Blockchain-based fair and secure protocol for decentralized data trading

Mi Hyeon Jeon[1] and Sang Uk Shin[2*]

[1]Dept. of Information Security, Graduate school, Pukyong National University,
45, Yongso-ro, Nam-Gu, Busan, Republic of Korea
jmh3850@pukyong.ac.kr

[2]Div. of Computer Engineering, College of Information Technology and Convergence,
Pukyong National University, 45, Yongso-ro, Nam-Gu, Busan, Republic of Korea
shinsu@pknu.ac.kr

Received: Jun 30, 2022; Accepted: August 15, 2022; Published: August 31, 2022

Abstract

Demand for data transactions has increased in various fields with the development of computing technology. Several trading platforms have emerged to cater to this demand; however, most platforms were designed with a centralized model, which has led to performance, security, and privacy concerns. In addition, an increasing number of businesses are using only private platforms for trading to maintain trade secrets. However, these platforms present security and fairness challenges, e.g., denial of their actions by transaction participants and reliance on the trust of trusted third parties (TTPs). To solve these problems, several studies have used blockchain to construct secure data trading platforms. However, in most cases, decentralization is still not satisfied, such as TTP intervention in the initial setting process of the protocol. This paper proposes a secure and fair data trading protocol using blockchain technology. The proposed protocol eliminates the intervention of TTP by using a randomized arbitrator node selection method for dispute resolution. Furthermore, all-or-nothing transform (AONT) and dual-receiver cryptosystem (DRC) techniques are applied to ensure the requirements for a fair trading platform such as fairness, timeliness and data confidentiality.

Keywords: Blockchain, non-repudiation, data trading, fairness, decentralization

1 Introduction

Several computing technologies such as the Internet of Things (IoT), blockchain, cloud, artificial intelligence (AI), and their combinations, have been proposed and developed for various applications. For example, the demand for data transactions collected from IoT devices has increased; therefore, IoT terminals have been deployed in various industries [1]. According to IDC's prediction, 38.6 billion devices will be connected to the Internet up to 2025 [2]. The vast amount of data generated from these IoT devices is becoming increasingly important in the era of big data, and data ownership is also becoming a source of interest. Therefore, a fair data sharing/trading platform helpful for both data owners (generators) and data users (consumers) has a practical value in a data-driven society. However, existing data sharing/trading solutions are mainly traded privately between data owners and users or on centralized

Journal of Internet Services and Information Security (JISIS), volume: 12, number: 3 (August), pp. 30-48
DOI:*10.22667/JISIS.2022.08.31.030*

*Corresponding author: Div. of Computer Engineering, College of Information Technology and Convergence, Pukyong National University, Busan, Republic of Korea, Tel: +82 51-629-6249, Web: https://cms.pknu.ac.kr/lacuc/

platforms, which has many drawbacks and problems: Major processes, including account management of participants on the platform and data exchange matchmaking, are performed through centralized agencies [3]. The problems of the centralized models, such as performance, security, and privacy concerns, still exist. As there is no public (open) transaction platform, several interactions are required to complete transactions between the two parties and several negotiations are necessary to reach an agreement. In addition, because private transactions are inherently insecure, the security and reliability of transactions cannot be guaranteed. The centralized platform essentially has a problem with trust. It is of great concern that the security of trading data and the fairness of transactions depend entirely on the untrustworthy platform. Therefore, research on a fair decentralized trading platform is receiving much attention.

Blockchain is in the spotlight as a technology for decentralization, and studies on applying blockchain to data transaction platforms have been conducted recently [4, 5, 6]. Blockchain is a distributed ledger technology in which data blocks take the form of a chain based on a peer-to-peer (P2P) method. Blockchain can provide immutability in a data transaction environment because it makes it impossible to manipulate content posted on the blockchain.

This paper proposes a distributed data transaction protocol that provides a fair non-repudiation function based on blockchain technology. The main contributions of the paper are as follows:

- It solves the problems of the centralized model by providing fair non-repudiation without trust third party (TTP) in a decentralized environment. Unlike conventional TTP-based platforms, the proposed model uses randomly selected blockchain nodes to arbitrate disputes during the transaction process. The arbitrator node is selected through Algorand's unpredictable leader selection technique [7]. Due to the property of not predicting which node will be chosen as the arbitrator node before the execution of the transaction protocol, the security and decentralization of the proposed scheme is guaranteed.

- In addition, the proposed protocol satisfies fairness, confidentiality and timeliness and is secured against collusion attacks. To satisfy the basic requirements for secure data trading and realize the practical security, the proposed scheme uses some cryptographic primitives such as dual receiver cryptosystem (DRC) technique, the digital signature, the cryptographic hash function and the symmetric key encryption.

- For security and efficiency, the trading data is transformed into the $(Stub||Package)$ pair by using all-or-nothing transform (AONT) before outsourcing to an external storage. And then the Package part is oursourced to untrusted open storage such as the cloud storage and the Stub part encrypted by a symmetric key is sent to on-chain smart contract. It reduces the on-chain trsnamission cost and improves the efficiency.

Thus, the proposed technique provides non-repudiation on a fully distributed sharing platform with no involvement of TTP. In addition, by offloading complex cryptographic operations to a blockchain node that acts as an arbitrator, it reduces gas consumption in on-chain smart contracts. Therefore, compared to existing fair trading schemes in a decentralized environment, the proposed scheme satisfies decentralization while being more efficient in computational and storage overhead in on-chain smart contracts.

2 Preliminaries

2.1 Arbitrator node selection technique

Chen et al. [7] proposed an unpredictable leader-selection technique using the random beacon protocol, for which nodes in the blockchain network become arbitrator nodes. In this study, the arbitrator node

acts as an arbitrator in the event of a denial dispute. Algorand's unpredictable leader-selection technique derives a new random beacon value for the next transaction by using a random beacon value in the previous round as an input value. Given the last random beacon value Q_{DO}^{r-1}, a new random beacon value Q_{DO}^{r} can be derived by two methods. First, the selected leader node discloses the arbitrator's credential and signature $SIG_{l_{DO}^{r}}(Q_{DO}^{r-1})$ of the previous random value. Subsequently, the random beacon value of the next round can be derived by obtaining a hash value along with the $r-1$ value as (Equation (1)).

$$Q_{DO}^{r} = H(SIG_{l_{DO}^{r}}(Q_{DO}^{r-1}), r-1) \tag{1}$$

If the first method fails, the random beacon value may be derived using Equation 2. (Q_{DO}^{r} is the random beacon value in the r-th transaction of the data owner(DO)).

$$Q_{DO}^{r} = H(Q_{DO}^{r-1}, r-1) \tag{2}$$

Whether node i plays the role of the arbitrator in the DO's r-th transaction depends on the following condition: $.H(SIG_i(r, Q_{DO}^{r-1})) \leq p$, where .H(x) represents the hash value of x expressed in 256-bit binary digits in the [0,1] interval, and p is the predefined probability that the participant will be chosen as the arbitrator.

Here, other participants must not know which node is the arbitrator in a particular transaction. Participant i alone must be able to calculate $SIG_i(\cdot)$, and thus determine whether it belongs to the arbitrator set. Additionally, participant i alone can declare its identity as an arbitrator by disclosing its own arbitrator credential $SIG_i(r, Q_{DO}^{r-1})$. When several potential arbitrators exist, the node with the smallest value of $.H(SIG_i(r, Q_{DO}^{r-1}))$ is the arbitrator for the transaction.

