Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2024.052378

ARTICLE

Cross-Domain Bilateral Access Control on Blockchain-Cloud Based Data Trading System

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

[1]Electronics and Information Communications Research Center, Pukyong National University, Busan, 48513, Republic of Korea

[2]Department of Information Security, Graduate School, Pukyong National University, Busan, 48513, Republic of Korea

[3]Division of Computer Engineering, Pukyong National University, Busan, 48513, Republic of Korea

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

Received: 31 March 2024 Accepted: 16 July 2024 Published: 20 August 2024

ABSTRACT

Data trading enables data owners and data requesters to sell and purchase data. With the emergence of blockchain technology, research on blockchain-based data trading systems is receiving a lot of attention. Particularly, to reduce the on-chain storage cost, a novel paradigm of blockchain and cloud fusion has been widely considered as a promising data trading platform. Moreover, the fact that data can be used for commercial purposes will encourage users and organizations from various fields to participate in the data marketplace. In the data marketplace, it is a challenge how to trade the data securely outsourced to the external cloud in a way that restricts access to the data only to authorized users across multiple domains. In this paper, we propose a cross-domain bilateral access control protocol for blockchain-cloud based data trading systems. We consider a system model that consists of domain authorities, data senders, data receivers, a blockchain layer, and a cloud provider. The proposed protocol enables access control and source identification of the outsourced data by leveraging identity-based cryptographic techniques. In the proposed protocol, the outsourced data of the sender is encrypted under the target receiver's identity, and the cloud provider performs policy-match verification on the authorization tags of the sender and receiver generated by the identity-based signature scheme. Therefore, data trading can be achieved only if the identities of the data sender and receiver simultaneously meet the policies specified by each other. To demonstrate efficiency, we evaluate the performance of the proposed protocol and compare it with existing studies.

KEYWORDS

Bilateral access control; blockchain; data sharing; policy-match

1 Introduction

The development of the Internet-of-Things (IoT) infrastructure and devices has led to the generation and collection of IoT data at an explosive rate. These data have become valuable assets in the era of the data economy. Data trading enables data owners and data requesters to sell and buy data. Data generated by IoT technologies has significant value for data owners seeking economic benefits and data requesters developing data-intensive applications. This trend necessitates a data marketplace

that enables data trading between the data owner (sender) and the requester (receiver) in a reliable and efficient manner.

Blockchain is a distributed tamper-resistant ledger with verifiable state updates [1,2] that can serve as a transparent and reliable data trading controller. It is believed that blockchain has the advantage of achieving reliable data trading without relying on a trusted third party. In blockchain-cloud based data trading systems, the blockchain can be used to track the actions performed by senders and receivers, and the cloud can provide a way to store and access vast amounts of data. In addition, smart contracts on blockchain can facilitate automatic payments in digital currency when predefined conditions are satisfied. The mediation of smart contracts enables the enforcement of consent-based access control over data and records data trading instances as provenance evidence.

IoT is made up of technologies in various domains and application areas. Organizations or developers of these domains may wish to participate in the IoT data marketplace. For example, as shown in Fig. 1, cloud providers, technology companies, research institutes, and application developers can form a consortium to manage a blockchain-based data trading platform [3]. Then, members of these organizations can share and trade their data on the platform. In this system model, it is essential to achieve access control and source identification for the data managed by the external cloud. While the data sender aims to specify policies on who can access the data entrusted to the external cloud, the data receiver aims to specify attributes for the specific data senders from whom the receiver wants to purchase data. This can be achieved by bilateral access control, where both the sender and receiver must meet each other's policies. In addition, it is necessary to ensure accountability by providing a transparent process for transactions to prevent misbehavior.

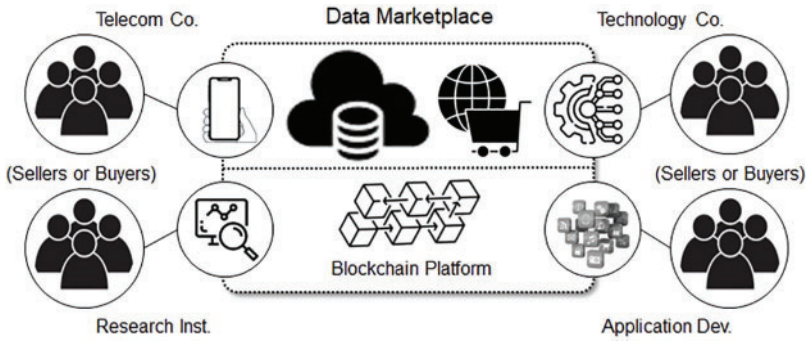

Figure 1: Concept of blockchain-cloud fusion IoT data marketplace

Recently, matchmaking encryption (ME) has been studied [4]. ME is a novel type of encryption that enables the sender to specify a receiver who can decrypt the message and the receiver to verify if the received message is from the intended sender [5]. Due to the functionality of ME, some recent studies have proposed bilateral access control schemes based on ME for secure data trading [6–8]. The authors of [8] proposed a secure data trading system with bilateral authorization using the identity-based ME (IBME) scheme [4]. They consider that the sender and receiver are under the authorization of the same domain authority responsible for the key generation center (KGC). Therefore, their IBME-based system is not suitable for a multiple domain environment because it requires another globally trusted KGC to issue identity-based keys to the sender and the receiver who belong to different organizations. As an alternative to this problem, a cross-domain IBME (cd-IBME) [9] may be employed in the data trading system. The authors of [7] proposed a secure data trading framework based on the cd-IBME. There is no doubt that IBME is useful for secure communication because no information is revealed

unless the target identities (i.e., access policies) specified by the sender and receiver match the respective counterparts. However, the policy-match is verified by the receiver after the final IBME decryption, which puts a heavy computational burden on the receiver. In [8], the authors proposed a data-sharing scheme for cloud services based on IBME (named IBME-DS) which delegates the policy-matching process to the cloud. However, their scheme works in a single-domain environment and still puts lots of computations on the receiver.

Based on the above considerations, we propose a cross-domain bilateral access control protocol for a blockchain-cloud based data trading system. Table 1 summarizes the features of the proposed protocol compared to some related work. In this paper, bilateral access control is to verify if the target identities specified by the sender and receiver satisfy the access policy of the other party. It ensures that only the designated receiver can decrypt the sender's encrypted data. The proposed protocol leverages identity-based encryption (IBE) [10] and the identity-based signature (IBS) [11] instead of using the IBME to design a more efficient protocol. The proposed policy-match procedure is based on the idea that the sender and receiver generate their authorization tags using the IBS and the cloud provider verifies whether the tags are simultaneously valid under the target identities of the sender and receiver when the receiver requests the sender's data stored in the cloud. In addition, to restrict access to the sender's data to the designated receiver, the sender's outsourced data is encrypted under the target receiver's identity by using the IBE. The IBE of [10] is practical in a multiple-domain environment because each domain can set up its own domain KGC which generates identity-based keys only for the users within the domain. The IBS of [11] enables batch verification that verifies multiple signatures efficiently at once. Hence, two authorization tags from the sender and receiver can be simultaneously verified by the cloud provider in an efficient manner. The contributions of this paper are summarized as follows:

- For cross-domain bilateral access control in a blockchain-cloud based data trading system, we present a system architecture that consists of domain authorities, data senders, data receivers, a blockchain layer, and a cloud provider.
- By using IBE and IBS, we propose a bilateral access control protocol that restricts data sharing and trading to only the sender and receiver who satisfy each other's policy in the system.
- We devise a policy-match procedure with IBS-based authorization tags of the sender and receiver. To reduce the computational overhead on the receiver, the cloud provider performs policy-match verification on the authorization tags to provide only the matched data to the receiver.
- To demonstrate the efficiency of the system, we evaluate the performance of the proposed protocol and compare it with existing IBME-based studies.

