B\}$ from DR.
    - Check whether state Σ is 'registered' and check whether the trading request is within timestamp T_B.
    - Set Σ to 'buy-req': $Status(\Sigma) \coloneqq buy - req$.
    - Send $\{Buy,\ \rho,\ H(M), Price_M,\ Deposit_{DR}, P_\rho,\ T_B\}$ to all nodes.

5) Trading phase

- Receive $\{Trade,\ \sigma,\ \rho,\ ID_{BN},\ H(M),\ eStub,\ Sign_{Stub}, T_T\}$ from DO.
    - Check whether state Σ is 'buy-req' and check whether timestamp is within T_T.
    - If T_T has expired, then cancel the trading: $Status(\Sigma) \coloneqq cancelled$.
    - After checking information such as timestamp T_T, set Σ to 'traded'.
    - Send $\{Trade,\ \sigma,\ \rho,\ ID_{BN},\ H(M),\ eStub,\ Sign_{Stub}, T_T\}$ to all nodes.

6) Data download and validation phase

- Receive $\{DataOK, H(M)\}$ from DR
    - Check whether the state Σ is 'traded' and check the timestamp T_T.
    - When $\{DataOK\}$ is received or timestamp T_T expires, the node DR's account balance is deducted by the $Price_M$, and the $Price_M$ is paid to the DO.
    - Set state Σ to 'completed' and send $\{Completed,\ \sigma,\ \rho,\ ID_{BN},\ H(M)\}$ to all nodes.

7) Dispute resolution phase

- Receive $\{Dispute,\ \sigma,\ \rho,\ ID_{BN},\ H(M),\ err-info\}$ from DR
    - Check whether timestamp T_T is before expiration.
    - After checking whether state Σ is 'traded', set Σ to 'dispute'.
    - Send $\{Dispute,\ \sigma,\ \rho,\ ID_{BN},\ H(M),\ err-info\}$ to all nodes.
- Receive $\{Resolution, \sigma,\ \rho,\ ID_{BN},\ H(M),\ result\}$ from the arbiter node BN.
    - Check whether state Σ is 'disputed'.
    - Set state Σ to 'cancelled'.
    - After a predefined period of time has elapsed, update the account balance:
        - ✓ If the verification result is 'fail', then the cheater is determined to be DO and $Deposit_{DO}$ is confiscated. Additionally, the node DO's account balance is deducted by $Deposit_{DR}$.
        - ✓ If the verification result is 'success', then the cheater is determined to be DR and $Depoist_{DR}$ is confiscated. Additionally, the node DR's account balance is deducted by $Deposit_{DR}$.
    - Send $\{Cancelled,\ \sigma,\ \rho,\ ID_{BN},\ H(M)\}$ to all nodes.
    - If there is no response from BN within the defined time, then BN's deposit $Deposit_{BN}$ is divided between DO and DR, and the trading is canceled.

(Here, "Send to all nodes" implies that the smart contract is transparent to the public.)

Figure 2: Smart Contract Functionality for the Proposed Scheme

4 Analysis

In this section, we show that the proposed scheme fulfills all the requirements that we set out to achieve. First, the proposed method ensures fairness, data confidentiality, and timeliness, and secondary reselling resistance. In addition, unlike existing methods, it does not depend on a trusted third party (TTP), but rather a randomly selected node among the nodes participating in the blockchain for each data trading process, thus ensuring a higher degree of decentralization.

Security Analysis

Lemma 1 (completeness). Provided that all participating parties are honest, the proposed protocol satisfies the completeness requirement in a synchronous authentication network and the stand-alone model.

Proof. The completeness of the proposed technique can be confirmed immediately. If both the DR and the DO follow the protocol honestly, the DO earns the net profit of $Price_M$ in step 6. The DR also obtains the valid data M in step 6.

Lemma 2 (fairness). Under the assumption that the underlying cryptographic mechanisms used in the proposed scheme in the synchronous authentication model and stand-alone setting are secure, the proposed scheme satisfies the fairness requirement even if one party of the DO or the DR is compromised by non-adaptive PPT attacker.