2.2 Encryption algorithms

A) DRC (dual receiver cryptosystem) [8]: This is a cryptosystem that enables a ciphertext to be decrypted by two independent receivers. It is a combined public key system composed of an encryption scheme secure against chosen-ciphertext attacks and a signature scheme secure against adaptive chosen message attacks. Both signatures and ciphertexts that use the same secret key consist of a single cryptosystem. Furthermore, a sender can escrow her message to a party of the sender's choosing without this party being involved in any pre-processing. This achieves full flexibility in key management and recovery policy. A DRC scheme consists of four random polynomial algorithms: The key generation $(PK, SK) = DRC_{KGen}(\lambda)$ is a randomized algorithm that takes a security parameter λ as input, and produces a pair (PK, SK) of corresponding public key and private key. The encryption algorithm $c = DRC_{Enc}(m, PK_1, PK_2)$ is a randomized algorithm that takes a message m and public keys (PK_1, PK_2) of two entities as inputs, and generates a ciphertext c. The decryption algorithm $m = DRC_{Dec}(c, SK_1, PK_2)$ is a randomized algorithm that takes a private key SK_1, a public key PK_2, and a ciphertext c as inputs, and produces a message m or a special reject symbol. The recovery algorithm $m = DRC_{Rec}(c, PK_1, SK_2)$ is a randomized algorithm that takes a public key PK_1, a private key SK_2, and a ciphertext c as inputs, and produces a message m or a special reject symbol.

B) AONT (all-or-nothing transform) [9]: This is an encryption mode for block ciphers that prevents a plaintext block from being obtained when a random search attacker decrypts one cipher block through proper pre-processing before encrypting a message. If a single block of ciphertext can be decrypted, the key search problem can be made relatively easy because it is sufficient to test candidate keys held by the malicious participant. This feature is called separability, and the AONT adopts non-separation characteristics to make the brute force search attack difficult. AONT guarantees the property that no single plaintext block is obtained unless all ciphertext blocks are decrypted.

- Transform the message sequence $m_1, m_2, ..., m_s$ into a "pseudo-message" sequence $m'_1, m'_2, ..., m'_{s'}$ (for some $s' \geq s$) with an "all-or-noting transform", and
- Encrypt the pseudo-message with an ordinary encryption mode with the given cryptographic key K to obtain the ciphertext sequence $c_1, c_2, ..., c_t$.

We call the encryption modes of this type " all-or-nothing" encryption modes. The AONT must have specific properties to execute this encryption work.

(Definition) A transform f mapping a message sequence $m_1, m_2, ..., m_s$ into a pseudo-message sequence $m'_1, m'_2, ..., m'_{s'}$, is said to be an AONT if

- The transformation f is reversible: the original message sequence can be obtained from the pseudo-message sequence.
- Both the transformation f and its inverse are efficiently computable, i.e., computable in polynomial time).
- Computing any function of any message block is infeasible if any one of the pseudo-message blocks is unknown.

2.3 Related works

Several studies have proposed non-repudiation and data-transaction platforms based on blockchain. Ferrer et al. [10] proposed a protocol in which neither of the two participants is in an advantageous or disadvantageous position with blockchain technology using the Bitcoin platform. Huang et al. [11] proposed a contract-signing protocol that enables fair-contract signing between three parties based on blockchain. For the three-way contract to be valid, the contract of one participant must be signed by the other two participants as well. In addition, Zhang et al. [12] proposed a protocol that uses the Ethereum blockchain platform and verifiably encrypted signature (VES) scheme to perform fair-contract signatures between agreeing two parties. Li et al. [13] generated communication evidence in the form of digital certificates to prevent the denial of interactive services, and serve as nodes for blockchain networks to collect information generated during the communication process and post it to the blockchain network as non-repudiation evidence. Xu et al. [14] used blockchain technology to provide fair-network computing services in industrial IoT (IIoT) scenarios. In this service, when the IIoT client requests a service from the blockchain, the service provider can prevent the client from denying participation by providing the service by dividing the program into two parts. Wang et al. [15] designed a data-delivery model by referring to the gradual release mechanism (GRM) model, in which data to be exchanged between the two parties are divided into several packets and transmitted alternately. Through mutual control of the proposed model of [15], non-repudiation can be achieved without relying on complex encryption methods or third parties in the data-transfer process. Li et al. [16] proposed distributed trading solutions for public process data transactions using blockchain networks, smart contracts (SCs), and cloud services. Sellers can encrypt and upload data to be sold into storage and realize secure storage and efficient exchange, delivering keys used to decrypt data to buyers who wish to purchase. Ahmad et al. [17] proposed a protocol that allows sellers to establish flexible access control policies on the data they will sell, and buyers to verify the accuracy of encrypted records without disclosing information about records using the advanced zk-SNARKS protocol for purchased data, realizing fair data exchange by using proof-of-delivery.

3 Proposed model

This section introduces the model and protocol for fair and secure data trading based on blockchain. The assumption and threat model are first described, and then the specific protocol of the proposed model is

Table 1: Notations of the proposed model

Symbols	Details	Symbols	Details
ID_x	Participant x's ID	$Deposit_x$	Participant x's deposit
PK_x	Participant x's public key	SK_x	Participant x's secret key
TK	One-time data key	M	Data
$Package$	Long part of M after AONT	$Stub$	Short part of M after AONT
σ_x	x's signature	$H(x)$	x's hash value
I_{Stub}	Value encrypted using DRC	ptime	Period of time to participate as an arbitrator
Fee_{bn}	Fees paid to arbitrator nodes	$Price_M$	The price of M
$Addr_{DO-M}$	External storage address	$Sig_A(r,Q)$	Arbitrator credential
$T1$	Arbitrator node selection deadline	$T2$	Trade request deadline
$T3$	Transaction approval deadline	$T4$	Dispute resolution request deadline
$Keywords_M$	Keywords required when searching for data	$Terms_M$	Terms required when searching for data

described. Table 1 shows the symbols and notations used in this paper.

3.1 Assumptions and threat model

A) Assumptions

(a) Blockchain (including smart contracts) used in the system inherits the security properties of the underlying blockchain platform. That is, basic properties such as integrity and immutability of the underlying blockchain, are assumed.

(b) It assumes a static corruption that an attacker can corrupt some entities only before the protocol-execution process [16].

(c) The proposed model assumes a synchronous authenticated point-to-point channel to describe an attacker's ability to control communication (i.e., synchronous authenticated channel). That is, for any message delivered between honest entities, we assumed that the attacker could delay messages until a prior known delay value Δ, but cannot delete, path change, modify them. Without loss of generality, we consider a global clock in the system [18].

(d) When using the arbitrator node selection technique, we assumed that the number of nodes that wants to become arbitrator nodes is sufficiently large.

B) Threat model and security goals

As the transaction participants, the data owner (DO) and data requester (DR) do not trust each other, they can engage in malicious actions that deceive the other party for profit. In the case of the DO, they may not deliver actual data or may provide false/inaccurate data despite receiving a legitimate payment for data from the DR. The DR may also attempt to obtain data without paying the DO for the data [19]. Nodes selected as blockchain arbitrator nodes may also be dishonest or honest-but-curious. During dispute resolution, the blockchain node (BN) may not send evidence for dispute resolution to the smart contract (SC) or may attempt to send incorrect information.

We aim to satisfy the following requirements:

- On the condition that all participants in the transaction are honest, it must achieve the completeness in the synchronous authenticated network and the standalone model.

