SFOM-DT: A Secure and Fair One-to-Many Data Trading Scheme Based on Blockchain

Shuming Xiong, Pengchao Chen, Shusheng Ge and Qiang Ni

***Abstract*—The requirements for large amounts of data have promoted the rapid emergence of an industry for trading data. However, the current one-to-one trading constraints in the existing data trading schemes lead to low security and low efficiency. To tackle the challenges, a novel one-to-many distributed data trading scheme is proposed based on blockchain, which enables a data seller to sell one piece of data to multiple data buyers simultaneously, saving storage resources and computing resources significantly. Firstly, some new smart contracts are devised for two decentralized applications. Then, attribute-based searchable encryption technology is proposed to establish a data circulation scheme that realizes end-to-end encryption of data and ensures data security and highly efficient access. Finally, an inspection mechanism based on zero-knowledge proof and a pricing strategy based on the Stackelberg game are designed to guarantee fairness in trading and maximize revenue. The experiment results show that, in comparison to one-to-one trading, the high efficiency of this data trading scheme gradually emerges as the number of buyers (n) is greater than 2, and the run time is less than 1/10 of the former when n = 35. Furthermore, the pricing strategy can enable buyers and sellers to obtain more revenue when n > 4.**

***Index Terms*—Data trading, blockchain, zero-knowledge proof, Stackelberg game, attribute-based encryption.**

I. INTRODUCTION

WITH the rapid development of the Internet of Things (IoT) and Artificial Intelligence (AI) technologies, the amount of data generated and collected in the intelligent life of human society has exploded, involving many scenarios such as smart homes, smart grids, smart healthcare, and social networks [1]–[5]. In these scenarios, data are widely held by a few people. Nevertheless, the subjects who can tap into the potential value of the data generally lack access to the required data. In addition to utilizing data as a new production element, data trading can break down data silos [6]–[8]. By aggregating commercially valuable data and constructing a trusted trading platform, the value concealed within data can be fully released.

This work was supported in part by the National Nature Science Foundation of China (No. 62002139); in part by the funding of Innovation and Entrepreneurship Training Program for College Students in Jiangsu Province (No. 202210299179Y). *(Corresponding author: Qiang Ni).*

Shuming Xiong and Pengchao Chen are with the School of Computer Science and Communication Engineering and Jiangsu Key Laboratory of Security Technology for Industrial Cyberspace, Jiangsu University, Zhenjiang, 212013, China (e-mail: xsm@ujs.edu.cn; cpc@stmail.ujs.edu.cn).

Shusheng Ge is with the School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, 212013, China (e-mail: gss@stmail.ujs.edu.cn).

Qiang Ni is with the School of Computing and Communications, Lancaster University, Lancaster, LA1 4WA, UK (emails: q.ni@lancaster.ac.uk).

As a digital asset, data differ from traditional commodities because it has the characteristics of a low reproduction cost and difficult pricing. On the centralized data trading platform discussed in the literature [9]–[12], users lose control of the data once they upload it, and all transactions are simply recorded. The centralized market may cause users' data to be exploited maliciously, substantially impairing the rights and interests of users. At the same time, it runs the risk of having a single point of failure (SPoF), which will lead to the shutdown of data trading and privacy leakage.

Blockchain has the properties of decentralization and non-tampering, which also provides transparency and auditability. The literature [13]–[17] constructs decentralized data trading markets based on blockchain, which can maintain partial trading fairness and establish trust. However, these studies focus on constructing the trading platform without considering the trading model in detail, and data security protection needs to be improved while storing the data on the centralized server. The literature [18]–[24] mainly considers security protection through maintaining fairness in trading and storing data on cloud servers, which still have a certain probability of a SPoF. The previous works essentially satisfy the data security and trading fairness requirements for data trading, but they are limited to one-to-one trading. When the market grows to a specific scale, the data of one seller needs to be sold to multiple buyers, and one-to-one trading requires (1) preforming data encryption and transmitting once for each buyer; (2) storage operators to store multiple ciphertext copies of the same original data; and (3) blockchain to record multiple trading messages. The first two points will increase system resource consumption regarding bandwidth utilization, power supply, and storage media. In contrast, the third puts more workload on the blockchain and extends the time taken for transactions completion. Each extra transaction requires more resources due to some consensus methods of the blockchain, such as PoW (Proof of Work) [25] and execute-order-validate in Hyperledger Fabric [26]. Compared to one-to-one data trading schemes, our one-to-many scheme can reduce the number of operations on these aspects, thereby reducing the total trading time and resource consumption. So, our one-to-many scheme is more efficient. The experimental results in Section VII will also specifically validate it.