Proof. The fairness of both the DO and the DR must be satisfied:

- Fairness of the DR: The fairness of the DR ensures that, regardless of any malicious actions by the DO, the honest DR is only responsible for payment for data that was legitimately and properly acquired. An adversary may try to disrupt or manipulate the delivery of $\{TradeTx\}$ at the trading phase by compromising the DO. At this time, if $\{TradeTx\}$ does not arrive within T_B, the SC cancels the trading. In other words, if an attacker disrupts the delivery of $\{TradeTx\}$, a normal trading will not take place, so the honest DR will not be able to obtain valid data and will not make payment for it. Additionally, an attacker has the ability to alter manipulate $\{TradeTx\}$, but the DR verifies the accuracy of the data acquired during the download & verification phases. Therefore, the fairness of DR is assured as long as an attacker is unalbe to (1) discover hash collisions, (2) create forged signatures, or (3) manipulate the execution of the SC on the blockchain. Finally, it satisfies the authenticity property of the trading data, which states that an attacker who does not possess the sender attribute can't generate a valid ciphertext. In other words, if the hash function, signature technique, and mCL-ME technique used are secure and the SC is modeled with ideal functionality, then the probability of violating the DR's fairness is negligible. Thus, the fairness of DR is guaranteed against a malicious DO.
- Fairness of the DO: The fairness of DO means that the honest DO is compensated equitably for valid data supplied to the DR. An attacker may attempt to violate an honest DO's fairness by allowing the DR to acquire valid data without paying for it. In the case of the proposed technique, in the data publishing phase, the DO discloses information of the trading data to the blockchain through $\{SellTx\}$ transmission. At this stage, only metadata for data M is disclosed, so actual valid data cannot be obtained. In the purchase request phase, the DR prepays for data M through $\{BuyTx\}$ and then obtains information about data M in step 5. In other words, during the trading process of the proposed technique, the DR can't acquire data M without paying the price. However, an attacker can try the following methods: An attacker may endeavor to present

information acquired through $\{TradeTx\}$ during step 5 to a DR that did not request a trading. In this case, $eStub$ can be obtained through $\{TradeTx\}$ in step 5, but this value cannot be decrypted because it is encrypted based on the mCL-ME technique using the decryption public key P_ρ of the $DR\ \rho$, the actual purchase requester. Therefore, $Stub$ and $Addr$ cannot be obtained. In other words, an attacker cannot obtain valid information for the DR that did not request a trading in step 4 from $\{TradeTx\}$ in step 5 and in order to obtain it, he must be able to violate the underlying cryptographic mechanism. Alternatively, an attacker may attempt to restore the original data M by obtaining only the $Package$ portion stored in external storage. To achieve this, the attacker must be able to break the applied AONT scheme. However, the fairness of the DO is guaranteed because an attacker cannot break the mCL-ME encryption technique and AONT scheme used in the proposed scheme.

Lemma 3 (resistance to collusion attacks). Provided that the arbitrator node selection method employed in the proposed model is secure, the technique is secure against collusion attacks involving both the attackers and the BN.

Proof. The fundamental premise of the proposed method is that an adversary is capable of compromising the involved entities solely prior to the execution of the protocol. Thus, in order for an adversary to carry out an attack through collusion with BN on the data trading, the BN that will carry out the collusion attack must be selected prior to execution of the trading protocol, and the selected BN should be designated as the arbiter node in the trading protocol for the target data. Due to the ingerent unpredictability of the output generated by the employed hash function and signature algorithm, an adversary is unable to foresee the specific BN that will be chosen prior to the initiation of the trading protocol. Consequently, the feasibility of a collusion attack involving the BN during the trading process is contingent upon the security of the leader selection mechanism of Algorand used to select the BN. In summary, provided that the employed hash function, signature algorithm, and Algorand's leader selection mechanism are secure, the likelihood of an attacker successfully executing a collusion attack with BN can be negligible.

Lemma 4 (confidentiality and bilateral access rights management). In a synchronous authentication network and stand-alone execution model, the proposed technique ensures confidentiality and bilateral access rights management against non-adaptive PPT attackers.