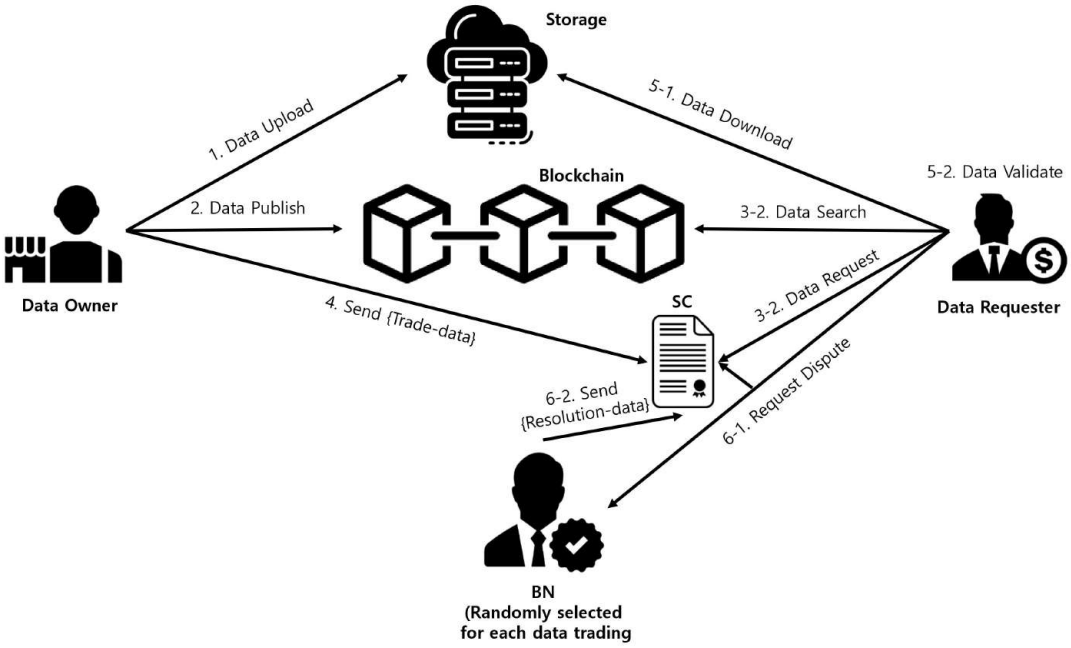

Figure 1: Overview of proposed model

- In the synchronous authenticated models and standalone settings, on the condition that the underlying cryptographic primitives are secure, it must satisfy fairness requirements even if either the DO or the DR is compromised by a non-adaptive PPT attacker.
- It must be secure against collusion attacks between attackers and BN.
- In synchronous authenticated networks and standalone models, it must satisfy confidentiality against non-adaptive PPT attackers.
- If either the DO or the DR is honest, it must satisfy timeliness in a synchronous authenticated network and standalone model.

3.2 Proposed model

Fig.1 illustrates the architecture and the components of the proposed model.

1. Components

   - DO (or seller): The DO is assumed to be a participant in the blockchain and P2P networks and possesses the relevant IDs and key pairs because it intends to trade its data to other entities through flexible access control policies. The DO wants fair exchange arbitration by promising a third party (i.e. arbitrator node) a fee of Fee_{bn} for smooth data transaction. He/she may also use a one-time key pair (PK_{DO}, SK_{DO}) for each data transaction.
   - DR (or buyer): The DR is assumed to be a participant in the blockchain and P2P networks and possesses the relevant ID and key pairs because it intends to purchase data from the DO. The DR also uses a one-time key pair (PK_{DR}, SK_{DR}) for each data transaction.
   - Blockchain/SC: Blockchain infrastructure that is secure and dependable and supports smart contracts is used. A P2P network is used to facilitate communication between participants.

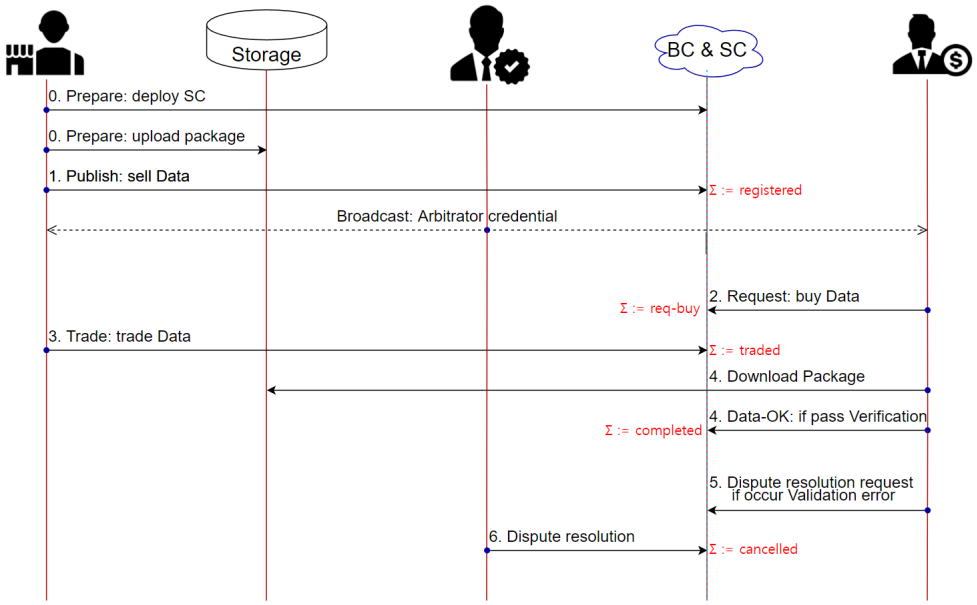

Figure 2: The overview of the proposed protocol

SCs are stateful functions that expose all internal states to participating entities (including attackers). An SC can transparently imitate contracts in real life by specifying certain invariant conditions in advance to transact as a "coin" on the cryptocurrency ledger. Actual contracts can be instantiated through blockchains such as Ethereum. Throughout this paper, the SC follows the pseudo-code notation provided in [18].

- BN: An entity participating in consensus in the blockchain infrastructure, and BN's selection depends on the blockchain infrastructure used. In the case of a public blockchain model, all participating entities can be BN. In the case of a private blockchain, only a few selected entities can participate as BN. BN's role is to perform dispute resolution by receiving arbitration fees for data transactions to support decentralized fair data trading protocols. If BN behaves unfairly, the deposit placed in advance to participate as an arbitrator node will be confiscated. Assuming that BN also has a key pair with its ID, BN participating as an arbitrator node uses a one-time key pair for each data transaction.
- Storage: Cloud storage or distributed storage such as IPFS may be used as the storage of the proposed model. Storage must reliably manage data and ensure availability. Storage is also not a trustworthy element, so data owners encrypt data to ensure confidentiality before outsourcing it to storage.

2. Cryptographic primitives
The proposed model uses the DRC and AONT scheme described in Section 2.2. We also consider the following cryptographic primitives:

   (a) A cryptographically secure hash function $H : \{0,1\}^* \rightarrow \{0,1\}^{\lambda}$. Here, λ is a security parameter.

(b) Semantically secure symmetric key encryption scheme consisting of (*SymKGen*, *SymEnc*, *SymDec*).

(c) Semantically secure asymmetric key encryption scheme for chosen ciphertext attacks consisting of (*ASymKGen*, *ASymEnc*, *ASymDec*).

(d) Secure digital signature algorithm with existential unforgeability under chosen message attack (EU-CMA) consisting of polynomial time algorithms (*SignKGen*, *Sign*, $Verify$).

3. Detailed service protocol

The proposed protocol is constructed by applying the AONT and DRC mechanisms described in Section 2.2 and proceeding to the Prepare, Publish, Request, Trade, Download & Validate, and Dispute resolution steps. The proposed scheme ensures fairness for each party, even if a non-adaptive PPT attacker compromises either the DO or DR. Fig.2 illustrates the overall process of the protocol. As shown in the figure, SCs are stateful ideal functionality with access to the blockchain for supporting fair data transactions and describe them according to the conventional pseudo-code notation of SCs [18, 20]. This notation reflects the essence of real-world SCs because Turing-complete SCs can be viewed as stateful programs that transparently deal with predefined functionalities, Conversely, SCs can access cryptocurrency blockchains that faithfully handle conditional payments according to their internal states. Unlike existing schemes, SCs of the proposed scheme do not perform complex and heavy operations. SCs only perform comparison, allocation, store, and addition and subtraction operations. Complex cryptographic computations are completed locally by participating entities. Fig.3 illustrates the role of SCs in the proposed model.