The rest of this paper is organized as follows: Section 2 briefly introduces related work on blockchain-cloud based data trading system models. Section 3 outlines the IBE and the IBS which serve as cryptographic building blocks of the proposed protocol. The system architecture is presented in Section 4, and the bilateral access control protocol in this system architecture is proposed in Section 5. Security and performance of the proposed protocol are evaluated in Section 6. Finally, Section 7 concludes this paper.

Table 1: Comparison of bilateral access control protocols in data sharing systems

	System model	Cross-domain	Crypto schemes	Match verification	Computation on user
[6]	Blockchain-cloud	X	IBME	By receiver	High
[7]	Blockchain-cloud	O	cd-IBME	By receiver	High
[8]	Cloud	X	IBME-DS	By cloud provider	High
Proposed	Blockchain-cloud	O	IBE, IBS	By cloud provider	Low

2 Related Work

Blockchain is a decentralized immutable ledger technology where the records on the blockchain are kept by a reliable and transparent way through a consensus mechanism. The types of blockchain can be broadly categorized into permissionless blockchain and permissioned blockchain [12]. Permissionless blockchain is a public and decentralized blockchain where any peer can participate in the consensus process. So, transactions transferred to the blockchain network can be read by all peers. Bitcoin [13] and Ethereum [14] are typical instances. However, due to privacy concerns, it is not advisable to implement data business solutions on a public blockchain. On the other hand, in a permissioned blockchain, only a limited set of authorized peers can join the blockchain network. The widely known instances of permissioned blockchain are Hyperledger Fabric [15] and Corda [16]. A consortium blockchain is a kind of permissioned blockchain collaboratively managed by multiple organizations. A consortium blockchain provides a method to secure the interactions among organizations that have a common goal.

Due to the immutability and auditability of the blockchain, accompanied by smart contracts, the blockchain is evolving into a platform to develop a decentralized applications in various fields. It is regarded that secure and reliable data trading can be achieved on the blockchain. Recently, research on blockchain-based data trading models has received a great deal of attention, and several system models have been introduced. The existing data trading system models can be roughly classified into on-chain models and on/off-chain models. The data is directly shared or traded via blockchain in the on-chain models [17–19]. Such on-chain models can find practical applications when the data volume is small. However, managing data in on-chain data trading models becomes ineffective due to the continuously growing data volume in IoT environments.

To reduce the on-chain storage cost, a hybrid model of on-chain and off-chain is considered. In this model, an off-chain storage service such as the cloud is employed to host a huge volume of data in the on/off-chain model [3,6,20,21]. Therefore, a novel paradigm of blockchain and cloud fusion has been widely considered as a promising data trading platform. Moreover, the data is usually encrypted before being outsourced to the blockchain or cloud storage. For secure sharing and trading of data, the data owner can specify access policies and manage decryption rights to the data. In addition to the on-chain and off-chain models, Wang et al. introduced the concept of an off-state system model for big data sharing [22]. They discussed some issues in off-chain based schemes built on a public blockchain and designed an off-state data-sharing protocol based on a permissioned blockchain which addresses the security and autonomy issues in off-chain data sharing schemes.

With regard to fair data trading, Li et al. proposed a decentralized data trading system using blockchain to guarantee fair data transactions with authentication [23]. For a digital data marketplace

based on the blockchain, Dixit et al. proposed a decentralized platform that hosts data in a reliable and fault-tolerant manner [24]. Chen et al. proposed a blockchain-based non-repudiable IoT data trading system [25], in which the blockchain records data trading behaviors of data sellers and buyers to facilitate on-chain and off-chain arbitration. However, their system does not consider secure data trading with access control for data confidentiality.

For secure data trading, Alsharif et al. proposed a medical data marketplace based on the blockchain [26]. They applied the ciphertext-policy ABE scheme [27] in order for the seller to enforce access control policies on the encrypted records. In [28], Li et al. proposed a secure blockchain platform for fair data trading by using the plaintext checkable encryption scheme [29]. However, Alsharif et al.'s and Li et al.'s systems only focused on the access control by the seller for data confidentiality and burdened the on-chain procedures with complex cryptographic computations.

There are some studies on cross-domain secure data sharing for industrial IoT [30–32]. Sing et al. proposed a centralized cloud-based cross-domain data-sharing platform using multiple security gateways that use the blockchain to store the information in the cloud [30]. Yu et al. proposed a consortium blockchain-based cross-domain industrial IoT data-sharing mechanism [31]. The authors introduced a consortium blockchain to construct trust among different domains, and proxy re-encryption and group signature schemes for secure cross-domain data sharing and privacy-preservation of end devices. In [32], Zheng et al. also proposed cross-domain data sharing by deploying permissioned blockchain. The authors developed a key agreement protocol and zero-knowledge proof to verify data ownership under the criterion of confidence and anonymity.

3 Cryptographic Building Blocks

We briefly outline the IBE in multiple KGC environment of Wang et al. [10] and the IBS of Cha et al. [11] based on bilinear map.

3.1 Bilinear Map

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two groups of the same prime order q and P be a generator of $\mathbb{G}$. A bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ satisfies the following properties:

- Bilinearity: $e(P^a, P^b) = e(P, P)^{ab}$ for all $a, b \in Z_q$.
- Non-degenerate: $e(P, P) \neq 1_{\mathbb{G}_T}$.
- Computable: It is efficient to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}$.