Most previous published works are limited to one-to-one trading, resulting in low efficiency and security flaws. Only Tian et al. [27] considered one-to-many data trading, but the research concern is different from this paper. They designed a mechanism for utility optimization in the multi-buyer scenario but did not design technically a detailed scheme. To address these problems, we propose a novel secure, fair one-to-many data trading scheme (SFOM-DT). It integrates multiple

transaction records of the same data purchased by multiple buyers into a single transaction and uploads them to the blockchain. It can reduce the demand for computing and storage overhead. The data circulation protocol and the inspection mechanism are constructed with robust security and great fairness. In addition, a pricing mechanism is designed to provide users with more revenue and encourage them to participate in one-to-many data trading. The contributions of this work can be summarized as follows:

1) To resolve the inefficiency of one-to-one data trading, a new blockchain-based one-to-many version is proposed, optimizing trading efficiency. To our knowledge, SFOM-DT is the first to technically focus on one-to-many data trading. It guarantees (i) the security of data, (ii) efficient trading, (iii) smaller resource consumption, and (iv) greater trading volume.

2) The encryption methods often used to protect data in one-to-one data trading works are unsuitable for one-to-many trading. To ensure data security in one-to-many data trading, end-to-end encryption is realized using improved attribute-based searchable encryption. We design partial decryption mechanism to improve security. Meanwhile, payment status is introduced as a critical attribute through which buyers can access the data only after paying to increase trading fairness.

3) The reputation mechanisms in existing work are vulnerable and only used post-sale. We construct a new trusted inspection mechanism based on zero-knowledge proofs to support marketplace inspection while keeping privacy. It will prevent defrauding and guarantee fairness. We design a non-negative proof based on the properties of square roots.

4) There is a deficiency in current research in calculating a suitable price for one-to-many data trading. By establishing a three-layer Stackelberg game architecture, we design a new one-to-many pricing mechanism by adding a new parameter representing the number of buyers. It can provide uniform and fair pricing and maximize the trading entities' revenue.

The remainder of this article is organized as follows: Section II outlines the related work. Section III constructs a framework for one-to-many data trading from the problem definition. Section IV describes the data circulation process. Section V presents the inspection mechanism. Section VI provides the data pricing scheme. Section VII shows the experiment setup and the comparison results. Section VIII analyzes the security of SFOM-DT. Section IX summarizes the work of this paper and provides an outlook for future work.

II. Related Work

IoT, AI, and other fast-growing technologies have triggered generation of massive volumes of data. Data trading as a new business paradigm has gradually drawn attentions of some works. With the development of distributed technologies, data markets are transforming into decentralized platforms, and the focus is gradually shifting to fairness and security.

A. Blockchain-based fair data trading solutions

Blockchain can provide auditability for data trading. Fair trading requires that neither buyer nor seller does not suffer additional losses. Ramchandaran *et al.* [28] constructed an IoT data trading platform to identify different elements that a fair decentralized market should have, such as payment schemes and ratings. However, they lack specifying the implementation scheme. Oh *et al.* [29] proposed a data trading model in which data agents consider providers' willingness while providers consider consumers' willingness. Their work specifies the details of fair trading but ignores data security protection. Zhang *et al.* [17] used smart contracts to design an efficient usage-controlled scheme for data trading that gives the data owner full control over the user's identity and operations. But there is a power imbalance between the owner and the user. Gupta *et al.* [13] introduced an intermediary between buyers and sellers. It makes trading easier, but the presence of the intermediary can significantly reduce the security of the system. Delgado *et al.* [14] proposed a data market with fair agreements, where the trading process can be terminated at any time based on signals to ensure that providers and consumers do not suffer. However, the scheme cannot detect false signals and only considers one trading process at a time. Nguyen *et al.* [30] presented an IoT data trading platform with three protocols customized for different demand scenarios of buyers and sellers to provide fair services. However, their work focused on performance evaluation while ignoring other trading elements. The above efforts utilize blockchain technology to build a data trading platform, which can avoid some problems of centralized data trading and improve fairness in trading. Nevertheless, some crucial components of the data market need to be sufficiently considered, such as payment settlement, storage methods, and data security.