(0) Prepare phase

The DO first deploys the SCs. Then the DO converts the data M to be shared/transacted into a pair ($Stub||Package$) through AONT processing. Subsequently, a relatively short-length $Stub$ is encrypted with $eStub$ using a symmetric key encryption algorithm such as AES with a one-time data key TK: $eStub = SymEnc(TK, (Stub||H(SK_{DO}))$. The DO stores data ($Package$, σ_{pkg}) in an external storage system (cloud or IPFS). σ_{pkg} is the signature value for the Package: $\sigma_{pkg} = Sign(Package, SK_{DO})$. The key for signing the Package is a one-time key pair (PK_{DO}, SK_{DO}) used for trading data M.

(1) Publish phase

The DO interacts with the SC as:

- The DO sends the following {Sell-Data} including metadata for the data M to the SC:
  {Sell, ID_{DO}, $H(M)$, $H(Stub||Package)$, $keywords_M$, $Terms_M$, (r,Q), $T1$, $Price_M$,Fee_{bn}, $Deposit_{DO}$, PK_{DO}, $T2$}.
  Here, (r,Q) is a parameter value required when selecting the arbitrator node.
  Nodes that wish to participate as a arbitrator node in the data transaction process publish the transaction TX_{BN} containing their information $\{ID_A, ptime, PK_A, Deposit\}$ to the blockchain. After that, the node checks {Sell-Data} and broadcasts its own arbitrator credential $Sig_A(r,Q)$ within $T1$. The DO selects the node with the smallest $.H(Sign_A)$ among the arbitrator credentials broadcasted within $T1$ as the arbitrator of the data M.

(2) Request phase

This phase is executed by the DR in the assistance of the SC, which proceeds as:

- After retrieving for the desired data through the search for metadata posted on the blockchain, the DR sends the following {Buy-data} to the SC:
  {Buy, ID_{DR}, $H(M)$, $Price_M$, $Deposit_{DR}$, PK_{DR},$T3$}.
  The DR also uses a one-time key pair (PK_{DR}, SK_{DR}) used in data M transactions.

(3) Trade phase
When the DO receives the transaction {Buy-data}, the following process is executed:

- First, check whether the state Σ is 'req-buy' and the reception time is within timestamp $T2$.
- Encrypt the TK with the DR's public key: $eTK = AsymEnc(PK_{DR}, TK)$.
- Compute the signature σ_{Stub} for validation: $\sigma_{Stub} = Sign((Stub, TK), SK_{DO})$.
- Compute I_{Stub} using the DRC encryption algorithm with the DR's public key and the arbitrator node A's public key:
  I_{Stub} = $DRC((eTK, Addr_{DO-M}), PK_{DR}, PK_A)$.
- The DO sends the following {Trade-Data} to the SC:
  {Trade, ID_{DO}, ID_{DR}, $H(M)$, $eStub$, $H(eStub)$, σ_{Stub}, I_{Stub}, $T4$}

(4) Download & Validate phase
When the DR receives {Trade-Data} from a SC, it confirms:

- Check the reception time is within timestamp T3 and the state Σ is 'traded'.
- Check I_{Stub} in transactions posted on the blockchain.
- $(eTK, Addr_{DO-M})$ is recovered through a decryption algorithm of the DRC using its own private key and the arbitrator node A's public key: $(eTK, Addr_{DO-M})$ = DRC_{Dec} (I_{Stub}, SK_{DR}, PK_A).

And then the DR downloads ($Package'$, σ'_{pkg}) from the external storage $Addr_{DO-M}$ and verifies the signature σ'_{pkg}. Next, after decrypting $eStub$ using TK' to recover $(Stub'||K')$, signature σ_{Stub} contained in {Trade-Data} is verified (it verifies the validation of $TK', Stub'$). If the preceding verification passes, then $H(Stub'||Package)$ is verified. If it is determined to be valid, the data M' is recovered using AONT inverse transformation from $(Stub'||Package)$ pairs, and then the $H(M')$ is verified. If the validation of the recovered data M is successful, the DR sends {Data-OK, $H(M)$} to the SC.

(5) Dispute resolution phase
This phase is executed by the DR and the arbitrator node in assistance of the SC, which proceeds as:

- If DR has obtained invalid data, the dispute resolution mechanism allows it to prove that it decrypts data that does not match with the disclosed/received hash value or have invalid signature values. The DR then requests a dispute resolution from the SC and arbitrator node A within timestamp $T4$, if a problem arises in the validity of the recovered data M', by sending the following {Dispute-Data}:
  {Dispute, ID_{DO}, ID_{DR}, ID_A, $H(M)$, $err-info$, I_{SK}}.
  Here, I_{SK} is a value obtained by encrypting the DR's one-time private key SK_{DR}: $I_{SK} = DRC_{Enc}(SK_{DR}, PK_A, PK_{DO})$.
- When arbitrator node A receives {Dispute-Data}, it performs the following tasks:
  - Check whether the state Σ is 'dispute'.
  - After checking $err-info$, decrypt I_{SK} to obtain a private key SK_{DR} of the DR: $SK_{DR} = DRC_{Dec}(I_{Stub}, SK_A, PK_{DO})$. And then it verifies whether the restored SK_{DR} corresponds to the public key PK_{DR} of the DR posted on the blockchain, and whether the hash value of the data restored through this SK_{DR} matches that posted on the blockchain, and whether the signature is valid.
  - After checking I_{Stub} in the transaction posted on the blockchain, $(eTK, Addr_{DO-M})$ is restored through the DRC recovery algorithm: $(eTK, Addr_{DO-M}) = DRC_{Rec}(I_{Stub}, PK_{DR}, SK_A)$.

- After downloading ($Package$, σ_{pkg}) from $Addr'_{DO-M}$, verify the signature σ_{pkg}.
- After recovering TK'' by decrypting eTK using the restored private key SK_{DR} of DR, obtain $Stub''$ by decrypting $eStub$.
- Verify the signature σ_{Stub}.
- Verify the $H(Stub||Package)$ and hash values $H(M)$.
- If any of the above processes fails to verify, send {Resolution-Data} specifying the DO as a "cheater" to the SC. Otherwise, send {Resolution-Data} specifying the DR as a "cheater" to the SC.

When the transaction is completed securely or the dispute resolution process is terminated, the DO generates a new one-time key pair (PK'_{DO}, SK'_{DO}), and then converts the data M back into a pair of $(Stub||Package)$ through AONT conversion to re-outsourcing the storage.

3.3 Application of AONT

In the proposed scheme, the AONT is applied for the DO to pre-process data M. We assume that cryptographically secure hash functions, such as SHA-2 and SHA-3, supporting at least 256-bit output are used.

- AONT Encrypt:

  1. Split data M by block size of the hash function used: $M = \{m_1, ..., m_n\}$, K is the hashed value of the DO's private key SK_{DO}: $K = H(SK_{DO})$
  2. Select a random number r having a size of 256-bits.
  3. $c_i = m_i \oplus G(r||i)$, where $i = \{1, \cdots, n\}$, and the function G is a cryptographically secure pseudo-random function.
  4. $c_0 = r \oplus H(c_1, || \cdots || c_n)$
  5. $EncodedData = (c_o||c_1|| \cdots ||c_n)$. $EncodedData$ consists of $(Stub||Package)$ pairs, where the $Stub$ part has a relatively short length.
  6. In Encoded Data, the $Stub$ part is encrypted via a symmetric encryption such as AES using the secret key TK: $eStub = Enc(TK, (Stub||K))$.

- AONT Decrypt:

  1. Decrypt the $eStub$ part using the secret key TK: $(Stub||K) = Dec(TK, eStub)$
  2. Because it is $EncodedData$ = $(Stub||Package)$ = $(c_o||c_1||...||c_n)$, r can be calculated as: $r = c_0 \oplus H(c_1||...||c_n)$
  3. $m_i = c_i \oplus G(r||i)$, where $i = \{1, \cdots, n\}$.

