Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.039462

Article

Blockchain-Based Secure and Fair IoT Data Trading System with Bilateral Authorization

Youngho Park[1], Mi Hyeon Jeon[2] and Sang Uk Shin[3],*

[1]Electronics and Information Communications Research Center, Pukyong National University, Busan, Korea
[2]Department of Information Security, Graduate School, Pukyong National University, Busan, Korea
[3]Division of Computer Engineering, Pukyong National University, Busan, Korea

*Corresponding Author: Sang Uk Shin. Email: shinsu@pknu.ac.kr

Received: 31 January 2023; Accepted: 29 May 2023; Published: 30 August 2023

Abstract: These days, data is regarded as a valuable asset in the era of the data economy, which demands a trading platform for buying and selling data. However, online data trading poses challenges in terms of security and fairness because the seller and the buyer may not fully trust each other. Therefore, in this paper, a blockchain-based secure and fair data trading system is proposed by taking advantage of the smart contract and matchmaking encryption. The proposed system enables bilateral authorization, where data trading between a seller and a buyer is accomplished only if their policies, required by each other, are satisfied simultaneously. This can be achieved by exploiting the security features of the matchmaking encryption. To guarantee non-repudiation and fairness between trading parties, the proposed system leverages a smart contract to ensure that the parties honestly carry out the data trading protocol. However, the smart contract in the proposed system does not include complex cryptographic operations for the efficiency of on-chain processes. Instead, these operations are carried out by off-chain parties and their results are used as input for the on-chain procedure. The system also uses an arbitration protocol to resolve disputes based on the trading proof recorded on the blockchain. The performance of the protocol is evaluated in terms of off-chain computation overhead and on-chain gas consumption. The results of the experiments demonstrate that the proposed protocols can enable the implementation of a cost-effective data trading system.

Keywords: Bilateral authorization; blockchain; data marketplace; fair exchange; policy matching; secure data trading

1 Introduction

Internet of Things (IoT) is the interconnection of physical devices, such as appliances, vehicles, and wearable devices, through the Internet. It has a number of benefits for industries and professionals. IoT devices are equipped with sensors to generate massive amounts of data. Moreover, these interconnected devices can communicate with each other and with humans, which leads to new applications and services that can improve efficiency, convenience, and quality of life [1]. The data generated by IoT devices can also be analyzed to gain insights and make informed decisions. With the development of IoT infrastructure and the rapid growth of IoT devices in recent years, we are experiencing an unprecedented surge in data.

Nowadays, there are many IoT application domains in our lives, such as healthcare, transportation, environment, and so on [2,3], and we are living in the era of the data economy. The huge volume of data collected by various IoT devices is regarded as a valuable asset. In particular, the data can be used for commercial purposes with the proliferation of machine learning solutions for artificial intelligence and big data analytics [4,5]. Such a trend requires a data marketplace [6,7] for buying and selling data, as shown in Fig. 1. However, online data trading platforms face some restrictions in trustworthiness. First of all, security and fairness are crucial challenges because online trading is carried out between non-face-to-face participants who do not fully trust each other. Therefore, it is essential to develop a secure and fair data trading platform where data can be traded between sellers and buyers in a trustworthy manner.

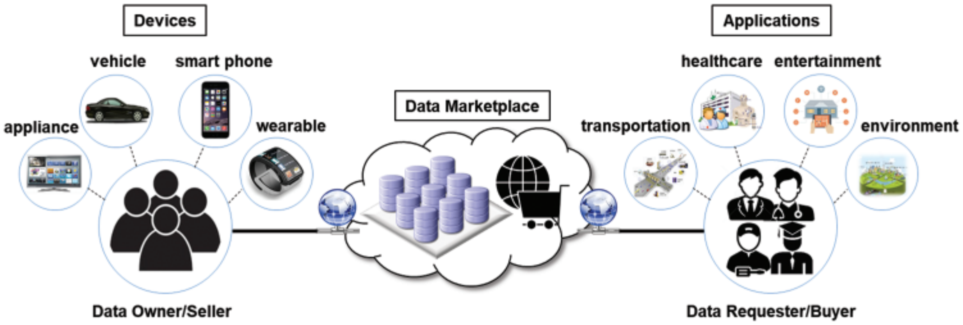

Figure 1: Concept of IoT data marketplace

Fair data trading implies that either the seller gets paid for the data and the buyer obtains the purchased data, or both parties fail to achieve their desired outcome. However, since online parties tend not to trust each other, it is hard to achieve a fair exchange without a trusted intermediary [8]. Hence, conventional data trading systems rely on a trusted third party (TTP) that has centralized control of the data trading platform and responsibility for recording the evidence of interactions [9,10]. However, the lack of accountability and transparency in such a centralized system are still concerns, as well as the single point of failure problem.

Blockchain is a decentralized tamper-resistant ledger technology that offers a secure and transparent way to record and verify transactions [11,12]. It functions as a distributed database where every node in the blockchain network has a copy of the ledger. Each transaction is verified and added to the blockchain through a consensus mechanism, which ensures that all nodes agree on the validity of the transaction. Once a transaction is added, it becomes immutable and cannot be altered or deleted. While commonly used to keep track of financial transactions, blockchain can be applied to a wide range of applications associated with smart contracts.

Since the advent of the blockchain, it is regarded that the role of the TTP can be replaced with the blockchain associated with smart contracts due to its reliability, transparency, and financial properties. Although the blockchain has the advantage that it can provide a decentralized platform for transparent and immutable ledgers, it has a limit on storage capacity. So, it is not viable to store bulk data on the blockchain. As an approach to this limitation, a combination of blockchain and external storage, such as the cloud, is being considered. In this approach, external storage offers a way to store and access the actual data for sale, whereas the blockchain is used to keep track of the actions taken by both the seller and the buyer during data trading.

However, in such a system model, it is required to guarantee access control and source identification for the data managed by the external storage service. Data owners (or sellers) may want to specify who can access their data entrusted to external storage as a policy. Data requesters (or buyers) may want to specify an attribute for certain data owners from whom they want to purchase data. Taking healthcare data as an example, a seller may want to provide its data only to hospitals or doctors but not insurance companies, and a buyer may want to obtain the data from men in their 20 s. This requires bilateral authorization, where the seller and the buyer must meet each other's policies. Therefore, it is necessary to design a secure and fair data trading system that enables access control and source identification for both parties.

Regarding the fair trading protocol considered in this paper, Chen et al. proposed a blockchain-based non-repudiable IoT data trading [13], where the trading behaviors of the seller and the buyer are recorded on the blockchain to facilitate dispute resolution. However, the authors do not address the issue of secure data trading. Thereby, data confidentiality is not guaranteed, as the secret key for data decryption is published in plaintext on the blockchain. In [14], the authors introduced the idea of a secure and fair data trading system, but did not present the detailed protocol for bilateral authorization.

Inspired by [13], in this paper, we aim to enhance the protocol of Chen et al. by taking secure data trading into account from the viewpoint of access control to the data and source identification. More specifically, in order to design a bilateral authorization-enabled secure data trading system, the proposed system makes use of the identity-based matchmaking encryption (IB-ME) scheme [15]. In the proposed system, the seller specifies the attribute of the target buyer (i.e., policy) under IB-ME encryption, and the buyer specifies the attribute of the target seller under IB-ME decryption. By exploiting the security guarantee of IB-ME, the data encrypted by the seller can be decrypted only by the intended buyer. Moreover, if the decryption is correct, it implies that the seller and the buyer are both valid parties specified by the policies of each other.

To support the use of IB-ME, a trusted off-chain arbitrator also acts as the key generator for IB-ME in the proposed system. However, the role of the arbitrator differs from that of the TTP in the conventional system. In the conventional system, the TTP is responsible for managing all trading transactions as interactive evidence which will be used to make arbitration when a dispute occurs. On the other hand, because the interactions between the seller and the buyer are handled by the smart contract on the blockchain, the arbitrator is not directly involved in dealing with data trading in the proposed system, except in a disputable situation.