Proof. In the proposed model, the raw data M is transformed into a $(Stub, Package)$ by AONT mode before being outsourced to the outside and then traded, and only the $Package$ part is outsourced to external storage. At this time, because the $Stub$ and the storage addresses $Addr_{\sigma-M}$ of the $Package$ are disclosed in an encrypted form by the $mCLME_{Enc}$ algorithm, an entity that has acquired only the $Package$ cannot obtain the raw data M due to the nature of AONT mode. Therefore, for an PPT adversary to successfully breach the confidentiality of data M, he or she must (1) break the mCL-ME primitive used in $eStub$ encryption and (2) violate the AONT mechanism. That is, if the cryptographic mechanism employed are secure, the confidentiality of the proposed technique is assured against non-adaptive PPT adversaries. From the perspective of bilateral access rights management, the mCL-ME technique inherits the characteristics of the ME (Ateniese et al., 2021) technique, especially IB-ME. In the mCL-ME technique, the access policies established by both the sender and the receiver are specified concurrently. In other words, the bilateral access policy is specified by the encryption key ek_σ corresponding to the sender's identity attribute σ, and the decryption key dk_ρ corresponding to the receiver's attribute ρ. Accordingly, if the sender properties and the receiver properties are not satisfied, the ciphertext cannot be decrypted, and the property of authenticity that an attacker who does not possess the property σ a cannot generate a valid ciphertext is also satisfied.

Lemma 5 (timeliness). In the event that either the DO or the DR behaves honestly, the proposed scheme demonstrates compliance with timeliness within a synchronous authentication network and a stand-alon execution model.

Proof. Timeliness refers to the ability of an honest participant to consistently arrive at a stage in the protocol where it can be concluded, guaranteeing fairness within a limited timeframe. In the context of the proposed scheme involving the SC and the presence of at least one honest participant, the following termination scenarios exist:

- No abort: If all involved parties behave honesty, the proposed technique terminates in step 6 after $\{DataOKTx\}$ is received or the timestamp T_T expires. At this time, both the DR and the DO are guaranteed to end after obtaining what they want.
- Cancellation during the publishing phase: At the phase of publishing the trading data, the timestamp T_S denotes the deadline for trading requests, and if a trading request is not received from the DR prior to the expiration of T_S , the trading is canceled. At this point, fairness for both the DO and the DR is assured. In other words, the DO has not yet completely supplied data M and the DR has not initiated a request for a trading.
- Cancel at the request phase: In the trading request phase, the timestamp T_B specifies the deadline for providing information about data M obtained by the DO in the trading stage of step 5. T_B expiration means that DO did not provide $\{TradeTx\}$, so SC cancels the trading. When T_B expires, the DR receives back the amount paid, and the DO's fairness is guaranteed because the DO did not provide $\{TradeTx\}$.
- Cancel at the trading phase: At the trading stage, the timestamp T_T refers to the deadline for requesting the dispute resolution. The expiration of T_T means that there was no dispute request by the DR and consequently, the SC completes the protocol normally. At this point, the fairness of both parties is assured.
- Cancellation during the dispute resolution process: If a dispute resolution request is raised by the DR within the timestamp T_T, the dispute resolution step is performed. In this step, in the event that the $\{ResolTx\}$ message is not transmitted by the arbitrator node BN within the designated timeframe, the SC seizes the deposit of BN, distributes it to the DO and the DR and subsequently annul the trading transaction. Therefore, fairness is guaranteed at this point because the DO and the DR receive compensation for the cancellation of the trade.

Theorem 6. Provided that the applied cryptographic mechanisms are secure and the security of the blockchain is assured, the proposed model fulfills the criteria of completeness, fairness, confidentiality, bilateral access control, and timeliness within both the synchronous authentication network and stand-alone execution model.

Proof. It is assured by Lemma 1, 2, 3, 4, and 5.