3.2 Wang-Cao Identity-Based Encryption

Wang and Cao's IBE is constructed as follows:

1. G-Setup: It sets the global public parameter $param = \langle \mathbb{G}, \mathbb{G}_T, q, e, P, H_1, H_2 \rangle$, where H_1: $\{0,1\}^* \to \mathbb{G}$ and H_2: $\mathbb{G}_T \to \{0,1\}^n$ are hash functions.
2. Setup: It chooses a random $\alpha \in Z_q$ and computes $A = P^\alpha \in \mathbb{G}$, and sets the master secret key $msk = \alpha$ and the master public key $mpk = A$.
3. KeyGen: Given an identity $id \in \{0,1\}^*$, it computes $Q_{id} = H_1(id) \in \mathbb{G}$ and generates the private key $dk_{id} = Q_{id}^{1/\alpha}$, where α is the msk.
4. Encrypt: To encrypt a message $m \in \{0,1\}^n$ under an identity id, it chooses a random $r \in Z_q$, then sets the ciphertext $C = \langle C_1, C_2 \rangle$ as
$$Q_{id} = H_1(id) \tag{1}$$

$$\omega = e(P, Q_{id}) \tag{2}$$

$$C_1 = A^r \tag{3}$$

$$C_2 = m \oplus H_2(\omega^r). \tag{4}$$

5. Decrypt: On input a ciphertext $C = \langle C_1, C_2 \rangle$ and the private key dk_{id}, it outputs the decrypted message m as

$$m = C_2 \oplus H_2(e(C_1, dk_{id})). \tag{5}$$

3.3 Cha-Cheon Identity-Based Signature

Cha and Cheon's IBS consists of the following algorithms:

1. Setup: Given the system parameter $\langle \mathbb{G}, \mathbb{G}_T, q, e, P, H_1, H_3 \rangle$, where $H_3 \colon \{0,1\}^* \times \mathbb{G} \to Z_q$ is a hash function, it picks a random $\beta \in Z_q$ as the master secret key ($msk = \beta$) and computes $B = P^\beta \in \mathbb{G}$ as the master public key ($mpk = P^\beta$).
2. KeyGen: Given an identity id, it computes $Q_{id} = H_1(id)$ and generates the secret signing key $sk_{id} = Q_{id}^\beta$ of the id, where β is the msk.
3. Sign: To sign a message m under the secret key sk_{id}, it picks a random $x \in Z_q$ and outputs the signature $\sigma = \langle \sigma_1, \sigma_2 \rangle$ as

$$\sigma_1 = H_1(id)^x \tag{6}$$

$$h = H_3(m, \sigma_1) \tag{7}$$

$$\sigma_2 = sk_{id}^{x+h}. \tag{8}$$

4. Verify: To verify a signature $\sigma = \langle \sigma_1, \sigma_2 \rangle$ of a message m under the identity id, it computes $Q_{id} = H_1(id)$ and $h = H_3(m, \sigma_1)$, then checks

$$e(P, \sigma_2) \stackrel{?}{=} e(B, \sigma_1 Q_{id}^h). \tag{9}$$

If it holds, the signature σ of m is accepted.

4 System Model

4.1 System Architecture

We consider the data trading architecture as shown in Fig. 2 which consists of supervising authority, data sender (owner), data receiver (requester), blockchain layer, and cloud provider.

- Supervisors consist of a set of domain authorities (DA) which cooperate to manage the data marketplace. For instance, a group of organizations that want to participate in the data marketplace may form a consortium and collaborate in managing the data trading platform based on the blockchain. DA is responsible for setting up the public system parameters for bilateral access control. Each domain authority acts as a KGC that issues identity-based private keys to senders and receivers belonging to it. In addition, if a dispute occurs, DA forms a committee to identify the dishonest party by examining the evidence recorded on the blockchain.
- Data sender (S) offers its own data through the system. To ensure that only an authorized receiver can access the data, S sets the receiver policy which specifies the allowed identity of the receiver, and encrypts the data under the target receiver identity. S obtains its identity-based private key from the domain authority to which S belongs. When S outsources the data to the

cloud, *S* attaches its authorization tag generated by the private key and the target receiver's identity.

- Data receiver (*R*) requests the data when *R* finds interesting data provided by the desirable sender *S* satisfying the sender policy. *R* obtains its identity-based private keys from the domain authority to which *R* belongs. When requesting the data, *R* presents its authorization tag generated by the private key and the target sender's identity to the cloud. *R* can obtain the data if and only if the encrypted data is associated with its identity which corresponds to the receiver policy of *S*.
- Blockchain layer serves as a decentralized and trusted platform that enforces data trading rules, verifiably updates data trading state, and facilitates payments using digital currency. Blockchain is a ledger that records the state of data trading between the sender and the receiver. Smart contracts on the blockchain execute the business logic of data trading and define the terms and actions necessary for the involved parties to carry out specific aspects of data trading transactions.
- The cloud provider (*CP*) serves storage to host a huge volume of data shared by the customers. For bilateral access control, *CP* runs the policy-match verification procedure for the authorization tags presented by the sender and the receiver. When successful match occurs, *CP* allows the receiver to access the sender's data.

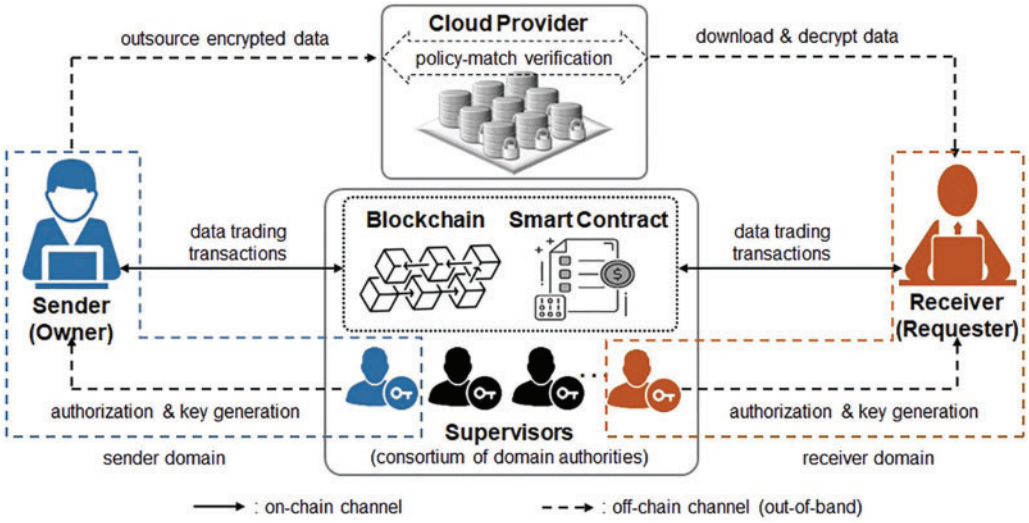

Figure 2: Overview of the system architecture for the proposed protocol

4.2 Threats and Design Goals

*DA*s establish a trust relationship among themselves as consortium members, while the sender and receiver only trust their respective affiliated authorities. Each *DA* advocates for the rights and interests of internal users, so *DA* will not collude with any external user. *CP* is semi-trusted entity that carries out the protocol honestly but may be curious about the data contents, which makes users anxious about data confidentiality of their data on the cloud. When a sender outsources its data to cloud storage, data confidentiality requires that *CP* do not learn any information about the data, while receivers designated by the sender can access the data. A malicious entity may impersonate the receiver or the sender to gain illegal access to data or mislead the receiver. Under the threat model, we consider the following security goals.