At this phase, it is important to note that encrypting the whole data (in the form of a large file) by using IB-ME may not be practical due to its performance. Hence, the proposed system incorporates the use of the all-or-nothing transform (AONT) [16] to input only a few transformed data blocks to IB-ME encryption while still maintaining the confidentiality of the whole data. Due to the property of AONT, it is hard to recover the original data without knowing all parts of the transformed data. In the proposed system, the seller first transforms its data by AONT and splits the transformed data into two parts, one small part and the remaining large part. Then, the seller provides the large part through external storage and publishes the small part on the blockchain in encrypted form using IB-ME. To obtain the complete data, the buyer must purchase the encrypted small part by paying the cost through the smart contract even if the buyer can access the large part from external storage. Therefore, the proposed system reduces the storage burden of the blockchain and the computations of IB-ME while maintaining the security of the data.

The contributions of this paper are summarized as follows:

- A secure data trading system architecture based on blockchain and external storage is proposed, which enables access control by the seller and source identification by the buyer at the same time.
- To guarantee fairness, the threats caused by a dishonest seller or buyer are classified, and a fair data trading protocol is designed with the inclusion of arbitration. The proposed protocol encourages trading parties to act with honesty by utilizing the smart contract.
- To demonstrate the efficiency of the proposed protocol, the off-chain computation overhead and the on-chain costs are evaluated.

The rest of this paper is organized as follows: Section 2 presents encryption schemes for secure data sharing/trading and related work on blockchain-based data trading systems. System architecture and design goals considered in this paper are presented in Section 3. Cryptographic building blocks for the proposed system are presented in Section 4. The proposed secure and fair data trading protocol is designed in Section 5. Security and performance of the protocol are evaluated in Section 6. Finally, Section 7 concludes this paper.

2 Related Work

In order to implement access control for secure data sharing in IoT and cloud computing, attribute-based encryption (ABE) [17] is widely used. ABE is a promising tool that enforces receiver access control to limit access to sender data, but does not provide sender access control for source identification [18]. Access control encryption (ACE) [19] is another type of encryption that allows fine-grained control over information flow. However, ACE is more suitable for organizations with hierarchical regulation rather than IoT environments [20]. Recently, a new cryptographic primitive called matchmaking encryption (ME) was proposed [15]. ME enables the sender to specify receivers who can reveal the messages, and the receiver to determine that the received message is from the desired sender.

Research on a blockchain-based decentralized data trading model has received a great deal of attention, and several solutions for data sharing/trading in the field of the IoT industry have been introduced [21–24]. Kang et al. proposed a data trading strategy in the vehicular P2P network in which blockchain and smart contracts are adopted for secure data caching and authorized data sharing [25]. Dixit et al. proposed a decentralized platform for the digital data marketplace enabled by the blockchain, which hosts IoT data in a reliable and fault-tolerant manner [26].

With regard to secure and fair data trading, Dai et al. proposed a blockchain-based secure data trading ecosystem [27] named SDTE. In their system, the data broker conducts business with the buyer on behalf of the seller, but neither the broker nor the buyer can access the raw data owned by the seller. However, SDTE requires a special hardware security module such as Intel Software Guard Extensions (SGX). Li et al. introduced a decentralized data trading framework based on blockchain to guarantee data availability and fairness in data trading [28]. They presented two different solutions. One is to improve reliability and data availability by making use of homomorphic encryption and data sample techniques. The other is to integrate smart contract with double-authentication-preventing signatures [29] to achieve fairness during data trading.

Alsharif et al. proposed a blockchain-based medical data marketplace model [30], in which sellers enforce access control policies on the encrypted records and buyers verify the correctness of the records without revealing any information about them. In order to achieve the design goals, their model is based on ciphertext-policy attribute-based encryption (CP-ABE) [31] and zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK) [32]. However, their model has the weakness that the secret value for decrypting the ciphertext is disclosed on the blockchain at the withdrawal phase.

Li et al. [33] proposed a blockchain-based secure data trading platform by using plaintext checkable encryption (PCE) [34]. Regarding transaction security and data protection, the encrypted data for sale is stored not on a blockchain but on distributed storage to alleviate the storage pressure of the blockchain. Instead, the decryption key of the data is traded on the blockchain platform. Fair exchange on this platform relies on the miners who act as arbitrators for resolving disputes and executing smart contract. However, in the event of a dispute, it causes a problem that the ciphertext and secret key are known to miners who are not trusted on the blockchain network.

The systems proposed by Alsarif et al.'s and Li et al.'s are particularly relevant to the proposed system. However, their systems only focused on the access control of the seller for data confidentiality, and also involved complex cryptographic operations with smart contracts that resulted in high computational costs on the blockchain. On the other hand, the proposed system does not involve complex cryptographic operations in on-chain procedures, but these operations are processed by each off-chain party. Table 1 briefly shows the features of the proposed system.

Table 1: Comparison of the proposed system with existing systems

	Alsarif et al.'s [30]	Li et al.'s [33]	The proposed
Crypto primitives	CP-ABE, zk-SNARK	PCE, digital signature	IB-ME, AONT
Authorization	Unilateral	Unilateral	Bilateral
Fair exchange	Guaranteed	Guaranteed	Guaranteed
TTP	KDC for key generation	No trusted entities, relying on the untrusted miner, but no incentive or penalty for the miners	Arbitrator for key generation, not directly involved in the trading
On-chain overhead	Expensive cryptographic operations	Expensive cryptographic operations	Basic operations

3 System Model

3.1 Architecture

Fig. 2 shows the data trading system architecture which consists of data owner (or seller), data requester (or buyer), arbitrator, blockchain layer, and storage layer.

- Data Owners or Sellers ($\mathbb{S} = \{S_1, S_2, \ldots,\ S_m\}$):

Each seller $S_i \in \mathbb{S}$ advertises and offers its own data for sale through the data trading system. In order to allow only the buyer desired by the seller to access the data, S_i sets the access control policy which specifies the preferable type of buyer. Then, S_i publishes the trading data encrypted by using IB-ME under its access policy. For this purpose, S_i needs to obtain its identity-based encryption key for IB-ME from the arbitrator.

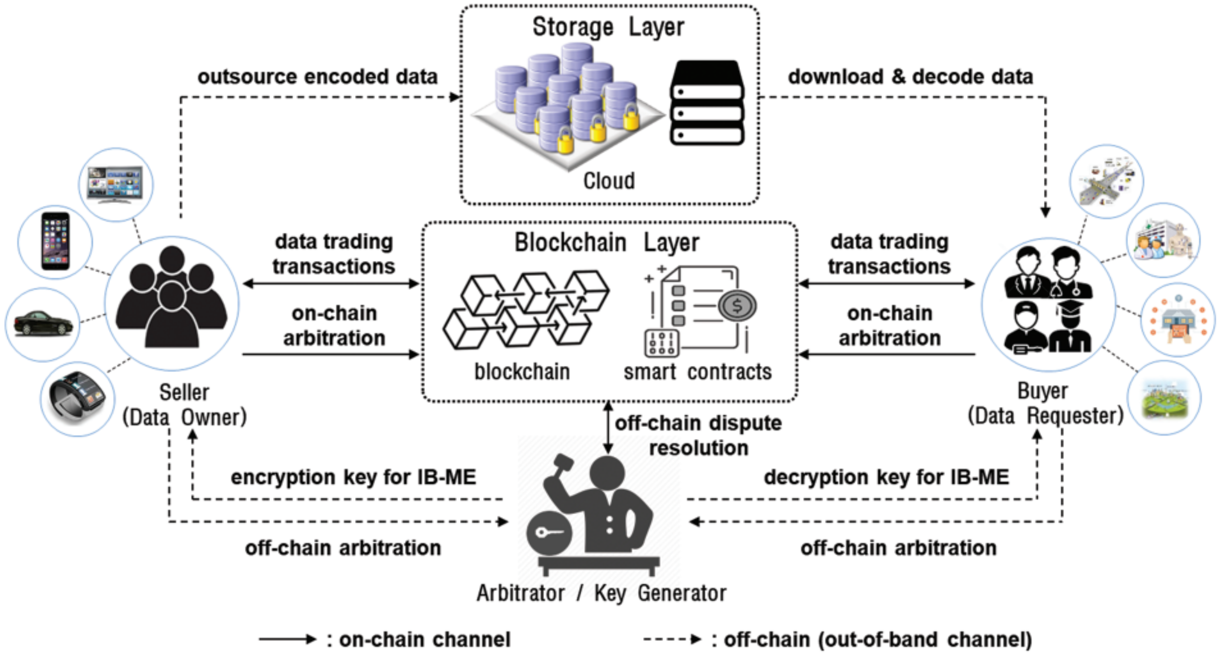

Figure 2: Overview of the proposed data trading system architecture

- Data Requesters or Buyers ($\mathbb{B} = \{B_1, B_2, \ldots, B_n\}$):

A buyer $B_j \in \mathbb{B}$ initiates the data purchase by requesting interesting data from the system. At this phase, the buyer B_j specifies the policy required to a data seller, and then places an order with the seller S_i which satisfies the policy of B_j. In order to obtain access right to the data owned by S_i, B_j must be issued the decryption key under its identity from the arbitrator for IB-ME as well as pay the price of the data in accordance with the smart contract. Then, B_j gets to obtain the data if and only if the encrypted data given by S_i is associated with the identity of B_j which corresponds to the access policy of S_i.