4 Analysis

The proposed scheme satisfies all the security goals described above. First, Section 4.1 explains how the proposed model satisfies confidentiality, fairness, and timeliness, and supports second resale resistance, Section 4.2 compares the proposed scheme with existing schemes regarding the computation and storage overhead.

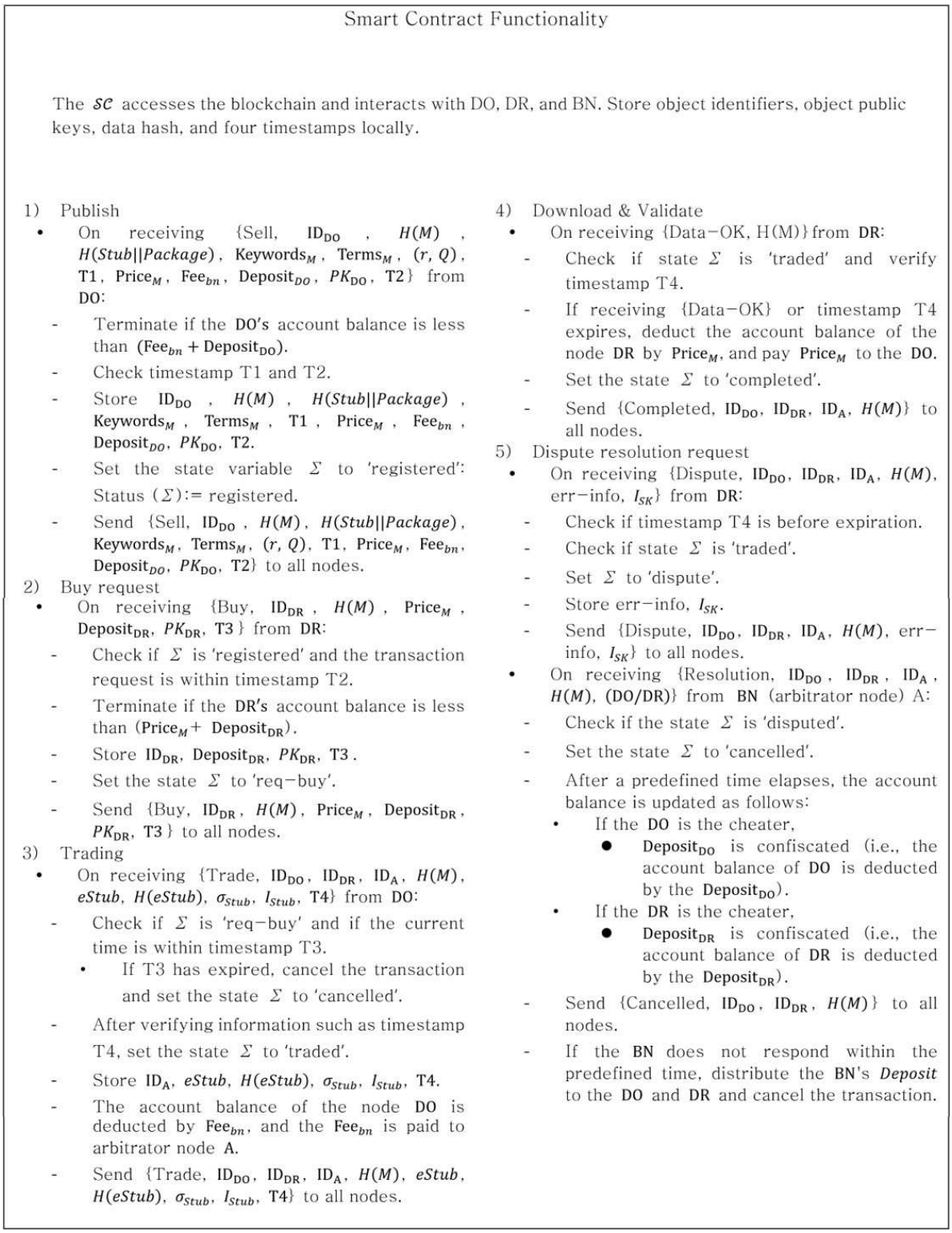

Figure 3: Smart contract functionality for the proposed protocol (here, "Sending to all nodes" implies that the smart contract is transparent to the public.)

4.1 Security analysis

[Lemma 1] On the condition that all participants in the transaction are honest, the proposed scheme meets the completeness in the synchronous authenticated network and the standalone model.
(*proof*) When both the DO and the DR participating in the transaction honestly follow the protocol, the DO obtains a net profit of $Price_M$ and the DR obtains valid data M. □

[Lemma 2] In the synchronous authenticated models and standalone settings, on the condition that the underlying cryptographic primitives are secure, the proposed protocol meets fairness requirements even if either the DO or the DR is corrupted by a non-adaptive PPT attacker.
(*proof*) Fairness requirements must satisfy both DR fairness and DO fairness:

a. DR fairness: Even if the DO behaves maliciously, the honest DR pays only for the data obtained actually and validly. This can be modeled as an attacker corrupting the DO who provides data to an honest DR. During the trade phase, an attacker may attempt to interfere with or manipulate the delivery of {Trade-Data}, in which case the SC cancels the transaction if {Trade-Data} does not arrive within timestamp $T3$. Therefore, if an attacker interferes with the delivery of {Trade-Data}, the transaction will not proceed normally; Thus an honest DR would neither obtain valid data nor pay for it. In addition, the attacker can manipulate {Trade-Data}, but the DR can verify the validity of the data obtained during the download & verification phase. If an attacker cannot (1) discover hash collisions, (2) forge signatures, or (3) manipulate SC execution on the blockchain, the fairness of DR is guaranteed. If the underlying hash function and the signature scheme are secure and additionally, the SC is modeled as an ideal function, the probability of breaking the fairness of DR is negligible. Therefore, the fairness of the DR is guaranteed against a malicious DO.

b. DO fairness: DO fairness means that an honest DO receives payment for valid data provided to the DR. This can be modeled as an attacker corrupting DR. An attacker can break the fairness of an honest DO by allowing the DR to obtain valid data without payment. In the data publishing step of the proposed scheme, the DO publishes transaction data to the blockchain by sending the {Sell-Data} message. In this step, only the metadata for data M are published; hence, it is not possible to obtain actual valid data. In the second step, the DR pays for the data M through {Buy-Data} before obtaining them in Step 3. Therefore, data M cannot be obtained without payment in the transaction process of the proposed scheme. To this end, an attacker may attempt to provide information obtained through {Trade-Data} in Step 3 to a DR who has not requested a transaction. In this case, $eStub$ and I_{Stub} may be obtained in {Trade-Data} of Step 3, but as I_{Stub} is encrypted by the DRC primitive through the application of the public key PK_{DR} of the DR requested in Step 2, eTK and $Addr$ cannot be obtained by decrypting it (eTK is also encrypted by PK_{DR}). As $eStub$ is encrypted by TK, TK cannot be obtained if eTK cannot be decrypted. Therefore, the DR who has not requested a transaction in Step 2 cannot obtain valid information from {Trade-Data}. For this, the underlying cryptographic mechanism must be breakable. That is, if an attacker cannot break the underlying public key encryption scheme, symmetric key encryption scheme, and the DRC scheme, the fairness of the DO is guaranteed. □