B. Blockchain-based secure data trading solutions

The blockchain's chain structure and consensus mechanism can protect the chain's data security. Secure data trading should ensure that data is only obtained by paid buyers and is not leaked. Dai *et al.* [31] proposed a data trading ecosystem based on Ethereum, where the buyers get the data analysis results rather than the actual dataset. It can effectively solve the data security problem, but buyers generally need to get the dataset itself. An *et al.* [18] proposed a crowd-sensed data trading system with an STDR mechanism to provide reliable ratings. However, the robustness of their rating system is hampered by the difficulty of the used Ethereum platform in resisting the Sybil attack. Li *et al.* [22] developed a privacy-preserving data sharing scheme that designs access licenses to achieve data access control. However, their primary focus is securing data-sharing while neglecting trading details. Liu *et al.* [24] designed a transparent and reliable data marketplace architecture using cloud servers and blockchain as data storage units and controllers, respectively. However, data is stored in cloud servers with the risk of SPoF and data leakage. The works mentioned above focus mainly on data privacy protection, and they are limited to one-to-one data trading mode, which will decrease trading efficiency and consume more resources in the multi-buyer scenario. To address the problems in the above works, this paper constructs a blockchain-based one-to-many data trading scheme that can guarantee data security and trading fairness while meeting the demand for one-to-many trading mode and achieving efficient data access. Based on five data trading elements, our scheme is compared with some related works in TABLE I.

TABLE I
COMPARISON OF RELATED WORK ON TRADING SCHEME ELEMENTS

Elements	[12]	[14]	[18]	[29]	[31]	[33]	ours
Decentralization	N/A	√	√	√	√	√	√
Data security	√	N/A	√	N/A	√	√	√
Trading Fairness	N/A	√	N/A	√	√	√	√
Data Pricing	N/A	N/A	√	N/A	N/A	N/A	√
Trading Mode	1v1	1v1	1v1	1v1	1v1	1v1	1vn

* "√" and "N/A" denote the SUPPORT and NOT APPLICABLE of each element; "1v1" and "1vn" denote one-to-one and one-to-many.

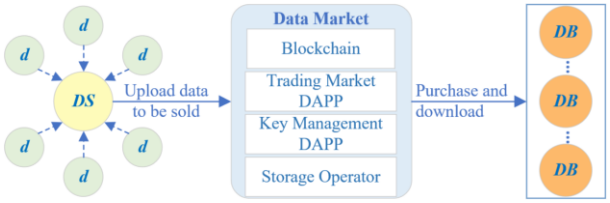


Fig. 1. Participating roles in one-to-many data trading.

III. ONE-TO-MANY DATA TRADING FRAMEWORK BASED ON THE BLOCKCHAIN

In this section, we describe the trading roles, analyze the problems and threats addressed in one-to-many data trading, and construct a secure and fair framework.

A. Data Trading Roles

Fig. 1 depicts the roles in data trading.

1) Data seller (*DS*): *DSs* are mainly composed of manufacturers or owners of various IoT devices (referred to as *d*), which continuously gather instant data, such as usage data from home devices, vital signs of patients from medical devices, and road information from the Internet of Vehicles, which are shown in the left part of Fig. 2. *DSs* want to leverage data market services to provide data commodity properties and match data buyers.

2) Data buyer (*DB*): *DBs* primarily consist of organizations interested in data that can mine the value of data and obtain more benefits by purchasing data, such as big data enterprises, factory R&D departments, and research institutes. *DBs* browse the information of commodities in the market, obtain data after payment, and then utilize the data according to their demands and mastered technologies to derive more benefits.

3) Data Market (*DM*): *DM* comprises blockchain, storage operators, decentralized applications (DAPPs), and several edge computing devices. DAPPs interact with the blockchain network using smart contracts. The Trading Market (TM) DAPP and the Key Management (KM) DAPP designed in this work handle the data trading process and maintain trading security, respectively. The blockchain is deployed on the edge computing devices to manage the ledger and record all data trading histories. As a for-profit organization, *DM* must profit from trading while also carrying out its responsibility as the supervising operator to ensure security and fairness.