- Arbitrator (A):

The arbitrator is an off-chain entity trusted by the trading participants. If a dispute occurs, the arbitrator takes part in off-chain arbitration to resolve the dispute. It is generally assumed that the arbitrator is a neutral third party to help resolve disputes fairly and efficiently. In the proposed system, the arbitrator also acts as a key generator for IB-ME which generates an encryption key for each seller and a decryption key for each buyer, respectively.

- Blockchain Layer:

Blockchain layer provides a trusted platform that enforces data trading rules, coordinates the data trading process, and deals with payment with digital currency. Blockchain records the result and state of the data trading protocol run between the seller and the buyer by means of transactions. Hence, the blockchain can be viewed as a recorder of evidence to prove whether the seller and the buyer comply with the data trading protocol. Smart contract defines the valid state of data trading progress and implements transaction logic to change one state to another. Seller and buyer will interact with each other by way of the blockchain client application which invokes the smart contract to perform agreed steps of the proposed data trading protocol.

- Storage Layer:

Because blockchain is not suitable for bulk data storage, external storage services such as cloud may be adopted in the storage layer to host a huge volume of data. External storage is regarded as an untrustworthy entity, so data owners outsource their data to the storage service in the form of an encoded package for confidentiality.

In addition, to clarify the proposed system, we make the following assumptions.

- Digital currency payment is implemented on the blockchain and each user has a digital wallet address/account with a balance.
- The operations of the blockchain network are generic and usually understood. Each transaction submitted to the blockchain network contains the digital signature of the transaction issuer, and only confirmed transactions are included in a block appended to the blockchain.
- The underlying blockchain platform is fault-tolerable even in the presence of malicious actors or failed components. Hence, the blockchain can act as a reliable platform and trusted third party.

3.2 Threat Model and Design Goals

Sellers and buyers who participate in online data trading are not fully trusted and may attempt to cheat each other for their own benefit. Table 2 classifies the behaviors of the seller and the buyer considered in the data trading system. Each type of honest or dishonest participant is denoted as HS, HB, DS, and DB.

Table 2: Classification of the behaviors by sellers and buyers

	Seller	Buyer
Honest	- Provide the data correctly (HS)	- Pay the cost of the data (HB)
Dishonest	- Refuse to provide the data (DS1) - Provide wrong data (DS2)	- Refuse to pay for the data (DB1) - Ask for false compensation (DB2)

A dishonest seller may attempt to charge the buyer for payment without providing the data or correct key, or may not provide the data by denying the receipt of the payment even though the buyer has paid. Moreover, in order to make unfair profits, a dishonest seller may provide wrong data that differs from what the buyer requested. On the other hand, even after taking the data, a dishonest buyer may deny receiving the data and refuse to pay for it. In addition, a dishonest buyer may deliberately ask for unfair compensation by falsely alleging that the data or key provided is incorrect, despite having received the accurate one.

Under the threat models, we take the following design goals into account for the proposed blockchain-based secure and fair data trading system.

- *Bilateral authorization and policy matching*: For secure and authorized data trading, both the seller and the buyer can specify their policies that the other party must satisfy to trade the data. That is, the data trading between the seller and the buyer is achieved only if their attributes satisfy the policies specified by each other.

- *Non-repudiation and fraud prevention*: The participants must carry out their responsibilities for data trading, and the transactions cannot be denied later by either of the parties involved in the data trading. Furthermore, malicious behaviors such as providing incorrect data or alleging false compensation must be prevented. In this case, no benefit should be provided to the malicious party.
- *Fair exchange*: At the end of the data trading protocol, either the seller receives the payment and the buyer receives the purchased data, or neither of them receives anything. In other words, if all participants honestly behave according to the protocol, then they will receive what they want.

4 Cryptographic Building Blocks

This section briefly outlines the IB-ME [15] and the AONT [16] which serve as cryptographic building blocks of the proposed system.

4.1 Identity-Based Matchmaking Encryption

Let $e\ : \mathbb{G} \times \mathbb{G}_T$ be a bililnear pairing and P be a generator of $\mathbb{G}$, where $\mathbb{G}$ and $\mathbb{G}_T$ be two groups of a prime order q. When we denote by snd and rcv the target identities (i.e., the access policy) respectively specified by the receiver and by the sender, IB-ME is constructed as follows.

1) Setup (1^λ): On input the security parameter 1^λ, the setup algorithm chooses two random values $r,\ s \in Z_q$, and sets $P_0 = P^r$. It outputs the master public key mpk $= (e, \mathbb{G}, \mathbb{G}_T, q, P, P_0, H_1, H_2, H_3, \Phi)$ and master secret key msk $= (r, s)$, where $H_1 : \{0,1\}^* \to \mathbb{G}$, $H_2 : \{0,1\}^* \to \mathbb{G}$, $H_3 : \mathbb{G}_T \to \{0,1\}^l$ are hash functions, and $\Phi :\ \{0,\ 1\}^n \to \{0,\ 1\}^l$ is a polynomial-time computable padding functioin.
2) SKGen (mpk, msk, σ): On input the master secret key msk, and identity σ, it outputs the encryption key $ek_\sigma = H_2(\sigma)^s$.
3) RKGen (mpk, msk, ρ): On input the master secret key msk, and identity ρ, it outputs the decryption key $dk_\rho = (dk_\rho^1 = H_1(\rho)^r, dk_\rho^2 = H_1(\rho)^s, dk_\rho^3 = H_1(\rho))$.
4) Enc (mpk, ek_σ, rcv, m): On input an encryption ek_σ, a target identity rcv $= \rho$, and a message m, this algorithm proceeds as follows:
    1. Choose two random values $u,\ t \in Z_q$.
    2. Compute $T = P^t$ and $U = P^u$.
    3. Compute $k_R = e\left(H_1(\rho),\ P_0^u\right)$ and $k_S = e(H_1(\rho),\ T \cdot ek_\sigma)$.
    4. Compute $V = \Phi(m) \oplus H_3(k_R) \oplus H_3(k_S)$.
    5. Output ciphertext $C = (T,\ U,\ V)$.
5) Dec (mpk, dk_ρ, snd, C): On input a decryption key dk_ρ, a target identity snd $= \sigma$, and a ciphertext C, this algorithm proceeds as follows:
    1. Parse C as $(T,\ U,\ V)$.
    2. Compute $k_R = e\left(dk_\rho^1,\ U\right)$ and $k_S = e\left(dk_\rho^2, H_2(\sigma)\right) \cdot e\left(dk_\rho^3, T\right)$.
    3. Compute $\Phi(m) = V \oplus H_3(k_R) \oplus H_3(k_S)$.
    4. If the padding is valid, return m. Otherwise, return $\perp$.

4.2 All-or-Nothing Transform

AONT is a randomized transformation that can be reversed, but it is difficult to do so without having knowledge of all the message blocks in the output. It can be used as input to an encryption algorithm. Encode and decode of AONT are constructed as follows.