In addition, the proposed technique supports security against the second reselling attack. Secondary reselling refers that a malicious buyer resells the data he purchased from the trading blockchain to make a profit. In the proposed scheme, the hash value associated with each data is recorded on the blockchain. And during the trading disclosure phase, when the seller reveals the trading data, the blockchain/SC verifies whether the hash of the disclosed trading data is a duplicate. If a duplicate is found, the owner of the raw data is checked, and if it is confirmed that the seller was the buyer in a previous trading, the trading can be invalidated. The proposed method effectively mitigates the occurrence of secondary reselling within the platform. Nevertheless, it is constrained by its inability to deter malicious actors from engaging in private reselling of the data through alternative means or on different platforms.

Comparative Analysis with Related Researches

In this section, we compare and analyze the characteristics of the proposed model with those of the existing researches on fair and secure data trading systems. Table 2 briefly compares the existing schemes with the proposed model.

Table 2: Comparison of Existing Schemes and the Proposed Scheme

	Alsharif & Nabil, (2020)	**Li et al., (2020)**	**Park et al., (2023)**	**Proposed scheme**
Main cryptographic primitives	CP-ABE, zk-SNARKs	PCE	IB-ME	mCL-ME, AONT
Data confidentiality	Confidentiality guaranteed during normal dealings, but compromised during disputes.	Confidentiality guaranteed during normal dealings, but compromised during disputes.	Confidentiality guaranteed during normal dealings, but compromised during disputes.	Guaranteed
Bilateral access control	One-way access control	One-way access control	Bilateral access control	Bilateral access control
Fair trading	Guaranteed	Guaranteed	Guaranteed	Guaranteed
Timeliness	Guaranteed	Guaranteed	Guaranteed	Guaranteed
Decentralization	It requires a full trusted entity of KDC.	There are no fully trusted entities, but the security of the scheme depends on the fairness of the miner. However, in blockchain, miners themselves are not trusted entities, and there are no rewards or punishments for miners.	There exists an arbiter element trusted by the participants in the trading process.	There are no established entities that can be considered fully trustworthy. A blockchain node is selected randomly to act as an arbiter prior to the trading process, and rewards and punishments exist for fair trading mediation.

Li et al., (2020) proposed a fair data trading platform based on blockchain and applied the PCE (Plaintext Checkable Encryption) technique to it. The scheme proposed by Li et al relies on miners to promote a fair trading process, but miners cannot be considered as fully trusted entities in the blockchain. In addition, miners play the role of arbiters who resolve the dispute that arises during the trading process and do not take into account miners committing illegal acts. In other words, it is assumed that miners simply execute and verify smart contracts based on the presented evidence. In addition, the process of compensating miners who mediate the dispute that arises during the trading process is not specified, so there is a lack of incentive to participate honestly in the dispute resolution process. Conversely, the process for punishing fraudulent miners is not specified, so responsibility for resolving disputes fairly is weak. A significant concern regardings the method put forth Li et al. is that the secret key k_D and the ciphertext E_i associated with the data are disclosed during the dispute process. Consequently, in the event of a dispute, other users can also gain access to k_D and E_i. This situation poses a risk whereby a fraudulent buyer colud deliberately instigate a dispute, thereby enabling other users to acquire the data without incurring the appropriate costs.

Alsharif & Nabil, (2020) proposed a fair data exchange technique based on CP-ABE primitive and zk-SNARKs. In this technique, data owners enforce flexible access control policies for encrypted data, and data requesters can use zk-SNARKs to verify the accuracy of the encrypted data without being provided with information about the data. In the technique proposed by Ahmad et al, a trust element called KDC (Key Distribution Center) is configured to generate keys used in CP-ABE. It is responsible for providing public/private keys to all entities and provides buyers with the properties and decryption keys that identify them. This trust factor KDC can be seen as a factor that hinders the decentralization of the proposed technique. Additionally, during the withdrawal stage, the secret value r is revealed to the blockchain. Because of this, there is a problem that even users who have not paid for the service can obtain data.