[Lemma 3] On the condition that the applied arbitrator node selection technique is secure, the proposed protocol is secure against collusion attacks between attackers and BN.
(*proof*) As described above, the basic assumption of the proposed model is that an attacker can corrupt the participating entity only before the protocol is executed. Therefore, for an attacker to perform the collusion attack with a BN, the suitable BN must be selected before the trading protocol is executed, and that BN must be selected as an arbitrator node in the trading protocol of the corresponding data. However, the attacker cannot predict which BN will be selected before the execution of the trading protocol, and

therefore the collusion in the transaction process depends on the security of Algorand's leader-selection technique used for the randomized BN selection. In other words, if Algorand's leader-selection technique is secure, the probability of an attacker's success in a collusion attack with BN is negligible in the proposed scheme. □

[Lemma 4] In synchronous authenticated networks and standalone models, the proposed scheme satisfies confidentiality against non-adaptive PPT attackers.
(*proof*) In this protocol, the original data M is transformed into a pair of $(Stub, Package)$ via AONT before being outsourced. The *Package* is outsourced to external storage, and the stored address encrypted by the DRC is delivered to the DR. The *Stub* portion is also encrypted as *eStub* by symmetric key ciphers with TK, and TK is encrypted as eTK by the paid DR's one-time public key PK_{DR}, which is re-encrypted through the DRC primitive and disclosed to the blockchain. Therefore, for a PPT attacker to compromise the confidentiality of data M, an attacker has to break (1) the DRC primitive, (2) the public key encryption scheme used for the eTK encryption, and (3) the symmetric key encryption scheme used for the *eStub* encryption. Therefore, if the underlying cryptographic mechanism is secure, the confidentiality of the proposed protocol against non-adaptive PPT attackers is guaranteed. □

[Lemma 5] If the DO or DR is honest, the proposed protocol satisfies timeliness in a synchronous authenticated network and standalone model.
(*proof*) Timeliness implies that honest participants can always reach a point in a protocol that can be terminated while ensuring fairness within a finite time. For the proposed protocol with an SC and at least one honest party, the termination cases are as follows:

a. No abort: If all parties are honest, the proposal protocol ends after {Data-OK} in phase 4 is received or $T4$ expires. At this time, the DO and DR are guaranteed to quit after achieving their purpose.

b. Cancel in the publish phase: In publishing transaction data, timestamp $T2$ specifies the deadline for transaction requests. If the transaction request is not received from the DR before $T2$ expires, the transaction will be canceled. The DO has not yet provided data M, and the DR has not requested a transaction; therefore, both the DO and DR are guaranteed to be fair.

c. Cancel in the trade phase: Timestamp $T4$ specifies the deadline for dispute resolution requests at the trade phase. The expiration of $T4$ implies that the DR did not raise a dispute request. Therefore, the SC terminates the protocol normally, and the fairness of the two parties is guaranteed.

d. Cancel in the dispute process: The dispute resolution phase is performed if the DR files a dispute resolution request within $T4$. If the BN does not deliver the {Resolution-Data} message at this phase within the specified time, the SC confiscates the BN's deposit, distributes it to the DO and the DR, and cancels the transaction. Therefore, fairness is guaranteed because the DO and DR are compensated for the cancellation of the transaction. □

[Theorem 6] If the underlying cryptographic primitive is secure and the security of the blockchain/SC is guaranteed, the proposed scheme satisfies completeness, fairness, confidentiality, and timeliness in the synchronous authenticated network and standalone model.
(*proof*) It is guaranteed by Lemmas 1, 2, 3, 4, and 5. □

Additionally, the proposed scheme supports the "second reselling attack" security. The second reselling is for malicious buyers to purchase data from the transaction blockchain and resell it to earn profits. In the proposed scheme, hash values of each data are posted on the blockchain. When the seller

Table 2: Comparison between existing schemes and the proposed model

	Ahmad et al. [17]	Li et al. [16]	Proposed model
Cryptographic primitive	CP-ABE, zk-SNARKs	PCE	AONT, DRC
Confidentiality	Guranteed in the normal trading process, but exists weakness in the dispute process	Guranteed in the normal trading process, but exists weakness in the dispute process	Guranteed
Fair exchange	Guranteed	Guranteed	Guranteed
Timeliness	Guranteed	Guranteed	Guranteed
Decentralization	Need KDC(TTP)	No fully trusted entities. Relying on the fairness of the miner which is not the fully trusted entitiy. No reward or punishment for miners.	No fully trusted entities. An unpredictably selected blockchain node before the start of the protocol as an arbitrator. There is reward and punishment to the arbitrator node for fair transaction support.

posts transaction data on the public phase, the blockchain/SC checks whether the hash value of the trading data collides. If a collision (i.e., a duplicate) is found, it can invalidate the transaction after checking that the seller in the current transaction is a buyer in the previous transaction containing duplicate data (i.e., not the owner of the original data). Therefore, the proposed protocol can prevent the second resale within the platform. However, there is a limitation that it cannot prevent attackers from reselling data personally in other ways or on other platforms.

4.2 Comparative analysis

This section analyzes the proposed scheme regarding computation and storage overheads and compares it with existing methods that support fair exchange based on blockchain.

He et al. [20] presented a fair delivery technique in which a third-party deliverer participated as intermediate entities, which is a slightly different model of two-party fair exchange. He et al.'s protocol applies the verifiable public key encryption (VPKE) scheme as a major cryptographic primitive. For comparison with the proposed protocol, He et al. method can be viewed as a fair-exchange technique from the perspective of sellers and buyers excluding the deliverer. In this respect, only simple operations other than the hash computation were used in the case of a normal transaction without dispute. However, in a dispute, the SC must perform the $VerifyPKE$ operation of VPKE with the relatively high computational complexity over on-chain to resolve this.

Ahmad et al. [17] proposed the blockchain-based fair exchange technique using the ciphertext-policy attribute-based encryption (CP-ABE) technique and zk-SNARKs scheme. In this scheme the introduction of a trusted entity called the key distribution center (KDC) was required to generate a key for CP-ABE, thus hindering decentralization. The weakness of this protocol is that the secret value γ is disclosed on the blockchain in the withdraw phase. This causes a problem where users who have not paid for the information disclosed in the transaction process can obtain data. With the application of the CP-ABE technique, another user satisfying the access policy acquires the $Addr$ and ciphertext (CT) disclosed on the blockchain and acquired data without paying the data fee. Thereafter, an illegal buyer obtains data C^* during the off-chain fair exchange process and aborts the protocol. The seller does not deliver the secret value of γ in the transaction because the protocol is interrupted. However, the fraudulent buyer can obtain data using the secret γ disclosed in the withdraw phase. Preventing this requires additional computational overhead for the seller to re-compute C^* using the new secret value γ for each exchange.

Li et al. [16] proposed another type of blockchain-based fair data trading platform, which applied a plaintext checkable encryption (PCE) primitive. This technique relies on miners for fair exchange. In the blockchain, miners are not fully trusted, but this scheme does not consider the case that miners who act as arbitrators for resolving disputes in transactions engage in fraudulent acts. This scheme assumed that miners execute the SCs using the evidence submitted and verifying them. Further, it does not specify the

reward for miners who arbitrated fair transactions, so miners have inadequate incentives to participate in fair arbitration of the transaction. In addition, as the penalty for illegal miners is not specified, the responsibility to resolve the dispute fairly is also weak. Another major problem is the disclosure of the secret key k_D and the ciphertext of data E_i during disputes. In the event of a dispute, other users can also access k_D and E_i. If an illegal buyer intentionally executes the dispute process, other users can obtain data without paying.

Unlike the two studies above, the proposed scheme does not directly expose data encryption keys for data acquisition during a dispute. As the arbitrator node required to participate in the transaction process is randomly selected and the information needed to obtain data is encrypted and disclosed by the DRC primitive by applying the public key of the arbitrator node, other users cannot decrypt it. Therefore, data cannot be obtained from public information during a dispute. It also motivates blockchain nodes to participate by paying a reward for fair arbitration of the transaction and induces them to act fairly by confiscating participation deposits as punishment if they behave unfairly. Table 2 compares the existing techniques (Ahmad et al. [17] and Li et al. [16]) with the proposed techniques.