1) Encode (M): Let the input M be the sequence of message blocks $m_1, m_2, \ldots, m_n$. Encoding proceeds as follows:
   1. Choose a random value r.
   2. Compute $c_i = m_i \oplus G(r|i)$, for $1 \leq i \leq n$, where G is a cryptographically secure pseudo-random function.
   3. Compute $c_0 = r \oplus H(c_1|c_2|\ldots|c_n)$, where H is a cryptographic hash function.
   4. Output the encoded message $(stub|package) = (c_0|c_1|\ldots|c_n)$, where $stub$ can be set by the first a few blocks (ex., $stub = c_0|c_1$ and $package = c_2|\ldots|c_n$).
2) Decode ($stub|package$): Given the encoded message $(stub|package) = (c_0|c_1|\ldots|c_n)$, decoding proceeds as follows:
   1. Compute $r = c_0 \oplus H(c_1|c_2|\ldots|c_n)$.
   2. Compute $m_i = c_i \oplus G(r|i)$, for $1 \leq i \leq n$.
   3. Output the message $M = m_1|m_2|\ldots|m_n$.

5 Proposed System

The proposed blockchain-based secure data trading system is presented in this section. Fig. 3 shows the state transition of the proposed data trading protocol processed by the smart contract, and Table 3 describes the notations used in the protocol. Once the seller has registered its encoded data to the storage layer, the buyer can request access right to decrypt the data by invoking the order function of the smart contract. The seller will then offer the requested access right through the blockchain. If there is no order to the registered data or the requested access right is not offered within a predefined expiration time, then the smart contract will cancel the data trading process. When the data is successfully recovered with the purchased access right, the buyer confirms this data trade. Then, the seller receives the payment. However, if any fraudulent behavior occurs by either the seller or the buyer, a dispute resolution process will be initiated.

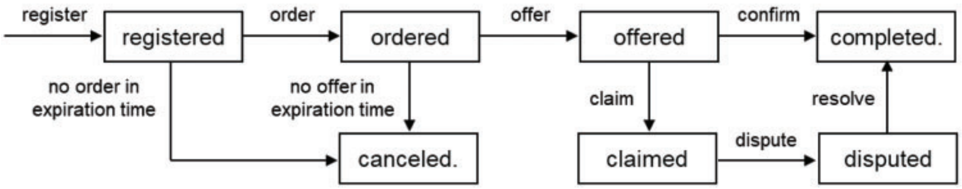

Figure 3: State transition of the proposed data trading protocols

Table 3: Notations used in the proposed protocols

Notation	Description
$\mathcal{D}$	Data of the seller for sale
$H(x)$	Cryptographic hash for the input x
S_i, B_j	Seller and buyer on the data trading protocol
msk, mpk	Master secret key and public parameters of the arbitrator for IB-ME
snd	Target identity required by the buyer as a policy to sellers
rcv	Target identity required by the seller as a policy to buyers
ek_{S_i}, dk_{B_j}	Encryption key of S_i and decryption key of B_j for IB-ME

(Continued)

Table 3 (continued)

Notation	Description
Σ_{ij}	State of the data trading protocol run between S_i and B_j
$X \overset{?}{=} Y$	Operation to check if X and Y are same or not
TX_f	Transaction for the procedure f of the data trading
$TX{:}=\{\,\}$	Composition of the TX
$TX.value$	The value in the data structure of the TX

Suppose that a seller S_i wants to sell data $\mathcal{D}$ and a buyer B_j wants to purchase the data owned by S_i through the proposed data trading system. S_i and B_j perform the data trading protocol with the mediation of the arbitrator A as described in the following sections.

5.1 Data Trading Protocol

Before describing the detailed protocol, we assume that the smart contract was deployed on the blockchain layer and the arbitrator set up its own master secret key *msk* and public parameters *mpk* by running IB-ME.Setup (1^λ) algorithm as a key generator. The *mpk* is publicly known to the system.

5.1.1 Advertisement and Request Phase

The seller S_i advertises a description of the data $\mathcal{D}$ for sale attached with the policy required to buyers, and the buyer B_j searches for the data of interest and submits a request with the policy required to sellers, as follows.

1) S_i submits the advertisement transaction $TX_{ad_i}{:}=\{desc_{\mathcal{D}}, rcv, price_{\mathcal{D}}\}$ to the blockchain layer, where $desc_{\mathcal{D}}$ is brief description about the data $\mathcal{D}$, rcv is the access policy that a buyer must satisfy, and $price_{\mathcal{D}}$ is the price of the data $\mathcal{D}$.
2) If B_j is interested in the data advertised by TX_{ad_i}, B_j submits the request transaction $TX_{req_j}{:}=\{TX_{ad_i}.id, snd\}$, where $TX_{ad_i}.id$ is the identifier referenced by the request and snd is the policy specifying the attribute required to the seller.

5.1.2 Register Phase

When S_i finds the request for its data, S_i first encodes the data by using AONT encoding algorithm and outsources the encoded data to the storage layer. Then, S_i invokes the register function of the smart contract to publish proof of the data to be traded on the system. At this phase, S_i also makes a guarantee deposit that will be temporarily locked in the smart contract and confiscated if S_i would act dishonestly.

1) For data $\mathcal{D}$, S_{i} prepares AONT encoded data as $\left(D_{stub}|D_{pkg}\right) \leftarrow$ AONT.Encode ($\mathcal{D}$), outsources the D_{pkg} part to the storage layer, and then computes the hashes of the data as follows:

$$\Delta_s = H\left(D_{stub}\right) \tag{1}$$

$$\Delta_p = H\left(D_{pkg}\right) \tag{2}$$

$$\Delta_{s|p} = \Delta_s \oplus \Delta_p \tag{3}$$

$$\Delta = H\left(\mathcal{D}\right) \tag{4}$$

2) S_i invokes the register procedure of the smart contract by submitting the transaction $TX_{reg_i} := \{TX_{req_j}.id,\ uri,\ \Delta,\ \Delta_{s|p},\ \Delta_P,\ deposit_{S_i},\ exp_{reg}\}$. Here, $deposit_{S_i}$ is the digital currency that the seller puts down a deposit on, *uri* is the link for downloading the data package from the storage, and Δ, $\Delta_{s|p}$, Δ_P are commitments of the data to be offered later.
3) On input the register call, if the referenced TX_{req_j} exists and S_i's account balance is sufficient for $deposit_{S_i}$, then the smart contract sets the state as $\Sigma_{ij} =$ "registered" and publishes the register transaction.

Note, exp_{reg} specified in TX_{reg_i} is the expiration time until when the buyer must place an order for the registered data. If the buyer does not place an order within exp_{reg}, smart contract will discard this data trading and give $deposit_{S_i}$ back to S_i.

5.1.3 Order Phase

Once the encoded data is registered, B_j can download D_{pkg} from the storage layer. However, due to the property of AONT, D_{pkg} alone is not perfect to recover the actual data $\mathcal{D}$, so B_j needs to order the remaining part D_{stub} to S_i by using the smart contract.

1) B_j downloads D_{pkg} from the storage linked by $TX_{reg_i}.uri$, then computes Δ_p' as follows and checks the validity of D_{pkg} if $\Delta_p' \overset{?}{=} TX_{reg_i}.\Delta_p$.

$$\Delta_p' = H\left(D_{pkg}\right) \tag{5}$$

2) If D_{pkg} is valid (i.e., $\Delta_p' \overset{?}{=} TX_{reg_i}.\Delta_p$ holds), B_j invokes the order procedure by submitting TX_{ord_j} $:= \{TX_{reg_i}.id, pay_{B_j}, deposit_{B_j}, exp_{ord}\}$, where pay_{B_j} is the digital currency to pre-pay the price of the data but locked until the end of the protocol and $deposit_{B_j}$ is B_j's guarantee money.

3) On input the order call, if the current protocol state is $\Sigma_{ij} =$ "registered", $TX_{reg_i}.exp_{reg}$ is not expired and B_j's account balance is sufficient for $pay_{B_j} + deposit_{B_j}$, then smart contract sets the state as $\Sigma_{ij} =$ "ordered" and publishes the order transaction. Otherwise, the order of B_j is discarded and the state is set as $\Sigma_{ij} =$ "canceled".