Li et al., (2020) proposed a blockchain-based secure and fair data trading system applying IB-ME (Identity-based Matchmaking Encryption) technique. This technique ensures bilateral authentication in which data trading occurs only when the policies required by both the data requester and the owner are simultaneously met. In the technique proposed by Park et al, the data requester's properties are specified using the IB-ME encryption algorithm, and the data owner uses the IB-ME decryption algorithm to specify the data requester's properties. Through this, data encrypted by the data owner can only be decrypted by the data requester, and if decryption is performed correctly, it is guaranteed that both the data owner and the requestor are valid parties specified in each other's policies. In the technique proposed by Park et al, there is an arbiter trusted by the entities participating in the trading process. The arbiter performs a dispute resolution role and also serves as IB-ME's key generator, generating encryption keys for each data owner and decryption keys for each data requestor. This arbiter can be seen as an element that hinders the decentralization of the blockchain and causes various problems such as single points of failure.

In contrast to prior studies, the scheme presented in this paper refrains from directly revealing the data encryption key to acquire data during the dispute process. Furthermore, the selection of arbitrator nodes involved in the trading process is conducted in an unpredictable manner, and a compensation fee is paid to the arbitrator nodes for fair trading participation. This motivates honest participation of blockchain nodes. Lastly, if an arbiter node behaves unfairly, it is encouraged to behave fairly by punishing it by confiscating its participation deposit.

Performance Analysis

In this section, we first look at the computational overhead of the proposed model, and then we analyze the on-chain storage overhead of existing researches and the proposed model. Finally, this section concludes by comparing the gas consumption of the proposed model with that of existing studies. The basic design principle of the proposed model is to minimize complex cryptographic operations in on-chain smart contracts as much as possible. Major cryptographic computations are performed off-chain by each participating entity, Table 3 shows the overhead of the mCL-ME technique in terms of major cryptographic operations. In Table 3, the execution time of each basic operation is the result of an implementation test assuming cryptographic parameters of 128-bit security. For details, see (Chen et al., 2021). In the DO, the highest computational overhead is the $mCLME_{Enc}$ operation, which takes approximately 1.32sec, and in the case of the DR, the $mCLME_{Dec}$ operation takes approximately 1.26sec. This computational overhead can be greatly reduced by using a more optimized implementation (Jiang et al., 2021).

Table 3: Main Cryptographic Operations of mCL-ME Primitives

	Operations			**Times ($\boldsymbol{ms}$)**		
	$\boldsymbol{DO}$	$\boldsymbol{DR}$	$\boldsymbol{BN}$	$\boldsymbol{DO}$	$\boldsymbol{DR}$	$\boldsymbol{BN}$
$\boldsymbol{SKGen}$	SM	-	$1SM + 1H$	48.674	-	323.053
$\boldsymbol{RKGen}$	-	SM	$2SM + 2H$	-	48.674	646.106
$\boldsymbol{Enc}$	$4Add + 2pairing + 3H + 4SM$	-	-	1,328.415	-	-
$\boldsymbol{Dec}$	-	$2SM + Add + 4pairing + 2Mul + 2H$	-	-	1,265.281	-
$\boldsymbol{Verify}$	-	-	$2pairing + 2H + Mul$	-	-	858.199

- SM is a standard scalar multiplication operation in $\mathbb{G}$, and the execution speed of the operation is $48.674ms$.
- Add is an addition operation in $\mathbb{G}$, and the execution speed of the operation is $0.293ms$.
- $Pairing$ is a bilinear pairing operation ($e: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$), and the execution speed of the operation is $154.705ms$.
- Mul is a multiplication operation in $\mathbb{G}_T$, and the execution speed of the operation is $0.031ms$.
- The operation execution speed of H is $274.379ms$.