- Policy-match: For bilateral access control, the sender and receiver can specify their own access policies that the other party must satisfy. The data trading is achieved only if their identities simultaneously meet the policies specified by the respective counterparts.
- Privacy: The outsourced data must be protected from being revealed to the cloud provider. In addition, the identities of the sender and receiver must be hidden from the cloud provider even in the policy-match verification.
- Accountability: The system can supervise and keep track of transactions processed by the sender and receiver to prevent misconduct.

The proposed protocol aims to ensure the above security goals on the data trading system so that the receiver specified by the sender obtains the correct data from the sender specified by the receiver. Here, the correct data means the same encrypted data as the sender commits to the blockchain. However, this paper does not consider cheating by a dishonest party attempting to gain unfair financial profits. Such a problem can be addressed by fair exchange protocols with incentive and penalty techniques [33–35], and arbitration protocols can be introduced to resolve disputes [36].

5 Proposed Protocol

The proposed cross-domain bilateral access control protocol for the data trading system is presented in this section. Table 2 describes the notations used in the protocol.

Table 2: Notations used in the protocol

Notation	Description
DA	Set of domain authorities, $DA = \{DA_1, DA_2, ..., DA_n\}$
F	Data file of the sender for marketing
CP	Cloud storage provider
SC	Smart contract for the data trading
$\mathbb{G}, \mathbb{G}_T$	Bilinear map groups of prime order q
$e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$	Bilinear pairing
$P \in \mathbb{G}$	Generator of $\mathbb{G}$
$H_1 : \{0,1\}^* \to \mathbb{G}$	Hash function
$H_2 : \mathbb{G}_T \to \{0,1\}^n$	Hash function
$H_3 : \{0,1\}^* \times \mathbb{G} \to Z_q$	Hash function
$\alpha_i, \beta_i \in Z_q$	Master secret keys of DA_i
$A_i = P^{\alpha_i}, B_i = P^{\beta_i} \in \mathbb{G}$	Master public keys of DA_i
dk_{id}, sk_{id}	Private keys for an identity id for IBE and IBS
$Enc_K(X)$	Symmetric encryption of the input X under the key K
$Dec_K(Y)$	Symmetric decryption of the input Y under the key K
$Atag_{id}$	Authorization tag of the id for policy-match
ind_{tx}	Index of the referenced transaction tx
$X \stackrel{?}{=} Y$	Operation to check if $X = Y$ is true or false

5.1 Overview

The basic idea of the proposed bilateral access control protocol is twofold. One is that the sender's data entrusted to the cloud is encrypted under the target receiver's identity by using the IBE to allow only the designated receiver to decrypt the data. The other is that the cloud provider checks if the authorization tags based on the IBS of the sender and receiver are valid under the target identities specified by the other party at the same time.

In the proposed system, data senders and receivers who want to participate in the data trading system should obtain their identity-based private keys by their domain authorities, respectively. The sender encrypts its data under the target receiver's identity, and generates its IBS-based authorization tag to prove that the sender is an authorized entity of the specified identity. Then, the sender registers the encrypted data along with the authorization tag to the cloud. The receiver also generates its authorization tag, and presents the tag to the cloud provider when requesting the sender's data. The cloud provider performs policy-match verification on the authorization tags of the sender and receiver. If the tags are valid under the specified sender's and receiver's identities at the same time, the cloud provider allows the receiver to access the sender's encrypted data. Therefore, the receiver can decrypt the actual data if the receiver is the authorized holder of the target identity specified by the sender.

5.2 Setup

The consortium of domain authorities DA establishes an efficient blockchain as a transparent and reliable controller for data trading, and agrees on the global public system parameters $\langle \mathbb{G}_1, \mathbb{G}_2, q, e, P, H_1, H_2, H_3 \rangle$. Then, each $DA_i \in DA$ randomly chooses $\alpha_i, \beta_i \in Z_q$ and computes $A_i = P^{\alpha_i}$ and $B_i = P^{\beta_i}$. DA_i sets its master secret key and master public key pair as $msk_i = (\alpha, \beta)$ and $mpk_i = (A_i, B_i)$ to issue id-based keys to the users authorized by it. At this phase, we assume that the public parameters are known to the system and the smart contract to control the data trading is deployed on the blockchain.

5.3 Data Trading with Bilateral Access Control

Suppose that a sender S provides a data F and a requester R purchases the data through the data trading system. S and R perform the data trading protocol with the mediation of the smart contract (SC) on the blockchain. For simplicity, we assume that S and R are under the authorization of $DA_S \in DA$ and $DA_R \in DA$, and denote the target identities (i.e., access policies) specified by the sender and the receiver as rcv and snd, respectively. Note that identities can be represented by group membership, roles, or other attributes associated with the entity.

5.3.1 Data Registration

To participate in the data trading, S must obtain its identity-based key issued by DA_S which will be used to generate an authorization tag for policy-match. Given the identity id_S, DA_S issues S with the private signing key $sk_S = H_1(id_S)^{\beta_S}$ generated by DA_S's master secret key. S prepares and outsources the encrypted data file package EF to the cloud storage. Hence, the data file F is first encrypted by using a symmetric key encryption and then the key is encrypted by using the IBE for the target receiver of rcv as follows:

1. choose a random secret key K and encrypt the file F as $E \leftarrow Enc_K(F)$
2. compute $Q'_R = H_1(rcv)$ and $\omega = e(P, Q'_R)$
3. choose a random $r \in Z_q$ and output $C_2 = K \oplus H_2(\omega^r)$.

Note that C_2 is the partial ciphertext of the IBE scheme [10] presented in Section 3.2. The computation of $\omega^r = e(P, Q'_R)^r$ is independent of any DA_i's public key A_i. Therefore, at this phase, S can pre-compute C_2 under the target identity rcv without knowing which domain the receiver who will request the decryption key is affiliated with. Particularly, this feature is useful in a multi-receiver scenario, where the receivers exist across multiple domains. S only needs to compute single C_2 no matter how many domains the receivers are from.

In addition to the encryption of data, S generates its authorization tag $Atag_S$ by using the IBS as follows:

1. pick a random $x \in Z_q$ and compute $Q_S = H_1(id_S)$, $S_1 = Q_S^x$, and $S_2 = sk_S^{x+h}$, where $h = H_3(Q_S|Q'_R, S_1)$
2. output $Atag_S = \sigma_S|Q'_R$, where $\sigma_S = \langle S_1, S_2 \rangle$.

S forms the encrypted file package as $EF = \langle E, C_2, Atag_S \rangle$ and computes $\Delta_F = H(F)$, $\Delta_K = H(K)$, and $\Delta_{EF} = H(EF)$, where $H()$ is a cryptographic hash function for message digest. S invokes SC to register the proof of the data by submitting the transaction $reg := [desc_F, prc_F, \Delta_F, \Delta_K, \Delta_{EF}]$, where $desc_F$ is short description about the data and prc_F is the price of the data. When reg is recorded on the blockchain, S outsources $[EF, ind_{reg}]$ to the cloud, where ind_{reg} is the index of the referenced reg on the blockchain.