With regard to exp_{ord}, which is the expiration time until when the seller must offer the remaining data part, if no offer is given by S_i in exp_{ord} then this trading will be canceled and both pay_{B_j} and $deposit_{B_j}$ be returned back to B_j.

5.1.4 Offer Phase

When S_i is notified that the trading state is in "ordered", S_i publishes the D_{stub} on the blockchain by using the smart contract so that B_j can purchase the data reliably. To provide the data in a secure manner, at this phase, S_i encrypts D_{stub} by using IB-ME under the encryption key issued by the arbitrator and the target identity *rcv* representing the access policy to the buyer.

1) The arbitrator issues S_i with the encryption key as $ek_{S_i} \leftarrow$ IB-ME.SKGen $(msk, attr_{S_i})$ under the $attr_{S_i}$ identifying S_i's attribute after checking if S_i's attribute satisfies the buyer's policy *snd* (i.e., $attr_{S_i} = snd$). The key generation is off-chain process, and we assume an out-of-band secure channel.
2) For the remaining data part D_{stub}, S_i generates the encrypted data as $C_{stub} \leftarrow$ IB-ME.Enc $(mpk, ek_{S_i}, rcv, D_{stub})$.

3) S_i offers the encrypted to B_j data through the blockchain by invoking the offer procedure with $TX_{offer_i} := \{TX_{ord_j}.id,\ C_{stub},\ exp_{offer}\}$.
4) If the current state is $\Sigma_{ij} =$ "ordered" and $TX_{ord}.exp_{ord}$ is not expired, smart contract sets the state as $\Sigma_{ij} =$ "offered" and publishes the offer transaction.

5.1.5 Confirmation Phase

When the trading state is set as $\Sigma_{ij} =$ "offered" as a response to B_j's order, B_j retrieves the encrypted data C_{stub} from the offer transaction TX_{offer_i} and reconstructs the AONT encoded data $\left(D_{stub}|D_{pkg}\right)$ after decrypting C_{stub} by using IB-ME under the decryption key issued by the arbitrator and the target identity snd. If the reconstructed data is valid and the actual data $\mathcal{D}$ is successfully decoded, B_j confirms this data trade.

1) B_j retrieves C_{stub} from TX_{offer_i} offered by the seller.
2) The arbitrator issues B_j with the decryption key as $dk_{B_j} \leftarrow$ IB-ME.RKGen $(mpk,\ msk,\ attr_{B_j})$ under the $attr_{B_j}$ identifying B_j's attribute after checking if $attr_{B_j}$ satisfies seller's policy rcv (i.e., $attr_{B_j} = rcv$).
3) B_j decrypts C_{stub} to get the remaining data part D_{stub} as $D_{stub} \leftarrow$ IB-ME.Dec $(mpk,\ dk_{B_j},\ snd,\ C_{stub})$ and combines it with D_{pkg} downloaded from the storage layer to construct the full AONT encoded data $\left(D_{stub}|D_{pkg}\right)$. B_j also computes the followings, where $\Delta'_p = H\left(D_{pkg}\right)$ is the result computed by Eq. (5) at the order phase.

$$\Delta'_s = H\left(D_{stub}\right) \tag{6}$$

$$\Delta'_{s|p} = \Delta'_s \oplus \Delta'_p \tag{7}$$

If $\Delta'_{s|p} \overset{?}{=} TX_{reg_i}.\Delta_{s|p}$ holds, B_j recovers the actual data $\mathcal{D}$ by AONT decoding the $\left(D_{stub}|D_{pkg}\right)$ as $\mathcal{D} \leftarrow$ AONT.Decode $(D_{stub}|D_{pkg})$, and if the validity of the actual data $H(\mathcal{D}) \overset{?}{=} TX_{reg_i}.\Delta$ holds then B_j invokes the confirmation procedure of the smart contract by sending $TX_{conf} := \{TX_{offer_i}.id,$ "ok"$\}$ to finally confirm this data trade on the blockchain.

4) Upon receiving the confirmation call from the buyer before $TX_{offer}.exp_{offer}$ is timeout in the state $\Sigma_{ij} =$ 'offered', smart contract transfers B_j's payment of $TX_{ord_j}.pay_{B_j}$ to S_i's account, and gives back $TX_{reg_i}.deposit_{S_i}$ and $TX_{ord_j}.deposit_{B_j}$ to S_i and B_j, respectively. The state is set as $\Sigma_{ij} =$ "completed" and the data trading between S_{i} and B_j is completed normally.

5.2 Arbitration Protocols

When the seller and the buyer honestly follow the data trading protocol as described above, they can get the payment and the data, respectively. However, one party may repudiate its trading behavior or have complaints against the repudiation of the other party. Hence, an arbitration protocol to resolve such problematic situations is needed. The proposed arbitration protocols are divided into on-chain arbitration and off-chain arbitration. The former is carried out by the smart contract on the basis of the recorded proof on the blockchain, and the latter is carried out by the arbitrator when any party raises an objection to the on-chain arbitration result.

5.2.1 On-Chain Arbitration

Buyer and seller can initiate the on-chain arbitration procedure of the smart contact. From the seller's viewpoint, when S_i finds out that the buyer B_j did not confirm the data trade even though B_{j} had taken the data, S_i invokes on-chain arbitration to get B_j's payment. On the other hand, at the confirmation phase, when B_j finds that the digests $\Delta'_{s|p}$ and $\Delta' = H(\mathcal{D})$ computed by itself are not the same as $\Delta_{s|p}$ and Δ committed to the blockchain, B_j invokes on-chain arbitration by giving $\Delta_{\mathrm{s}}^{'}$ and Δ' as the evidence. On-chain arbitration is processed as follows.

1) For S_i's on-chain arbitration call, if exp_{offer} is expired and the state Σ_{ij} is not "completed" but still in "offered", set result = "buyer_not_confirmed" which decides that the buyer did not confirm receipt of data so the payment is not settled to the seller yet.
2) For B_j's on-chain arbitration call, verify the correctness of the data by computing and checking if $\left(\Delta_{s}^{'} \oplus TX_{reg_i}.\Delta_p\right) \overset{?}{=} TX_{reg_i}.\Delta_{s|p}$ and $\Delta' \overset{?}{=} TX_{reg_i}.\Delta$. If the verification is fail (i.e., $\Delta_{s}^{'} \oplus TX_{reg_i}.\Delta_p \neq TX_{reg_i}.\Delta_{s|p}$ or $\Delta' \neq TX_{reg_i}.\Delta$), set the result = "seller_data_not_correct".
3) On the result of the above, set the state as Σ_{ij} = 'claimed' and publish the transaction $TX_{claim} := \{TX_{offer}.id$, invoker, result, $\Delta_{s}^{'}$, Δ', $exp_{claim}\}$.

After the TX_{claim} is published on the blockchain, if both parties have no objections to the result and do not initiate off-chain arbitration until exp_{claim} has passed, the smart contract finally settles the payment or gives a penalty depending on the result; transfers $TX_{ord_j}.pay_{B_j}$ to S_i's account if the result is "buyer_not_confirmed", or returns $TX_{ord_j}.pay_{B_j}$ back to B_j's account and confiscates S_i's deposit $TX_{reg_i}.deposit_{S_i}$ as the penalty if the result is "seller_data_not_correct".

5.2.2 *Off-Chain Arbitration*

Unfortunately, a malicious seller may give wrong data and a malicious buyer may present false evidence, which makes the on-chain arbitration lead to a wrong decision. With claiming "seller_wrong_data", it is possible that the buyer B_j attempts to cheat the on-chain arbitration by giving intentionally forged false $\Delta_{\mathrm{s}}^{'}$ and Δ' so as to deny paying the cost even though B_{j} received the correct data. It is also possible that the seller S_i cheats the verification by presenting wrong or useless data $\mathcal{D}'$ and recording hash values derived from $\mathcal{D}'$ on the blockchain from the beginning. The verification of the data by computing the hash values will be definitely passed as valid whereas the buyer receives wrong data different from what the buyer wants. On-chain arbitration is not sufficient to handle those suspicious behaviors, so data trading parties cannot help but rely on the judgment of the arbitrator.