Next, we examine the storage overhead on-chain of both existing researches and the proposed models. To concretely delve into the on-chain storage complexity associated with the proposed method, assuming a cryptographic technique with 128-bit security, it is possible to select the key lengths of a 128-bit symmetric cipher, a 256-bit hash function, and a 256-bit elliptic curve pairing. In this context, both pk and sk can be interpreted as having a length of 256-bit long, respectively, and when applying a BLS signature (Boneh et al., 2001), the signature length can be viewed as 256-bit long. In the case of the proposed technique, the on-chain storage overhead when including the dispute resolution process is approximately 860 bytes. Ahmad et al.'s technique, which uses cryptographic techniques such as CP-ABE and zk-SNARKs, is approximately 10,688 bytes, assuming 128-bit security and about 10 attribute information. Li et al.'s techniques that use the PCE cryptographic technique has a storage overhead of approximately 392 bytes. Lastly, the technique proposed by Park et al. uses the IB-ME cryptographic technique and has a storage overhead of approximately 627 bytes. Assuming that the Ethereum blockchain platform is applied, we compare the proposed model with existing researches regarding the gas cost per opcode of basic operations supported by EVM (Ethereum Virtual Machine). The gas cost required for basic arithmetic operations such as Add, comparison operations such as LT, and modular operations such as MOD is not large, at 3 to 8. EXP is approximately 30, so gas costs are somewhat high (Wood, 2014). The most important operation from a gas consumption perspective is the STORE operation, which requires 20,000 gas per word (Wood, 2014). Therefore, comparing gas usage from the perspective of storage, which requires the most cost, Ahmad et al.'s technique uses ≈680,000 gas, Li et al.'s technique uses ≈260,000 gas, and Park et al.'s technique uses ≈400,000 gas. The proposed technique requires ≈540,000 gas which is moderate. In terms of ensuring decentralized and fair data trading without the existence of fixed trust entities and ensuring security during the dispute resolution process, the on-chain storage overhead of the proposed scheme has an appropriate overhead compared to existing techniques. Table 4 shows the comparison results for major storage overheads on-chain and Figure 3 compares the gas costs on-chain with the existing techniques.

Table 4: The Overhead Comparison between Existing Schemes and the Proposed Model

	Ahmad et al	**Li et al**	**Park et al**	**Proposed model**
On-chain storage overhead and gas consumption (excluding metadata, including dispute process)	1,068 bytes $(\approx 680{,}000 gas)$	392 bytes $(\approx 260{,}000 gas)$	627 bytes $(\approx 400{,}000 gas)$	860 bytes $(\approx 540{,}000 gas)$
Major computation overhead on-chain	Requires Signature verification operation.	Requires PCE_{chk} operation and signature verification operation.	The operations conducted are limited to comparison, assignment, storage, addition and subtraction.	The operations conducted are limited to comparison, assignment, storage, addition and subtraction.