5.3.2 Data Request and Policy-Match Verification

When R is interested in the data registered by reg, R must obtain its identity-based keys issued by DA_R to participate in the data trading. Given the identity id_R, DA_R issues R with the private keys $dk_R = H_1(id_R)^{1/\alpha_R}$ and $sk_R = H_1(id_R)^{\beta_R}$ generated by DA_R's master secret keys. To request access to EF coupled with ind_{reg}, R first generates its authorization tag $Atag_R$ as follows:

1. compute $Q_R = H_1(id_R)$ and $Q'_S = H_1(snd)$, where snd is the identity of a target sender
2. pick a random $y \in Z_q$ and compute $R_1 = Q_R^y$ and $R_2 = sk_R^{y+l}$, where $l = H_3(Q'_S|Q_R, R_1)$
3. output $Atag_R = \sigma_R|Q'_S$, where $\sigma_R = \langle R_1, R_2 \rangle$.

R presents $Atag_R$ to CP, then CP runs policy-match verification that checks if the identities (i.e., id_S and id_R) match the access policies (i.e., snd and rcv) of each other. By taking $(Atag_S, B_S, Atag_R, B_R)$ as input, where B_S and B_R are public keys of DA_S and DA_R, respectively, the policy-match verification works as follows:

1. parse the tags as $\sigma_S = \langle S_1, S_2 \rangle |Q'_R \leftarrow Atag_S$ and $\sigma_R = \langle R_1, R_2 \rangle |Q'_S \leftarrow Atag_R$
2. compute $h = H_3(Q'_S|Q'_R, S_1)$ and $l = H_3(Q'_S|Q'_R, R_1)$
3. check $e(P, S_2R_2) \stackrel{?}{=} e(B_S, S_1 Q_S'^h) e(B_R, R_1 Q_R'^l)$.

If S and R satisfy the policies, that is, $id_S = snd$ and $id_R = rcv$, hence $Q_S = Q'_S$ and $Q_R = Q'_R$ are valid, then match occurs and CP allows R to access the package EF. The correctness of the policy-match verification will be discussed in Section 6.1.

5.3.3 Order and Offer Decryption Key

When $Atag_R$ passes the policy-match verification, R can download the encrypted file package EF from the cloud storage. To verify the integrity of the downloaded EF, R takes Δ_{EF} from reg on the blockchain and checks $\Delta_{EF} \stackrel{?}{=} H(EF)$. However, R cannot recover the actual data F yet without

knowing the key K encapsulated in C_2. Therefore, R needs to order the decryption key to S by using SC with payment. R invokes SC by submitting the transaction $ord := [ind_{reg}, pay_R, dep_R, exp_{ord}]$, where pay_R is the payment for the data and exp_{ord} is expiration time. Note that, after this order transaction, if the decryption key is not offered by S within exp_{ord} then SC will cancel this trading and return pay_R to R.

Upon receiving the order from R, S takes DA_R's public key A_R and computes the partial ciphertext $C_1 = A_R^r$, where r is the random value chosen in the data registration phase to compute C_2. S offers C_1 to R through SC by submitting the transaction $ofr := [ind_{ord}, C_1, dep_S, exp_{ofr}]$, where dep_S is a guarantee deposit that will be confiscated if S offers invalid decryption key.

5.3.4 Decryption and Confirmation

Once *ofr* is recorded on the blockchain, R retrieves C_1 from *ofr* and combines it with C_2 in EF. To extract the key K, R decrypts $\langle C_1, C_2\rangle$ under its private key dk_R as $K = C_2 \oplus H_2(e(C_1, dk_R))$ and checks $\Delta_K \stackrel{?}{=} H(K)$, where Δ_K is the digest recorded in *reg*. If it holds, R recovers the file as $F \leftarrow Dec_K(E)$. If R is a valid receiver specified by *rcv*, R can recover the key K and then the data F by computing the same ω^r generated by S. The correctness of computing the key K will be discussed in Section 6.2.

After decrypting the file, R verifies the authenticity of F by checking $\Delta_F \stackrel{?}{=} H(F)$, and if it holds, then invokes confirmation procedure. Upon receiving the confirmation call, SC transfers the escrowed pay_R in *ord* to S's account and also returns dep_S in *ofr* to S. Finally, the data trading between S and R is successfully completed. Therefore, if S and R are the authorized sender and receiver who meet the access policies specified by the other party and honestly follow the protocol, they can respectively receive the payment and the data.

6 Analysis

6.1 Policy-Match

Bilateral access control is that the identities of the sender and receiver must satisfy the policies specified by the other party simultaneously to accomplish data trading between them. In the proposed protocol, the sender S and receiver R are required to presents their authorization tags $Atag_S$ and $Atag_R$ to CP for policy-match verification. In practice, $\sigma_S = \langle S_1, S_2\rangle = \left\langle Q_S^x, sk_S^{x+h}\right\rangle$ in $Atag_S$ and $\sigma_R = \langle R_1, R_2\rangle = \left\langle Q_S^y, sk_R^{y+l}\right\rangle$ in $Atag_R$ are the signatures of S and R for the specified policies *snd* and *rcv* as the input message $H_1(snd)|H_1(rcv)$. The policy-match procedure of CP is batch verification of σ_S and σ_R [37], where both of these signatures must be valid under the target identities specified by the sender and receiver, respectively. Therefore, if S and R satisfy the policies of the other party (i.e., $id_S = snd \wedge id_R = rcv$), then they will pass the policy-match verification as shown in the below equations and be authorized as the intended parties:

$$e(P, S_2R_2) = e(P, S_2)e(P, R_2) \tag{10}$$

$$= e(P, sk_S^{x+h})e(P, sk_R^{y+l}) \tag{11}$$

$$= e(P, Q_S^{\beta_S(x+h)})e(P, Q_R^{\beta_R(y+l)}) \tag{12}$$

$$= e(P^{\beta_S}, Q_S^x Q_S^h)e(P^{\beta_R}, Q_R^y Q_R^l) \tag{13}$$

$$= e(B_S, S_1 Q_S^h)e(B_R, R_1 Q_R^l). \tag{14}$$

Moreover, these tags are based on Cha-Cheon's IBS scheme secure against existential forgery on a chosen message and identity attack according to the hardness of the computational Diffie-Hellman problem [11]. Only S and R, who obtained their identity-based private keys $sk_S = H_1(id_S)^{\beta_S}$ and $sk_R = H_1(id_R)^{\beta_R}$ from their domain authorities DA_S and DA_R, can generate valid authorization tags. Hence, it is hard to impersonate the sender and receiver to deceive the policy-match verification without knowing the private keys.