B. Problem Definition and Threat Model

Compared to one-to-one trading, our one-to-many trading scheme can effectively reduce runtime and resource consumption in a multi-buyer scenario. However, the trading scheme elements in Table I are the primary requirements for data trading. The way to meet these conditions and respond to potential threats is a challenge to achieve one-to-many data trading. Therefore, four problems were defined.

1) Centralized platform: On the centralized data trading platform, threats mainly come from unsafe databases or SPoF. Users lose control of their data once they upload it. There is a risk that the platform will exploit and sell it illegally without authorization. Furthermore, there is no traceability or accountability for data utilization. So, data sellers do not have enough trust in a centralized platform to deliver data.

2) Insecure Data: Data is fragile, and there have been too many data theft incidents in real life. If data is stolen, it will lose its commercial value, which ruins trading. Some security threats exist in IoT systems, such as Distributed Denial of Service (DDoS) and Man-in-the-Middle (MITM) Attacks. Although the blockchain can help remove some threats, it still faces security threats, such as Sybil and Forking Attacks. In addition to network attacks, malicious users attempting to obtain data for free in other ways also threat data security.

3) Unfair Trading: The data platform should ensure that buyers can get the corresponding data after making payments and sellers can get the payments in time. The market should also punish dishonest sellers who provide poor-quality data. During the data circulation process, the data should only be accessible to buyers who have paid and not be stolen by any third party. Data trading must meet some criterion; otherwise, it will be considered unfair. The threat to fair trade comes from the exchange conflict between data and payment, as some buyers and sellers may be greedy. It should be pointed out that if data loses security, trading fairness will also be lost.

4) Pricing difficulties: The value of data is closely related to the trading price. Since the marginal cost is almost zero, the total revenue of data products cannot be maximized using the marginal cost and marginal revenue.

C. Trading Framework

The data trading framework is shown in Fig. 2. The *DS* continuously gathers data, which is aggregated and processed to participate in the *DM* as commodities. The distributed data market relies on TM DAPP to fulfill the trading mechanism, and it is secured by KM DAPP to ensure that there is never any direct communication between the DS and the DBs.

1) Trading Market DAPP: As a decentralized application, it simulates a data market to complete the trading between buyers and sellers, primarily carrying out the following four functions: ①Trading Process: *DS* sends a request for sale and uploads the commodity information. After receiving purchase requests and payments from *DB*s, the *DM* sends the data to each buyer and completes a settlement with *DS*. The overall trading process is shown in Fig. 3. Each process will call the corresponding smart contract. All trading details will be recorded on the blockchain; ②Pricing Mechanism: To solve the problem of difficult pricing, a pricing mechanism based on the Stackelberg game can provide a unified and optimal strategy; ③ Inspection Mechanism: Before the *DS* uploads data, the *DM* will inspect the data value indicators provided by the *DS*, and only data that is inspected successfully could be traded; ④ Credibility Mechanism: Each *DS* is assigned a

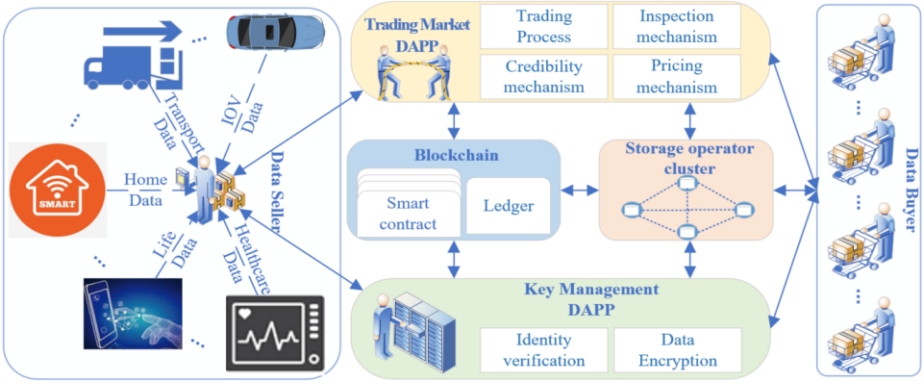


Fig. 2. One-to-Many data trading framework.