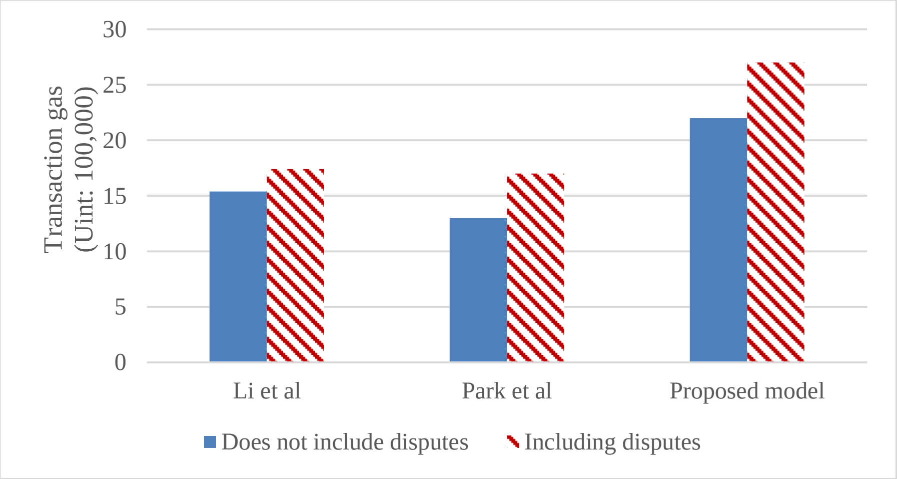


Figure 3: Comparison of Gas Consumption with Existing Schemes

5 Conclusion

In this paper, we proposed a decentralized and fair data trading scheme that supports bilateral access rights management in a blockchain environment. In the proposed scheme, unlike existing researches, instead of using a fixed trust factor such as TTP, a randomly selected blockchain node is used as an arbiter node during each data trading process. Additionally, a modified CL-ME technique and AONT mechanism were applied to provide data confidentiality and fairness. In addition, bilateral access rights management is supported to secure access rights management on data. The proposed model was designed in a hybrid way that combines on-chain and off-chain to minimize the storage space and performance limitations of the blockchain. In terms of on-chain storage overhead and gas consumption of smart contracts, the performance of the proposed method has a moderate overhead compared to the existing methods, but the proposed model offers a higher usability due to advantages such as ensuring bilateral access control and eliminating fixed trust factors.

Ethical Declaration

Conflict of interest: No declaration required.

Financing: This work was supported by a Research Grant of Pukyong National University (2023).

Peer review: Double anonymous peer review.

References

[1] Abbadini, M., Beretta, M., Di Vimercati, S. D. C., Facchinetti, D., Foresti, S., Oldani, G., ... & Samarati, P. (2024). Supporting Data Owner Control in IPFS Networks. In *Proceeding of the IEEE International Conference on Communications (IEEE ICC 2024)*.

[2] Abe, M., & Ohkubo, M. (2012). A framework for universally composable non-committing blind signatures. *International Journal of Applied Cryptography*, *2*(3), 229-249. https://doi.org/10.1504/IJACT.2012.045581

[3] Al-Kuwari, S., Davenport, J. H., & Bradford, R. J. (2011). Cryptographic hash functions: Recent design trends and security notions. *Cryptology ePrint Archive*.

[4] Alsharif, A., & Nabil, M. (2020, December). A blockchain-based medical data marketplace with trustless fair exchange and access control. In *GLOBECOM 2020-2020 IEEE Global Communications Conference* (pp. 1-6). IEEE. https://doi.org/10.1109/GLOBECOM42002.2020.9348192

[5] Asokan, N., Schunter, M., & Waidner, M. (1997, April). Optimistic protocols for fair exchange. In *Proceedings of the 4th ACM Conference on Computer and Communications Security* (pp. 7-17).

[6] Ateniese, G., Francati, D., Nunez, D., & Venturi, D. (2021). Match me if you can: matchmaking encryption and its applications. *Journal of Cryptology*, *34*, 1-50. https://doi.org/10.1007/s00145-021-09381-4

[7] Bajoudah, S., Dong, C., & Missier, P. (2019, July). Toward a decentralized, trust-less marketplace for brokered IoT data trading using blockchain. In *2019 IEEE international conference on blockchain (Blockchain)* (pp. 339-346). IEEE. https://doi.org/10.1109/Blockchain.2019.00053

[8] Bellare, M., Desai, A., Jokipii, E., & Rogaway, P. (1997, October). A concrete security treatment of symmetric encryption. In *Proceedings 38th Annual Symposium on Foundations of Computer Science* (pp. 394-403). IEEE. https://doi.org/10.1109/SFCS.1997.646128

[9] Boneh, D., Lynn, B., & Shacham, H. (2001, November). Short signatures from the Weil pairing. In *International conference on the theory and application of cryptology and information security* (pp. 514-532). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45682-1_30

[10] Chen, B., Xiang, T., Ma, M., He, D., & Liao, X. (2021). CL-ME: Efficient certificateless matchmaking encryption for Internet of Things. *IEEE Internet of Things Journal*, *8*(19), 15010-15023. https://doi.org/10.1109/JIOT.2021.3073008

[11] Chen, J., & Micali, S. (2016). Algorand. https://doi.org/10.48550/arXiv.1607.01341

[12] Dai, W., Dai, C., Choo, K. K. R., Cui, C., Zou, D., & Jin, H. (2019). SDTE: A secure blockchain-based data trading ecosystem. *IEEE Transactions on Information Forensics and Security*, *15*, 725-737. https://doi.org/10.1109/TIFS.2019.2928256

[13] Goldwasser, S., & Micali, S. (1984). Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2), 270-299. https://doi.org/10.1016/0022-0000(84)90070-9