Therefore, in such cases, S_i or B_j initiates off-chain arbitration by sending $TX_{dispute} := \{TX_{offer}.id$, $X_{claim}.id$ } to the blockchain, in which the smart contract will set the state as Σ_{ij} = 'disputed', in order to ask for the arbitrator to resolve the dispute situation. Note, in the proposed system, the arbitrator also acts as the key generator for IB-ME. We make use of the key escrow property inherited from identity-based cryptography in order for the arbitrator to examine the traded data and the proofs recorded on the blockchain on behalf of S_i and B_j. The arbitrator deals with the off-chain arbitration as follows.

1) First, collect D_{pkg} from the storage and C_{stub}, Δ, $\Delta_{s|p}$, Δ_P, $\Delta_{s}^{'}$, Δ' from the blockchain.
2) Reconstruct the AONT encoded data $\left(D_{stub}|D_{pkg}\right)$ after decrypting C_{stub} by using B_j's escrowed IB-ME decryption key dk_{B_j}. Then, recover the data $\mathcal{D}$ by decoding $\left(D_{stub}|D_{pkg}\right)$, and compute the followings:

$$\Delta_{s}^{\mathrm{A}} = H\left(D_{stub}\right) \tag{8}$$

$$\Delta_{p}^{\mathrm{A}} = H\left(D_{pkg}\right) \tag{9}$$

$$\Delta_{s|p}^{\mathrm{A}} = \Delta_{s}^{A} \oplus \Delta_{p}^{A} \tag{10}$$

$$\Delta^{\mathrm{A}} = H\left(\mathcal{D}\right) \tag{11}$$

3) If $(\Delta_p^{\mathrm{A}}, \Delta_{s|p}^{\mathrm{A}}, \Delta^{\mathrm{A}})$ are the same as S_i's $(\Delta_P, \Delta_{s|p}, \Delta)$ but $(\Delta_s^{\mathrm{A}}, \Delta^{\mathrm{A}})$ are different from B_{j}'s $(\Delta_s^{'}, \Delta')$, then "seller_wrong_data" of the on-chain arbitration is regarded as resulting from B_{j}'s forged false evidence. So, judge that B_j is malicious and set judgment = "buyer_malicious".
4) Even though the verification of hash-based data correctness is valid, further check the usefulness of the data. If the data $\mathcal{D}$ does not meet the description $desc_{\mathcal{D}}$ in TX_{ad_i}, then judge that S_i offered nonsense data and set judgment = "seller_malicious".
5) Invoke the resolve procedure of the smart contract by submitting $TX_{resolve}{:}=\{TX_{dispute}.id$, judgment}.

It is worth noting that, in the blockchain platform, the data $\mathcal{D}$ is an external item and trading the data is a real-world event. So, it is not possible for the on-chain procedure to verify the truth of data $\mathcal{D}$ on its own. Therefore, in step (4), the arbitrator can judge whether the data $\mathcal{D}$ actually contains useful content corresponding to the description $desc_{\mathcal{D}}$ or meaningless data after looking into the data $\mathcal{D}$.

Upon receiving the resolve call from the arbitrator, the smart contract processes and records the final arbitration result. If the judgment is "seller_malicious" then S_i's deposit $TX_{reg_i}.deposit_{S_i}$ is confiscated as a penalty. On the other hand, if the judgment is "buyer_malicious" then B_j's deposit $TX_{ord_j}.deposit_{B_j}$ is confiscated as a penalty and $TX_{ord_j}.pay_{B_j}$ is paid to S_i. The resolve transaction is published on the blockchain and the protocol is completed.

6 Evaluations

6.1 Security Evaluation

As presented so far, the proposed system is based on blockchain accompanied with smart contract and makes use of IB-ME and AONT schemes. The proposed system satisfies the design goals mentioned in Section 2, assuming the security and reliability features of the underlying cryptographic primitives and blockchain.

6.1.1 Bilateral Authorization and Policy Matching

When the data is provided through external storage instead of by the data owner directly, it would be needed for the seller to specify a policy restricting access to its own data and for the buyer to specify the condition of the data provider. In the proposed data trading protocol, seller access policy rcv required for buyers and buyer policy snd required for sellers are specified in TX_{ad} and TX_{req} at advertisement and request phase, respectively. Data $\mathcal{D}$ of the seller is first transformed to the AONT encoded data $\left(D_{stub}|D_{pkg}\right)$ and only the D_{pkg} part is provided by way of the storage layer, in which it is hard to recover the data without knowing the entire. Hence, to get the actual data, the buyer has to purchase the remaining data D_{stub} to be given in the encrypted form by the seller.

At this phase, the data D_{stub} is protected by using IB-ME under the policies, snd and rcv. To perform data trading, seller and buyer have to be issued their encryption key and decryption key from the arbitrator who checks that the attributes of the seller and the buyer correspond to the policies of the other, respectively. During the protocol, the encrypted data $C_{stub} \leftarrow$ IB-ME.Enc $(mpk, ek_{S_i}, rcv, D_{stub})$ can be decrypted by only the buyer who has the decryption key dk_{B_j} derived from its attribute (that is, $attr_{B_j} = rcv$) as $D_{stub} \leftarrow$ IB-ME.Dec $(mpk, dk_{B_j}, snd, C_{stub})$. In addition, if the encrypted data is decrypted correctly, the buyer can be sure that the seller who has the encryption key ek_{S_i} derived from its attribute (that is, $attr_{S_i} = snd$) offered the data. Therefore, the data trading is achieved if and only if the seller's policy for authorization to the data and the buyer's policy for source identification are matched.

6.1.2 Non-Repudiation and Fraud Prevention

If the seller and the buyer honestly follow the protocol, then the data trading would be completed satisfactorily. However, the seller and the buyer are not fully trusted participants to each other, so one of them may behave maliciously as mentioned in Section 2. When the seller or the buyer attempts to cheat and repudiate trading behavior, the system must be able to detect and prevent such attempts.

Data trading transactions between seller and buyer are processed by means of the smart contract executed in compliance with the pre-defined rules, and are recorded on the tamper-resistant blockchain ledger. Transactions recorded on the blockchain at each protocol phase can be regarded as proof of the data trading between the seller and the buyer, which enables non-repudiation, and the previous protocol step must be processed before proceeding to the next step.

In the proposed system, in order to acquire the actual whole data $\mathcal{D}$, the buyer must obtain the D_{stub} from the seller at the offer phase after pre-paying the cost pay_{B_j} of the data to the smart contract at the order phase. That is, the trading proceeds according to the sequence of "ordered" → "offered" state transition. Assuming the secrecy of IB-ME and AONT schemes, the buyer can hardly recover the data $\mathcal{D}$ without D_{stub} as discussed above. Therefore, the system can prevent DB1 type behavior because the buyer has no choice but to prepay the cost to recover the perfect data.

The system can also prevent not only DS1 type behavior but also DS2 type behavior of the seller even though the buyer prepays the cost. The prepaid pay_{B_j} is locked by the smart contract until the buyer makes a confirmation by checking the correctness of the received data. After the buyer's order transaction, if no offer is given before exp_{ord} is expired, pay_{B_j} is returned back to the buyer by the smart contact and the dishonest seller has nothing. Furthermore, at the confirmation phase, if the buyer received incorrect data whose hashes are not the same as the hashes recorded on the blockchain, the buyer can claim compensation for the incorrect data by invoking on-chain arbitration. Due to the collision resistance property of the cryptographic hash function, the hashes of the traded data (Δ, $\Delta_{s|p}$, Δ_p) recorded on the blockchain makes the seller hard to forge a different data with the same (Δ, $\Delta_{s|p}$, Δ_p). Although DS2 type seller may record both the wrong data and its hashes on the blockchain at the first place to defraud the on-chain arbitration, the wrong data cannot evade the off-chain examination by the arbitrator. So, DS2 type fraud of the seller can be handled and prevented.