6.2 Privacy

Except the sender and receiver specified by the other party, no one can know the data contents and the identity information about the sender and receiver in the proposed protocol. For access control and data confidentiality, the data file F is encrypted by using a symmetric key encryption as $E \leftarrow Enc_K(F)$ and the key K is again encrypted by using the IBE scheme [10] under the target receiver identity as $C_1 = A_R^r$ and $C_2 = K \oplus H_2(\omega^r)$, where $\omega = e(P, H_1(rcv))$. Data decryption depends on the correctness of the key K resulting from the computation of ω^r as shown in the following equations:

$$e(C_1, dk_R) = e(A_R^r, H_1(id_R)^{1/\alpha_R}) \tag{15}$$

$$= e(P^{\alpha_R r}, H_1(id_R)^{1/\alpha_R}) \tag{16}$$

$$= e(P, H_1(id_R)^r) \tag{17}$$

$$= e(P, H_1(rcv)^r) \tag{18}$$

$$= \omega^r. \tag{19}$$

According to the security of Wang-Cao's IBE, only the authorized receiver R that possesses the identity-based private key $dk_R = H_1(id_R)^{1/\alpha_R}$ satisfying $id_S = rcv$ can decrypt the data $F \leftarrow Dec_K(E)$ after extracting the correct key $K = C_2 \oplus H_2(e(C_1, dk_R))$. On the other hand, it is infeasible that not only CP but also any malicious entity calculate R's private key dk_R to decrypt the data even though A_R, Q_R, and $\langle C_1, C_2 \rangle$ are known to the system.

Furthermore, CP may also need to know the identities of the target sender and receiver for policy-match verification. However, the identity information of S and R given to CP are hidden in $Q'_S = H_1(snd)$ and $Q'_R = H_1(rcv)$, respectively. Due to the one-way property of hash function, it is hard to infer the identities from Q'_S and Q'_R. CP is just able to run policy-match verification for $Atag_S$ and $Atag_R$ without knowing the identities of S and R. Therefore, identity privacy can be also preserved in the proposed protocol.

6.3 Accountability

Data trading is performed by means of smart contracts which define the valid state of data trading progress and transaction logic. All transactions processed by the smart contract at each phase are recorded on the tamper-resistant ledger on the blockchain. These records can be regarded as evidence of the behaviors performed by the sender and receiver throughout the data trading protocol.

The recorded transactions at each phase can be regarded as evidence of the behaviors taken by the participants during the data trading protocol. One goal of the proposed system is to ensure that the receiver obtains the correct data as same as the sender commits to the blockchain. S must submit reg containing the message digests Δ_{EF}, Δ_K, and Δ_F as a proof. Then, CP and R can verify the correctness of the encrypted package, decryption key, and data by checking the recorded digests. Furthermore,

when a dispute occurs, the consortium of domain authorities can examine the recorded transactions to determine whether the participants adhered to the rules and provided correct data.

6.4 Performance

In this section, we evaluate the performance of the proposed protocol in terms of the computational overhead of the sender, receiver, cloud provider, and key generation center. Table 3 shows the analysis of computational overhead and comparison with the existing approaches of IBME-based [6], cd-IBME-based [8], and IBME-DS [7] systems. In Table 3, T_p denotes bilinear pairing computation, T_{e_1} denotes exponentiation in $\mathbb{G}$, T_{e_2} denotes exponentiation in $\mathbb{G}_T$, T_{m_1} denotes multiplication in $\mathbb{G}$, and T_{m_2} denotes multiplication in $\mathbb{G}_T$. Note that IBME-based and IBME-DS systems employ a global single-domain KGC for bilateral access control while the proposed system considers multiple domain KGCs. Cloud provider is not involved in policy-match in IBME-based and cd-IBME-based systems.

Table 3: Computational overhead

	Sender	Receiver	Cloud Provider	KGC	
				DA_S	DA_R
IBME-based [6]	$2T_p + 3T_{e_1}$	$3T_p + 1T_{e_1}$	–	$3T_{e_1}$	
IBME-DS [8]	$4T_p + 8T_{e_1} + 2T_{m_1} + 2T_{m_2}$	$5T_p + 2T_{e_1} + 3T_{m_2}$	$2T_p + 1T_{m_1}$	$3T_{e_1}$	
cd-IBME-based [7]	$2T_p + 4T_{e_1} + 3T_{m_1} + 1T_{m_2}$	$5T_p + 3T_{m_2}$	–	$2T_{e_1}$	$2T_{e_1}$
Proposed	$1T_p + 3T_{e_1} + 1T_{e_2}$	$1T_p + 2T_{e_1}$	$3T_p + 2T_{e_1} + 2T_{e_2} + 1T_{m_2}$	$1T_{e_1}$	$2T_{e_1}$

In addition, we estimated the processing time by using the benchmark results of Java Pairing-Based Cryptography (JPBC) library [38] tested on Intel(R) Core(TM) Quad CPU Q6600 @ 2.4 GHz for the group $\mathbb{G}$ of 512-bit base field with 1024-bit security. Fig. 3 shows the processing time required by each entity. The policy-match process of [6,8] relies on IBME schemes which causes additional complexity to the end users due to the matchmaking aspect of IBME algorithms. However, to reduce the computation cost of the end users, the proposed protocol devises the authorization tag based on the IBS scheme and the policy-match verification on the tags is delegated to the cloud provider on behalf of the end users. Therefore, as shown in Fig. 3, the sender and receiver in the proposed protocol consume less processing time than other systems. On the other hand, the cloud provider in the proposed system takes more processing time than that of IBME-DS. Nevertheless, the cost would not be a significant burden for the cloud which is generally regarded as a powerful computing platform.

To demonstrate the efficiency on the receiver's side, we also estimated the processing time expected for the receiver and cloud provider when the receiver finds and decrypts policy-matched data among multiple data packages on the cloud. Let N_t be the total number of packages and N_m be the number of matched packages among N_t. Fig. 4 shows the processing time depending on N_t assuming $N_m = N_t \times 40\%$. In the IBME-based and cd-IBME-based systems, the receiver processes all N_t data packages to find policy-matched ones among them. On the other hand, in the proposed system and IBME-DS, the receiver only needs to decrypt N_m data filtered by the cloud provider that verifies N_t authorization tags. Consequently, the receiver does not need to check all data packages to find policy-matched ones because the cloud provides the receiver with only allowed data packages after policy-match verification. As shown in Fig. 4, we can observe that the proposed protocol can reduce the computational overhead on the receiver.