The proposed scheme is compared with two existing schemes regarding their computation and storage complexity below. First, considering the on-chain storage complexity of the proposed technique, the information stored at each phase of the trading process is as follows:

- Publish: $(ID_{DO}, H(M), H(Stub||Package), Keywords_M, Terms_M, T1, Price_M, Fee_{bn}, Deposit_{DO}, PK_{DO}, T2)$
- Buy-Request: $(ID_{DR}, Deposit_{DR}, PK_{DR}, T3)$
- Trade:$(ID_A, eStub, H(eStub), \sigma_{Stub}, I_{Stub}, T4)$
- Dispute: $(err-info, I_{SK})$

Assuming the key length of a cryptographic primitive with 128-bit security, we can select a 128-bit symmetric cipher, 256-bit hash function, and 256-bit elliptic curve pairing. In this case, a public key (PK) and a private key (SK) may be viewed as having a length of 256 - bits each. A signature size may be viewed as having a length of 512-bits for elliptic curve digital signature algorithm (ECDSA) and having a length of 256 bits for BLS signatures [21]. In addition, the asymmetric encryption that takes the 128-bits symmetric key as a plaintext input has an output ciphertext of 512-bit length. The DRC encryption that takes 256-bits plaintext as an input has an output of 1,024-bits. Assuming the ID is 160-bit length, the price and timestamp are 64-bit lengths, and $Addr$ is 128-bit length, the storage complexity in the publish phase is (1,248-bits + metadata length). Here, the metadata is ($Keywords_M$, $Terms_M$). The storage overhead of the buy request phase is 544-bits, and the trade phase is 2,592-bits long. The storage bit length of the dispute resolution phase is (1,024-bits + $err-info$ length). Therefore, except for metadata and err-info, approximately 1 Kbytes or less (5,408-bits length) of storage is required when proceeding to the dispute resolution phase, indicating that the storage overhead on the on-chain is not as high as expected.

Unlike existing techniques, the basic design principle of the proposed scheme is to reduce complex operations in SCs to the maximum possible extent. Although SCs on blockchain platforms such as Ethereum support Turing completeness, processing cryptographic operations with high computational complexity, such as zero-knowledge proof in SCs is still a heavy burden regarding processing time and cost. Therefore, the SC of the proposed protocol is designed to perform only basic operations such as comparison, allocation, storage, and addition and subtraction. Complex cryptographic operations are performed locally by participating entities.

Specifically, analyzing the overhead of Ahmad et al.'s scheme, since cryptographic primitives such as CP-ABE and zk-SNARKS are used, additional storage parameters of attribute keys and proof are added,

Table 3: The overhead comparison between existing schemes and the proposed model

	Ahmad et al. [17]	Li et al. [16]	Proposed model
Storage overhead on on-chain SC (including dispute process, excluding metadata)	8,544-bit ($\approx$ 34 words$^{(\ddagger)}$) ($\approx$ 680,000 gas)	3,136-bit ($\approx$ 13 words) ($\approx$ 260,000 gas)	5,408-bit ($\approx$ 22 words) ($\approx$ 440,000 gas)
Expensive computational overhead on on-chain SC	Need signature verification operations	Need PCE_{chk} operations and signature verfication operations	No cryptographically expensive computation. Need only basic operations such as comparison, allocation, storage, and addition, and subtraction
Gas consumption required on the blockchain	19×10^5 gas	170×10^5 gas	53×10^5 gas

($^{(\ddagger)}$ Ethereum EVM uses 256-bit-wide words)

respectively. Therefore, the storage information required on the on-chain is (ID, metadata, ID_M, (H(M), Addr, AccessPolicy, CT, proof, fee), and the number of attributes is assumed to be 10, and 8,544-bit is required, excluding metadata and access policy. Furthermore, assuming 128-bit security, when the Gate count is 2^{10} at zk-SNARKS, the size of the key pair of the proof key (pk) and the verification key (vk) is 311.4 KBytes in pk and 3.6 KBytes in vk [22]. [17] argued that about 19×10^5 gas is consumed through three steps, but considering the key pair required for zk-SNARKS, more gas costs are required.

In Li et al.'s scheme, information that needs to be stored in the on-chain storage is (metadata, H_{root}, pk_{sign}, T_1, T_2, T_3, price, pk_{pce}, CT_{pce}, k_d, E_i, σ). Approximately 2,816-bit storage complexity is required assuming 128-bit security, except for metadata. Including information such as IDs which is not specifically specified in [16], it is calculated as about 3,136-bit. In addition, [16]'s scheme applied PCE cryptographic primitives for a fair data trading platform. Thus, in the event of a dispute, smart contracts should perform computational complexity PCE_{chk} operations and signature verification operations on-chain. The complexity required for PCE_{chk} requires four pairing operations and 1 div operation, and has a bit length of 256*3. Li et al. estimated that the gas consumption suggested by the proposed method is about 170×10^5 Gas. Table 3 shows the comparison results between the existing methods and proposed models in terms of on-chain storage overhead and major computation overhead.

The proposed model was implemented on Ethereum blackchain to measure gas usage. We experimented with Windows 10 x64, Intel® Core™ i7-10700 CPU 2.9GHZ, and 16 GB RAM computer environments, and the Ethereum blockchain environment was run from the Java Script VM. The smart contract writing language used version 0.8.7 as the solidity language. Smart contracts were created with Remix IDE in the online environment. Since Remix IDE contained its blockchains, such as Solidity compiler and Ganache, it did not use separate debugging tools and compiler, and was immediately available for distribution on TestNet. The proposed model consists of five stages except for the prepare stage, but the smart contract is a total of six because the dispute stage is divided into two. Figure 4 shows a graph diagram of the gas consumption used for each stage, divided by 7.1×10^5 for the publish stage, 7.2×10^5 for the buy-request stage, 11×10^5 for the trade stage, 9.4×10^5 for the dispute stage, and 8.5×10^5 for the resolution stage. It can be seen that, on average, approximately 53×10^5 of gas is consumed throughout the entire phase under the worst assumption that it occurs up to the dispute phase.

5 Conclusions

This paper proposed a fair and secure data trading protocol based on blockchain technology. The proposed scheme uses a randomly selected node among blockchain nodes as an arbitrator node to resolve

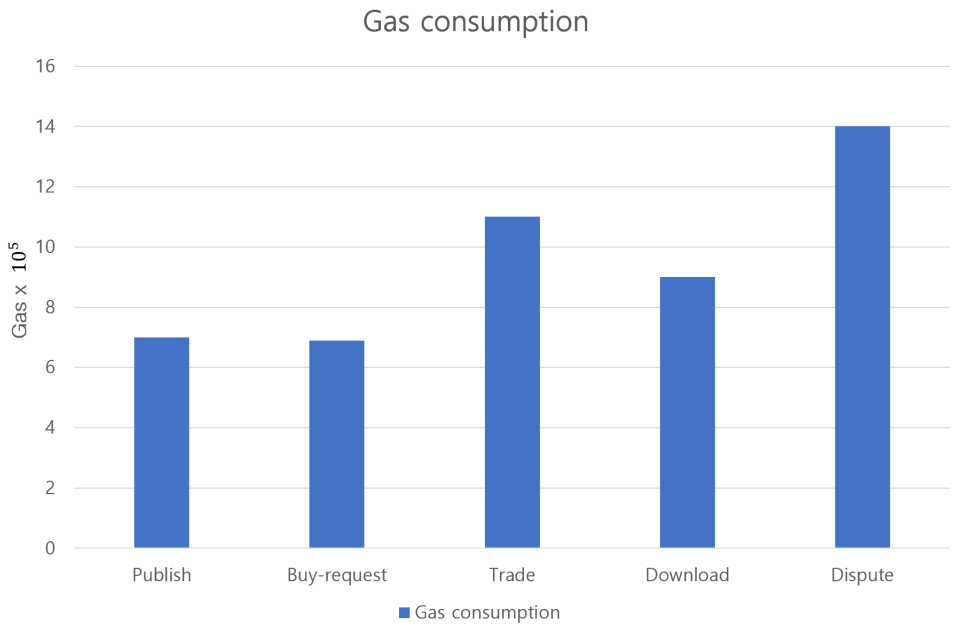

Figure 4: Gas consumption at each stage

disputes instead of TTP. Due to the property that it is impossible to predict which node will be selected as an arbitrator node before the trading protocol is executed, the security and decentralization of the proposed protocol are guaranteed. In addition, AONT and DRC schemes are applied to ensure the security requirements for a fair and secure data trading platform. The proposed protocol satisfies fairness, confidentiality, and timeliness and is secured against collusion attacks. Compared to existing schemes, the proposed scheme has advantages in computational complexity and storage overhead on the on-chain SC.