Another concern, from the seller's perspective, is the cheating of DB2 type malicious buyer that alleges false compensation. A malicious buyer may input intentionally forged evidence ($\Delta_s^{'} \neq \Delta_{\mathrm{s}}$, $\Delta' \neq \Delta$) for the purpose of leading the on-chain arbitration to "seller_data_not_correct" falsely. To cope with such a suspicious case, in the proposed system, the settlement on the result of on-chain arbitration is suspended for a while so that the seller can ask for the arbitrator to judge whose fraud by invoking off-chain arbitration. Then, the arbitrator collects and examines the data and the proofs, (Δ_P, $\Delta_{s|p}$, Δ) recorded by the seller and ($\Delta_s^{'}$, Δ') by the buyer. DB2 type fraud of buyer can be detected, because the hashes computed by the arbitrator (Δ_s^{A}, Δ^{A}) would be the same as seller's but not as buyer's.

6.1.3 Fair Exchange

Once again, when the seller and the buyer honestly follow the data trading protocol, then the data trading will be completed successfully. Then, the buyer will indeed get the data $\mathcal{D}$, and the seller will get the payment pay_{B_j}, as following the state transition "registered" → "ordered" → "offered" → "completed". However, when the trading is canceled in the middle of the protocol run (i.e., $\Sigma_{ij} =$ 'canceled'), neither the buyer's payment nor the seller's data is provided to them, respectively. So, both of them get nothing. In addition, due to the functionality of non-repudiation and dispute resolution described above, it is difficult for them to obtain unfair benefits by cheating each other, and penalties are even deducted if they perform the data trading protocol dishonestly.

Each party makes a guarantee deposit, $deposit_{S_i}$ and $deposit_{B_j}$, which are locked in the smart contract until the protocol is completed. If no dispute occurs, the smart contract unlocks and returns the deposits back to each honest participant. However, if a dispute occurs, the dishonest party gets to forfeit its deposit as a penalty depending on the on-chain or the off-chain arbitration result. Eventually, the seller and the buyer will lose their guarantee money if they act maliciously during the trading protocol, which encourages both the seller and the buyer to perform the data trading honestly. Thus, the proposed system guarantees fairness so that both parties obtain what they want or gain no befit.

6.2 Performance Evaluation

The basic design principle of the proposed system is not to burden complex cryptographic operations to the smart contract as possible in order to implement lightweight on-chain procedures. Although blockchain platforms such as Ethereum support smart contract to implement complicated crypto algorithms with big-number arithmetic, the more complex operations are burdened, the more time and cost are spent. Therefore, we make the system run only basic simple operations in the on-chain procedure while rather complex cryptographic operations are processed by each off-chain local entity whose results are given to the on-chain procedure as input.

Table 4 shows the cryptographic overhead processed by each entity during the data trading protocol. In the system, the most time-consuming cryptographic scheme is IB-ME. To measure the overhead of IB-ME operations as shown in Table 5, we used the benchmark results of the Miracl cryptography library [35] implemented on Intel Core i7 3 GHz with a supersingular curve with 512-bit based field (i.e., $|\mathbb{G}| = 512$ bits). So, this result shows that each entity can process the operations efficiently.

Table 4: Off-chain cryptographic computational overhead

	Seller	Buyer	Arbitrator
Register	$1AONT.Enc + 4H$		
Order		$1H$	
Offer	$1IB\text{-}ME.Enc$		$1IB\text{-}ME.SKGen$
Confirmation		$1IB\text{-}ME.Dec + 3H + 1AONT.Dec$	$1IB\text{-}ME.RKGen$
Off-chain arbitration			$1AONT.Dec + 4H + 1IB\text{-}ME.Dec$

Table 5: Performance of the IB-ME

	SKGen	RKGen	Encryption	Decryption
Operations	$1MP + 1SM$	$3MP + 2SM$	$2PA + 1MP + 3SM$	$3PA + 1MP + 1Mul$
Time (ms)	5.42	13.54	15.66	17.4

- PA: bilinear pairing
- MP: map to point (i.e., H_1, H_2 function)
- SM: scalar multiplication on $\mathbb{G}$
- Mul: multiplication on $\mathbb{G}_T$

In the proposed system, the interaction between the seller and the buyer is handled by the smart contract whose procedures are triggered by the transactions submitted by the seller and the buyer. We do not restrict the underlying blockchain platform to Ethereum, but we estimate the storage overhead and the gas units consumed in Ethereum to show the cost of the proposed protocols. According to [36], it additionally costs the input data fee to the fixed initial fee of 21000 gas units to execute every transaction. With regard to the input data fee, 4 gas units per zero valued byte of data and 16 gas units per nonzero valued byte of data are paid, respectively. So, the gas costs vary depending on the input data values. Table 6 shows the data field and size used in the proposed system.

Table 6: Data field and size of the proposed protocol

Data	*TX.id*	*exp*	Δ	*sdn, rcv*	*pay*	*deposit*	*uri*	*result*	C_{stub}
Bytes	32	32	32	64	32	32	64	8	267

We evaluated the performance of the on-chain procedures in terms of storage overhead, gas consumption, and economic cost. We implemented the smart contract by using Solidity 0.8.7 with Remix IDE and Ethereum test network, then compared the costs of the proposed system with Alsarif et al.'s [30] and Li et al.'s [33]. The storage and the gas costs of Alsarif et al.'s and Li et al.'s are estimated by using the quantities presented in their work. Strictly speaking, it may not be appropriate to directly compare the measurements because each experiment was done separately in different environments by the authors. Nevertheless, we intend to show that the proposed system can be as efficient as others or better.

In Table 7, the gas costs are total gas units including transaction costs and code execution costs. As mentioned before, heavy cryptographic operations are not included in the proposed on-chain procedures while digital signature verification and PCE check are included in Alsarif et al.'s and Li et al.'s. Hence, the proposed system consumes less gas than others to carry out data trading even though the proposed system inputs more data. Furthermore, to show the economic expenses, we also calculated the gas price in US dollars. At the time of writing this paper, the gas price is on average 19 Gwei for 1 gas unit, 1 Gwei is 1×10^{-9} Ether, and 1 Ether is 1660 $. Therefore, the cost for the data trading in normal is about 13×10^5 gas ($ 41.002) and about 17×10^5 gas ($ 53.618) when disputed.

Table 7: On-chain storage overhead, gas consumption, and economic costs

	Alsarif et al.'s [30]	Li et al.'s [33]		The proposed	
		Normal	Disputed	Normal	Disputed
Input (Byte)	1068	392		947	1239
Gas	$\approx 19 \times 10^5$	$\approx 15 \times 10^5$	$\approx 17 \times 10^5$	$\approx 13 \times 10^5$	$\approx 17 \times 10^5$
Ether	≈ 0.0361	≈ 0.0285	≈ 0.0323	≈ 0.0247	≈ 0.0323
Cost in $	≈ 59.926	≈ 47.31	≈ 53.618	≈ 41.002	≈ 53.618

7 Conclusion

In today's data-driven economy, the vast amount of data gathered by various IoT devices is regarded as a valuable asset, and this trend demands a data trading platform to sell and buy data.

The emergency of the blockchain promotes the development of a decentralized trustworthy data trading platform. Therefore, in this paper, we presented a secure and fair data trading system by taking advantage of the blockchain integrated with smart contract and matchmaking encryption. The proposed system enables bilateral authorization by making use of the security features of IB-ME so that access control by the seller and source identification by the buyer are guaranteed at the same time. Moreover, we designed the fair data trading protocol incorporated with on-chain and off-chain arbitrations by leveraging the smart contract to make the parties honestly carry out the protocol. In addition, we evaluated off-chain computation overhead and on-chain costs of the proposed protocol. In comparison with existing systems relevant to our work, the proposed protocol makes it possible to implement a cost-efficient data trading system. Even though we assumed a single arbitrator, it is more desirable to adopt a decentralized arbitration system with multiple arbitrators to increase the reliability of dispute resolution. Development of a decentralized arbitration platform that allows for fair and transparent dispute resolution remains a future work.

Acknowledgement: The authors are thankful to the reviewers for their helpful comments and suggestions.

Funding Statement: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2022R1I1A3063257), and supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean Government [22ZR1300, Research on Intelligent Cyber Security and Trust Infra].