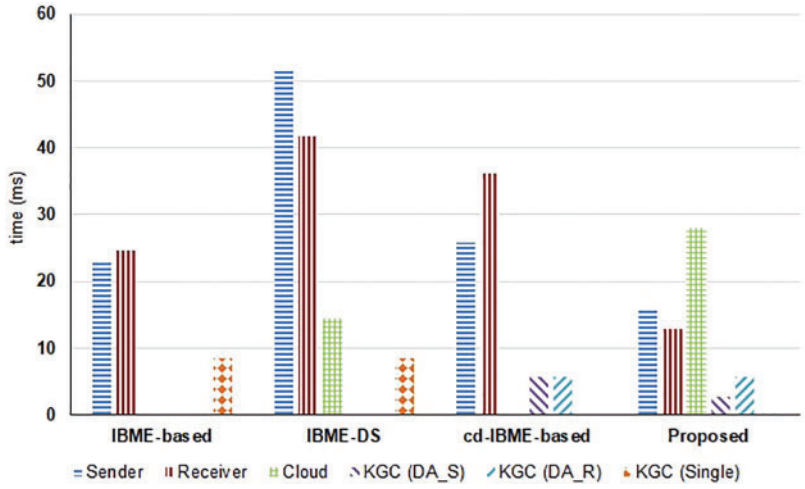

Figure 3: Processing time required by each entity

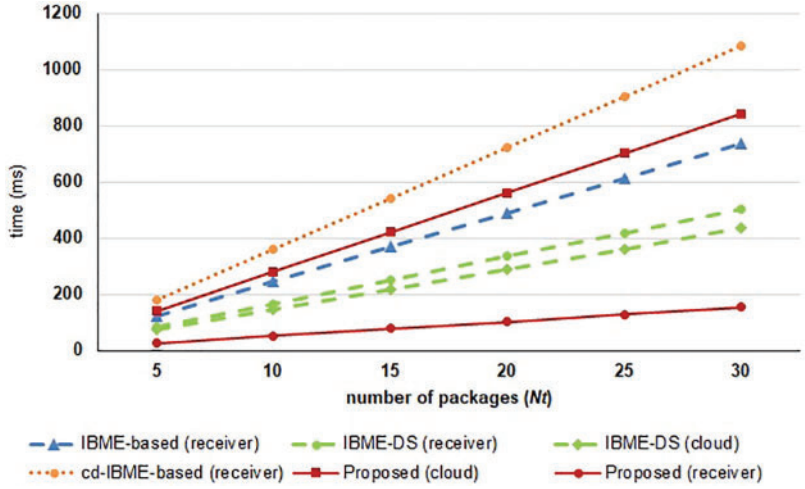

Figure 4: Processing time of the receiver and cloud provider to find and decrypt matched data

Another advantageous aspect of the proposed system is the efficiency on the sender's side in multi-receiver scenarios. Intuitively, to generate separately encrypted data for N receivers, the sender needs to perform N data encryptions. However, in the proposed protocol, the sender can generate N encrypted data in less time than N encryptions. We consider two scenarios to demonstrate the sender's efficiency in a multi-receiver environment. One scenario is for the receivers with different identities in a single domain (scen1) and the other is for the receivers with the same identity across multiple domains (scen2). In the proposed protocol, when the sender generates an encrypted data package, the partial ciphertext C_1 is computed on the public key of the receiver's side DA while the other part C_2 is computed on the identity of the target receiver. That is, scen1 only affects the computation of C_1, and scen2 only affects the computations of C_2. Therefore, the sender can compute single C_2 no matter how many domains there are in scen1, and compute single C_1 no matter how many receivers are across multiple domains in scen2. Fig. 5 shows the estimated processing time depending on the scenarios. We can observe that the proposed protocol outperforms others.

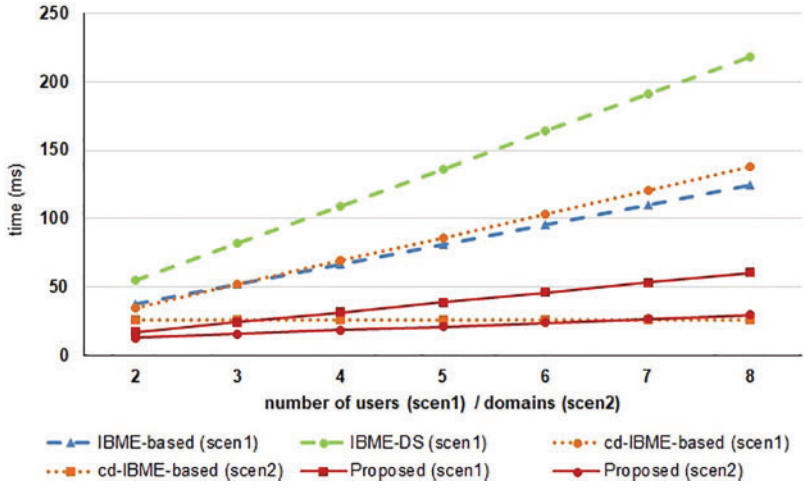

Figure 5: Processing time of the sender in multi-receiver scenarios

7 Conclusion

In a blockchain-cloud based data trading system, the blockchain can be used to keep track of behaviors performed by both the sender and receiver, while the cloud can provide a way to store and access a vast amount of data. However, it remains a challenge to securely trade the data outsourced to the external cloud in a way that restricts access to the data only to authorized users across multiple domains. Therefore, in this paper, we proposed a bilateral access control protocol by leveraging identity-based encryption and signature schemes. In the proposed protocol, the outsourced data of the sender is encrypted under the target receiver's identity, and the cloud provider runs policy-match verification on the authorization tags of the sender and receiver generated by an identity-based signature scheme. Therefore, the proposed protocol enables data trading is achieved only if the identities of the data sender and receiver simultaneously meet the policies specified by each other. In addition, we evaluated the performance of the proposed protocol and compared it with the existing IBME-based systems to demonstrate improved efficiency. In a multi-receiver environment, the performance of the proposed protocol is more affected by the number of identities rather than the number of domains. It remains future work to explore a bilateral access control protocol for multiple receivers with different identities.

Acknowledgement: The authors wish to express their appreciation to the reviewers for their helpful suggestions which greatly improved the presentation of this paper.

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 the MSIT (Ministry of Science and ICT), Korea, under the Special R&D Zone Development Project (R&D)—Development of R&D Innovation Valley Support Program (2023-DD-RD-0152) supervised by the Innovation Foundation.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Youngho Park, Sang Uk Shin; data collection: Su Jin Shin; analysis and interpretation of results: Youngho Park, Sang Uk Shin; draft manuscript preparation: Youngho Park. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