[14] Goldwasser, S., Micali, S., & Rivest, R. L. (1988). A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on computing*, *17*(2), 281-308. https://doi.org/10.1137/0217017

[15] He, S., Lu, Y., Tang, Q., Wang, G., & Wu, C. Q. (2021). Fair peer-to-peer content delivery via blockchain. In *Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part I 26* (pp. 348-369). Springer International Publishing. https://doi.org/10.1007/978-3-030-88418-5_17

[16] Jiang, Y., Shen, X., & Zheng, S. (2021). An effective data sharing scheme based on blockchain in vehicular social networks. *Electronics*, *10*(2), 114. https://doi.org/10.3390/electronics10020114

[17] Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016, May). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE symposium on security and privacy (SP)* (pp. 839-858). IEEE. https://doi.org/10.1109/SP.2016.55

[18] Li, Y. N., Feng, X., Xie, J., Feng, H., Guan, Z., & Wu, Q. (2020). A decentralized and secure blockchain platform for open fair data trading. *Concurrency and Computation: Practice and Experience*, *32*(7), e5578. https://doi.org/10.1002/cpe.5578

[19] Lindell, A. Y. (2009, April). Adaptively secure two-party computation with erasures. In *Cryptographers' Track at the RSA Conference* (pp. 117-132). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-00862-7_8

[20] Muralidharan, J. (2024). Machine Learning Techniques for Anomaly Detection in Smart IoT Sensor Networks. *Journal of Wireless Sensor Networks and IoT*, *1*(1), 10-14. https://doi.org/10.31838/WSNIOT/01.01.03

[21] Naor, M., & Yung, M. (1990, April). Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing* (pp. 427-437).

[22] Park, Y., Jeon, M. H., & Shin, S. U. (2023). Blockchain-based secure and fair iot data trading system with bilateral authorization. *Computers, Materials & Continua*, *76*(2), 1871-1890. http://dx.doi.org/10.32604/cmc.2023.039462

[23] Wei, X., Yan, Y., Guo, S., Qiu, X., & Qi, F. (2021). Secure data sharing: Blockchain-enabled data access control framework for IoT. *IEEE Internet of Things Journal*, *9*(11), 8143-8153. https://doi.org/10.1109/JIOT.2021.3111012

[24] Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, *151*(2014), 1-32. https://ethereum.github.io/yellowpaper/paper.pdf

[25] Zhang, R., Hanaoka, G., & Imai, H. (2004). On the security of cryptosystems with all-or-nothing transform. In *Applied Cryptography and Network Security: Second International Conference, ACNS 2004, Yellow Mountain, China, June 8-11, 2004. Proceedings 2* (pp. 76-90). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-24852-1_6

[26] Ziga Kodric, S. V., & Jelovcan, L. (2021). Securing edge-enabled smart healthcare systems with blockchain: A systematic literature review. *technology*, *1*, 48. https://doi.org/10.22667/JISIS.2021.11.30.019

Authors Biography

Su Jin Shin, received her B.E. degree in Dept. of Software and Artificial Intelligence from Pukyong University, Republic of Korea in 2023. She is currently a master course student in Department of Information Security, Graduate School from Pukyong National University. Her research interests are related with blockchain security, applied cryptography, and vehicle security.

Youngho Park, received his M.S. degree in computer science and Ph.D. degree in information security from Pukyong National University, Republic of Korea in 2002 and 2006, respectively. He is currently a researcher of Electronics and Information Communications Research Center, Pukyong National University. His research interests are related with applied cryptography and its applications, communication security, and blockchain.

Sang Uk Shin, received his M.S. and Ph.D. degrees from Pukyong National University, Busan, Korea in 1997 and 2000, respectively. He worked as a senior researcher in Electronics and Telecommunications Research Institute, Daejeon Korea from 2000 to 2003. He is currently a professor in Division of Computer Engineering and Artificial Intelligence, College of Information Technology and Convergence, Pukyong National University. His research interests include cryptographic protocol, blockchain security, mobile and wireless network security, and digital forensics.