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. A. Ajagbe, M. O. Adigun, J. B. Awotunde, J. B. Oladosu and Y. J. Oguns, "Internet of things enabled convolutional neural networks: Applications, techniques, challenges, and prospects," in *IoT-enabled Convolutional Neural Networks: Techniques and Applications*, 1st ed., New York, USA: River Publishers, pp. 27–63, 2023.

[2] 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*, vol. 9, no. 11, pp. 8143–8153, 2021.

[3] D. Koteshov, "Manufacturing Digitization: Facts, figures, and predictions," 2019. [online] Available: https://www.elinext.com/industries/manufacturing/trends/manufacturing-facts-figures-predictions/

[4] T. Jung, X. Y. Li, W. Huang, J. Qian, L. Chen *et al.,* "AccountTrade: Accountable protocols for big data trading against dishonest consumers," in *Proc. of the IEEE INFOCOM 2017-IEEE Conf. on Computer Communications*, Atlanta, GA, USA, pp. 1–9, 2017.

[5] X. Lin, J. Li, J. Wu, H. Liang and W. Yang, "Making knowledge tradable in edge-AI enabled IoT: A consortium blockchain-based efficient and incentive approach," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6367–6378, 2019.

[6] P. Banerjee and S. Ruj, "Blockchain enabled data marketplace – design and challenges," *arXiv*, vol. 1811.11462, 2018. [online]. Available: https://arxiv.org/abs/1811.11462

[7] J. Christidis, P. A. Karkazis, P. Papadopoulos and H. C. Leligou, "Decentralized blockchain-based IoT data marketplaces," *Journal of Sensor and Actuator Networks*, vol. 11, no. 3, pp. 1–18, 2022.

[8] N. Asokan, M. Schunter and M. Waidner, "Optimistic protocols for fair exchange," in *Proc. of the 4th ACM Conf. on Computer and Communications Security*, Zurich, Switzerland, pp. 7–17, 1997.

[9] T. Coffey and P. Saidha, “Non-repudiation with mandatory proof of receipt,” *ACM SIGCOMM Computer Communication Review*, vol. 26, no. 1, pp. 6–17, 1996.

[10] J. Zhou and D. Gollman, “A fair non-repudiation protocol,” in *Proc. of the IEEE Symp. on Security and Privacy*, Oakland, CA, USA, pp. 55–61, 1996.

[11] S. Underwood, “Blockchain beyond bitcoin,” *Communications of the ACM*, vol. 59, no. 11, pp. 15–17, 2016.

[12] S. Ruoti, B. Kaiser, A. Yerukhimovich, J. Clark and R. Cunningham, “Blockchain technology: What is it good for?” *Communications of the ACM*, vol. 63, no. 1, pp. 46–53, 2020.

[13] F. Chen, J. Wang, C. Jiang, T. Xiang and Y. Yang, “Blockchain based non-repudiable IoT data trading: Simpler, faster, and cheaper,” in *Proc. of the IEEE INFOCOM 2022 – IEEE Conf. of Computer Communications*, London, United Kingdom, pp. 1958–1967, 2022.

[14] M. Jeon and S. Shin, “Decntralized fair data trading with random arbitrator node,” in *Proc. of the 23rd World Conf. on Information Security Applications*, Jeju, South Korea, pp. 1–5, 2022.

[15] G. Ateniese, D. Francati, D. Nuñez and D. Venturi, “Match me if you can: Matchmaking encryption and its applications,” *Advances in Cryptology – CRYPTO 2019, Lecture Notes in Computer Science*, vol. 11693, pp. 701–731, 2019.

[16] R. L. Rivest, “All-or-nothing encryption and the package transform,” in *Proc. of Fast Software Encryption – FSE 1997*, Haifa, Israel, pp. 210–218, 1997.

[17] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” *Advances in Cryptology-EUROCRYPT 2005, Lecture Notes in Computer Science*, vol. 3494, pp. 457–473, 2005.

[18] S. Xu, J. Ning, Y. Li, Y. Zhabng, G. Xu *et al.,* “Match in my way: Fine-grained bilateral access control for secure cloud-fog computing,” *IEEE Transactions on Dependable Secure Computing*, vol. 19, no. 2, pp. 1064–1077, 2022.

[19] I. Damgård, H. Haagh and C. Orlandi, “Access control encryption: Enforcing information flow with cryptography,” *Theory of Cryptography Conference, Lecture Notes in Computer Science*, vol. 9986, pp. 547–576, 2016.

[20] B. Chen, T. Xiang, M. Ma, D. He and X. Liao, “CL-ME: Efficient certificateless matchmaking encryption for internet of things,” *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 15010–15023, 2021.

[21] H. Niavis, N. Papadis, V. Reddy, H. Rao and L. Tassiulas, “A blockchain-based decentralized data sharing infrastructure for off-grid networking,” in *Proc. of the 2020 IEEE Int. Conf. on Blockchain and Cryptocurrency (ICBC)*, Toronto, ON, Canada, pp. 1–5, 2020.

[22] G. S. Ramachandran, R. Radhakrishnan and B. Krishnamachari, “Towards a decentralized data marketplace for smart cities,” in *Proc. of the 2018 IEEE Int. Smart Cities Conf. (ISC2)*, Kansas City, MO, USA, pp. 1–8, 2018.

[23] Y. Xu, P. Ahokangas, S. Yrjölä and T. Koivumaki, “The fifth archetype of electricity market: The blockchain marketplace,” *Wireless Networks*, vol. 27, pp. 4247–4263, 2021.

[24] Y. Xu, J. Ren, G. Wang, C. Zhang, J. Yang *et al.,* “A blockchain-based nonrepudiation network computing service scheme for industrial IoT,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3632–3641, 2019.

[25] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan *et al.,* “Blockchain for secure and efficient data sharing in vehicular edge computing and networks,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4660–4670, 2018.

[26] A. Dixit, A. Singh, Y. Rahulamathavan and M. Rajarajan, “FAST DATA: A fair, secure and trusted decentralized IIoT data marketplace enabled by blockchain,” *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 2934–2944, 2023.

[27] W. Dai, C. Dai, K. K. R. Choo, C. Cui, D. Zou *et al.,* “SDTE: A secure blockchain-based data trading ecosystem,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 725–737, 2020.

[28] Y. Li, L. Li, Y. Zhao, N. Guizani, Y. Yu *et al.,* “Toward decentralized fair data trading based on blockchain,” *IEEE Network*, vol. 35, no. 1, pp. 304–310, 2020.

[29] B. Poettering and D. Stebila, “Double authentication preventing signatures,” *International Journal of Information Security*, vol. 16, no. 1, pp. 1–22, 2017.

[30] A. Alsharif and M. Nabil, “A blockchain-based medical data marketplace with trustless fair exchange and access control,” in *Proc. of the GLOBECOM 2020–2020 IEEE Global Communications Conf.*, Taipei, Taiwan, pp. 1–6, 2020.

[31] J. Bethencourt, A. Sahai and B. Waters, “Ciphertext-policy attribute-based encryption,” in *Proc. of 2007 IEEE Symp. on Security and Privacy (SP’07)*, Berkeley, CA, USA, pp. 321–334, 2007.

[32] E. Ben-Sasson, A. Chiesa, E. Tromer and M. Virza, “Succinct non-interactive arguments for a von Neumann architecture,” in *Proc. of the 23rd USENIX Conf. on Security Symp.*, Berkeley, CA, USA, pp. 781–796, 2014.

[33] Y. -N. Li, X. Feng, J. Xie, H. Feng, Z. Guan *et al.,* “A decentralized and secure blockchain platform for open fair data trading,” *Concurrency and Computation: Practice and Experience*, vol. 32, no. 7, pp. e5578, 2020.

[34] S. Ma, Y. Mu and W. Susilo, “A generic scheme of plaintext-checkable database encryption,” *Information Sciences*, vol. 429, pp. 88–101, 2018.

[35] MIRACL Cryptographic SDK, Available: https://github.com/miracl/MIRACL

[36] G. Wood, “Ethereum: A secure decentralized generalized transaction ledger,” Berlin Version, 2022. [online] Available: https://ethereum.github.io/yellowpaper/paper.pdf