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

References

1. Underwood S. Blockchain beyond bitcoin. Commun ACM. 2016;59(11):15–7. doi:10.1145/2994581.
2. Ruoti S, Kaiser B, Yerukhimovich A, Clark J, Cunningham R. Blockchain technology: what is it good for? Commun ACM. 2020;63(1):46–53. doi:10.1145/3369752.
3. Liu D, Huang C, Ni J, Lin X, Shen XS. Blockchain-cloud transparent data marketing: consortium management and fairness. IEEE Trans Comput. 2022;71(12):3322–35. doi:10.1109/TC.2022.3150724.
4. Ateniese G, Francati D, Nunez D, Venturi D. Match me if you can: Matchmaking encryption and its applications. In: Advances in cryptology—CRYPTO 2019; 2019. vol. 11693, p. 701–31. doi:10.1007/978-3-030-26951-7_24.
5. Xu S, Ning J, Li Y, Zhang Y, Xu G, Huang X, et al. Match in my way: fine-grained bilateral access control for secure cloud-fog computing. IEEE Trans Dependable Secure Comput. 2022;19(2):1064–77. doi:10.1109/TDSC.2020.3001557.
6. Park Y, Jeon MH, Shin SU. Blockchain-based secure and fair IoT data trading system with bilateral authorization. Comput Mater Contin. 2023;79(2):1871–90. doi:10.32604/cmc.2023.039462.
7. Park Y, Shin SJ, Park YH, Shin SU. A cross-domain secure data trading framework based on blockchain-cloud fusion. In: Proceedings of the 7th International Conference on Mobile Internet Security, 2023; Okinawa, Japan; p. 70.
8. Wu T, Ma X, Yan H. Enabling privacy-preserving data sharing with bilateral access control for cloud. Electronics. 2023;12(23):4798. doi:10.3390/electronics12234798.
9. Wu A, Weng J, Luo W, Yang A, Liu J-N, Jiang Z. Cross-domain identity-based matchmaking encryption. *Cryptol ePrint Arch*. 2022. Available from: https://eprint.iacr.org/2022/085. [Accessed 2024].
10. Wang S, Cao Z. Practical identity-based encryption (IBE) in multiple PKG environments and its applications. *Cryptol ePrint Arch*. 2007. Available from: https://eprint.iacr.org/2007/100. [Accessed 2024].
11. Cha JC, Cheon JH. An identity-based signature from gap diffie-hellman groups. In: Public key cryptography—PKC 2003; 2003. vol. 2567, p. 18–30.
12. Wüst K, Gervais A. Do you need a blockchain? In: Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), 2008; Zug, Switzerland; p. 45–54.
13. Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. 2008. Available from: https://bitcoin.org/bitcoin.pdf. [Accessed 2024].
14. Wood G. Ethereum: a secure decentralized generalized transaction ledger. 2024. Available from: https://ethereum.github.io/yellowpaper/paper.pdf. [Accessed 2024].
15. Androulaki E, Barger A, Bortnikov V, Cachin C, Christidis K, De Caro A, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the 13th EuroSys Conference (EuroSys'18), 2018; New York, NY, USA; p. 1–15.
16. Brown RG. The corda platform: an introduction. 2018. Available from: https://www.corda.net/content/corda–platform–whitepaper.pdf. [Accessed 2024].

17. Bhaskaran K, Ilfrich P, Liffman D, Vecchiola C, Jayachandran P, Kumar A, et al. Double-blind consent-driven data sharing on blockchain. In: Proceedings of the 2018 IEEE International Conference on Cloud Engineering (IC2E), 2018; Orlando, USA; p. 385–91.
18. Gunasinghe H, Kundu A, Bertino E, Krawczyk H, Chari S, Singh K, et al. Prividex: privacy preserving and secure exchange of digital identity assets. In: Proceedings of the World Wide Web Conference, 2019; New York, NY, USA; p. 594–604.
19. Kokoris-Kogias E, Alp EC, Gasser L, Jovanovic P, Syta E, Ford BA. CALYPSO: private data management for decentralized ledgers. Proc VLDB Endow. 2020;14(4):586–99.
20. Zhu L, Wu Y, Gai K, Choo K-KR. Controllable and trustworthy blockchain-based cloud data management. Future Gener Comput Syst. 2019;91:527–35. doi:10.1016/j.future.2018.09.019.
21. Francati D, Ateniese G, Faye A, Milazzo AM, Perillo AM, Schiatti L, et al. Audita: a blockchain-based auditing framework for off-chain storage. In: Proceedings of the Ninth International Workshop on Security in Blockchain and Cloud Computing (SBC'21), 2021; New York, USA; p. 5–10.
22. Wang S, Yang M, Jiang S, Chen F, Zhang Y, Fu X. BBS: a secure and autonomous blockchain-based big-data sharing system. J Syst Architect. 2024;150:103133. doi:10.1016/j.sysarc.2024.103133.
23. Li Y, Li L, Zhao Y, Guizani N, Yu Y, Du X. Toward decentralized fair data trading based on blockchain. IEEE Netw. 2020;35(1):304–10. doi:10.1109/MNET.011.2000349.
24. Dixit A, Singh A, Rahulamathavan Y, Rajarajan M. Fast data: a fair, secure and trusted decentralized iiot data marketplace enabled by blockchain. IEEE Internet Things J. 2023;10(4):2934–44. doi:10.1109/JIOT.2021.3120640.
25. Chen F, Wang J, Jiang C, Xiang T, Yang Y. Blockchain based non-repudiable IoT data trading: simpler, faster, and cheaper. London, UK; 2022. p. 1958–67.
26. Alsharif A, Nabil M. A blockchain-based medical data marketplace with trustless fair exchange and access control. In: Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, 2020; Taipei, Taiwan; p. 1–6.
27. Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In: Proceedings of the 2007 IEEE Symposium on Security and Privacy—SP 2007, 2007; Berkeley, USA; p. 321–34.
28. Li Y-N, Feng X, Xie J, Feng H, Guan Z, Wu Q. A decentralized and secure blockchain platform for open fair data trading. Concurr Comput. 2019;32(7):e55785. doi:10.1002/cpe.5578.
29. Ma S, Mu Y, Susilo W. A generic scheme of plaintext-checkable database encryption. Inf Sci. 2018;429: 88–101. doi:10.1016/j.ins.2017.11.010.
30. Sing P, Masud M, Hossain MS, Kaur A. Cross-doamin secure data sharing using blockchain for industrial iot. J Parallel Distr Comput. 2021;156(7):176–84. doi:10.1016/j.jpdc.2021.05.007.
31. Yu X, Xie Y, Xu Q, Xu Z, Xiong R. Secure data sharing for cross-domain industrial iot based on consortium blockchain. In: Proceeding of the 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2023; Brazil: Rio de Janeiro; p. 1508–13.
32. Zeng S, Cao B, Sun Y, Sun C, Wan Z, Peng M. Blockchain-assisted cross-domain data sharing in industrial IoT. IEEE Internet Things J, 2023. doi:10.1109/JIOT.2023.3329577.
33. Dziembowski S, Eckey L, Faust S. FairSwap: how to fairly exchange digital goods. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security—CCS 2018, 2018; Toronto, Canada; p. 967–84.
34. Eckey L, Faust S, Schlosser B. OptiSwap: Fast optimistic fair exchange. In: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security—ASIA CCS 2020, 2020; Taipei, Taiwan; p. 543–57.

35. Qin B, Wang Q, Wu Q, Li S, Shi W, Bi Y, et al. BDTS: Blockchain-based data trading system. In: Information and communications security; 2023. vol. 14252, p. 645–64. doi:10.1007/978-981-99-7356-9_38.
36. Lesaege C, Ast F, George W. Kleros—long paper v2.0.2. 2021. Available from: https://kleros.io. [Accessed 2024].
37. Cha JC, Cheon JH, Kim Y. Batch verifications with id-based signatures. In: Information security and cryptology—ICISC 2004; 2005. vol. 3506, p. 233–48.
38. De Caro A, Iovino V. jPBC: java pairing based cryptography. In: Proceedings of the 16th IEEE Symposium on Computers and Communications (ISCC), 2011; Kerkyra (Corfu), Greece; p. 850–5.