As a future research, the formal security of the proposed protocol will be studied. One of the most widely used methods to describe and analyze complex cryptographic protocols is the universal composability (UC) framework of Canetti [23]. In the UC framework the security of a cryptographic protocol is analyzed by comparing its real world execution with an idealized protocol running in an ideal world. Therefore, we will study the provable scurity of the proposed protocol in the UC framework, and also explorer secure fair trading protocols through the application of other cryptographic primitives, such as attribute-based encryption [24] and access control encryption [25].

Acknowledgments

This work was supported by a Research Grant of Pukyong National University(2021).

References

[1] X. Wei, Y. Yan, S. Guo, X. Qiu, and F. Qi. Secure data sharing: Blockchain enabled data access control framework for iot. *IEEE Internet of Things Journal*, 9(11):8143–8153, June 2021.

[2] D. Koteshov. Manufacturing Digitization: Facts, Figures, And Predictions, June 2019. `https://www.elinext.com/industries/manufacturing/trends/\manufacturing-facts-figures-predictions/` [Online; Accessed on 11 January 2022].

[3] D. Pöhn and W. Hommel. Universal identity and access management framework for future ecosystems. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Application*, 12(1):64–84, March 2021.

[4] W. Dai, C. Dai, KKR. Choo, C. Cui, D. Zou, and H. Jin. Sdte: A secure blockchain-based data trading ecosystem. *IEEE Transactions on Information Forensics and Security*, 15:725–737, July 2019.

[5] S. Bajoudah, C. Dong, and P. Missier. Toward a decentralized, trust-less marketplace for brokered iot data trading using blockchain. In *Proc. of the 1st IEEE international conference on blockchain (Blockchain'19), Atlanta, USA*, pages 339–346. IEEE, July 2019.

[6] Z. Kodric, S. Vrhovec, and L. Jelovcan. Securing edge-enabled smart healthcare systems with blockchain: A systematic literature review. *Journal of Internet Services and Information Security*, 11(4):19–32, November 2021.

[7] J. Chen and S. Micali. Algorand, May 2017. `https://arxiv.org/pdf/1607.01341v9.pdf` [Online; Accessed on August 08, 2022].

[8] T. Diament, HK. Lee, AD. Keromytis, and M. Yung. The dual receiver cryptosystem and its applications. In *Proc. of the 11th ACM conference on Computer and communications security (CCS '04), Washington DC, USA*, pages 330–343. ACM, October 2004.

[9] RL. Rivest. All-or-nothing encryption and the package transform. In *Fast Software Encryption*, volume 1267 of *Lecture Notes in Computer Science*, pages 210–218. Springer Berlin Heidelberg, January 1997.

[10] JL. Ferrer-Gomila, MF. Hinarejos, and AP. Isern-Deya. A fair contract signing protocol with blockchain support. *Electronic Commerce Research and Applications*, 36:100869, July-August 2019.

[11] H. Huang, KC. Li, and X. Chen. A fair three-party contract singing protocol based on blockchain. In *Proc. of the 9th International Symposium on Cyberspace Safety and Security (CSS'17), Xian, China*, volume 10581 of *Lecture Notes in Computer Science*, pages 72–85. Springer, October 2017.

[12] L. Zhang, H. Zhang, J. Yu, and H. Xian. Blockchain-based two-party fair contract signing scheme. *Information Sciences*, 535:142–155, October 2020.

[13] Z. Li, W. Lei, H. Hu, and W. Zhang. A blockchain-based communication non-repudiation system for conversational service. In *Proc. of the 13th IEEE International Conference on Anti-counterfeiting Security and Identification (ASID'19), Xiamen, China*, pages 6–10. IEEE, October 2019.

[14] Y. Xu, J. Ren, G. Wang, C. Zhang, J. Yang, and Y. Zhang. A blockchain-based nonrepudiation network computing service scheme for industrial iot. *IEEE Transactions on Industrial Informatics*, 15(6):3632–3641, June 2019.

[15] L. Wang, J. Liu, and W. Liu. Staged data delivery protocol: A blockchain-based two-stage protocol for non-repudiation data delivery. *Concurrency and Computation: Practice and Experience*, 33(13):e6240, July 2021.

[16] Li YN, X. Feng, J. Xie, H. Feng, Z. Guan, and Q. Wu. A decentralized and secure blockchain platform for open fair data trading. *Concurrency and Computation: Practice and Experience*, 32(7):e5578, April 2020.

[17] A. Alsharif and M. Nabil. A blockchain-based medical data marketplace with trustless fair exchange and access control. In *Proc. of the 39th IEEE Global Communications Conference (GLOBECOM'20), Taipei, Taiwan*, pages 1–6. IEEE, December 2020.

[18] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *Proc. of the 37th IEEE symposium on Security and Privacy Workshops (SPW'16), San Jose, California, USA*, pages 839–858. IEEE, May 2016.

[19] C. Gu, Y. Zhu, and Y. Zhang. An id-based optimistic fair signature exchange protocol from pairings. In *Computational Intelligence and Security*, volume 3802 of *Lecture Notes in Computer Science*, pages 9–16. Springer Berlin Heidelberg, December 2005.

[20] S. He, Y. Lu, Q. Tang, G. Wang, and CQ. Wu. Fair peer-to-peer content delivery via blockchain. In *Proc. of the 26th European Symposium on Research in Computer Security (ESORICS '21), Darmstadt, Germany*, volume 12972 of *Lecture Notes in Computer Science*, pages 348–369. Springer-Cham, October 2021.

[21] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In *Advances in Cryptology — ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532. Springer Berlin Heidelberg, December 2001.

[22] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza. Succinct {Non-Interactive} zero knowledge for a von neumann architecture. In *Proc. of the 23rd USENIX Security Symposium (USENIX Security'14), San diego, California*, pages 781–796. USENIX Association, August 2014.

[23] R. Canetti. Security and composition of multiparty cryptographic protocols. *Journal of CRYPTOLOGY*, 13(1):143–202, April 2000.

[24] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *Proc. of the 28th IEEE symposium on Security and Privacy Workshops (SPW'07), Berkeley, California, USA*, pages 321–334. IEEE, May 2007.

[25] I. Damgård, H. Haagh, and C. Orlandi. Access control encryption: Enforcing information flow with cryptography. In *Theory of Cryptography Conference*, volume 9986 of *Lecture Notes in Computer Science*, pages 547–576. Springer Berlin Heidelberg, October 2016.

Author Biography



Mi Hyeon Jeon received her B.E. degree in Dept. of IT Convergence and Application Engineering from Pukyong University, Republic of Korea in 2021. She is currently a master course student in Dept. of Information Security from Pukyong National University. Her research interests are related with blockchain, IoT, and supply chain.



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, College of Information Technology and Convergence, Pukyong National University. His research interests include cryptographic protocol, mobile and wireless network security, digital forensics, e-Discovery, and multimedia content security.