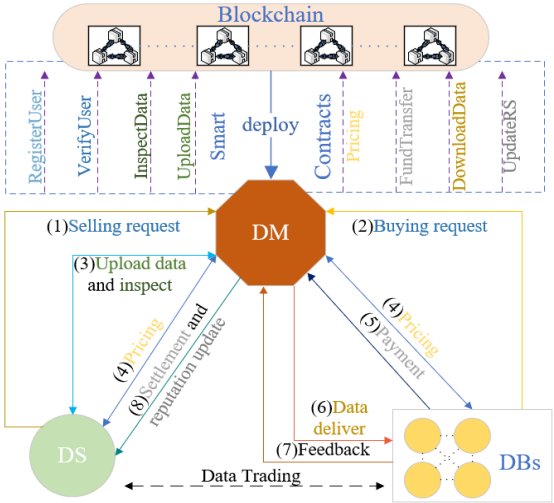


Fig. 3. Data trading process and corresponding smart contracts deployed on the blockchain.

corresponding reputation score, which will be updated based on the inspected results and the comprehensive feedback from *DB*s, and the data sold by the *DS* with good reputation score can get a priority push from the *DM* and more trust from *DB*s.

2) Blockchain: Although the public blockchains offer excellent security and reliability, they have performance bottlenecks [32], making them unsuitable for data trading with strict requirements for high throughput and low latency. The Hyperledger Fabric (henceforth called Fabric), which is applied in this framework as a modular consortium blockchain platform requiring user identification, contains pluggable consensus mechanisms and the membership service provider.

3) Key Management DAPP: As a security component in the data trading framework, it monitors the communication between various entities, protects data privacy in the process of circulation, and meets security requirements for data trading. It primarily accomplishes the following two functions: ① Identity Verification: It generates a unique identifier for each user according to their attributes and saves the attribute sets to verify users' identity; ② Data Encryption: It is responsible for generation and hosting of keys for attribute-based searchable encryption, confirming the validity of the trapdoors, and ensuring end-to-end data encryption.

4) Storage operator cluster: As a distributed cluster consisting of several peer nodes, it is responsible for storing data encrypted by sellers and delivering it to buyers.

IV. Data Circulation Scheme Under The One-to-Many Trading Framework

Data security is a fundamental requirement of data trading. Protecting data privacy helps maintain the economic benefit and safeguards the rights of data owners. Our data circulation scheme is constructed to fulfill the needs of one-to-many secure trading through the effective integration of blockchain and improved attribute-based searchable encryption.

A. Overview

The secure transmission of keys between buyers and sellers is crucial for safeguarding privacy in data trading. In FAST [33], the seller encrypts the data using simple symmetric encryption and stores the decryption key in the blockchain. However, it runs the risk of key leakage because other users can access the blockchain record. In BCDT [18], the seller encrypts the data using the buyer's public key according to asymmetric encryption. Then the buyer decrypts the data using his private key. Although BCDT can secure data and decryption keys, it is only suitable for one-to-one trading.

Searchable encryption is often used for data sharing and fits multi-user scenarios. Smart contracts run automatically as scripts without interference from user operations, which can ensure the accuracy of the search results. Attribute-based encryption provides fine-grained access control, which can accurately grant access rights to users in multi-user scenarios. Further, we have improved the attribute-based searchable encryption algorithm to adapt to the one-to-many data trading. Among them, the encryption of the data is performed by a symmetric encryption algorithm, which can quickly complete encryption and decryption operations. The larger the amount of data, the more advantageous it is. However, the security of data is completely guaranteed by the security of the key. Therefore, the attribute-based searchable encryption algorithm can obtain security by encrypting the symmetric key, and we design an end-to-end dual encryption scheme based on the bilinear operation mechanism that can ensure data security.

The scheme involves the theoretical basics of bilinear mappings. Let G_1, G_2, and G_T be cyclic groups of prime order p. Let g_1 and g_2 be the generators of G_1 and G_2, respectively. Let e be a bilinear mapping, $e: G_1 \times G_2 \rightarrow G_T$, and the detailed features of e can be referenced from [34].

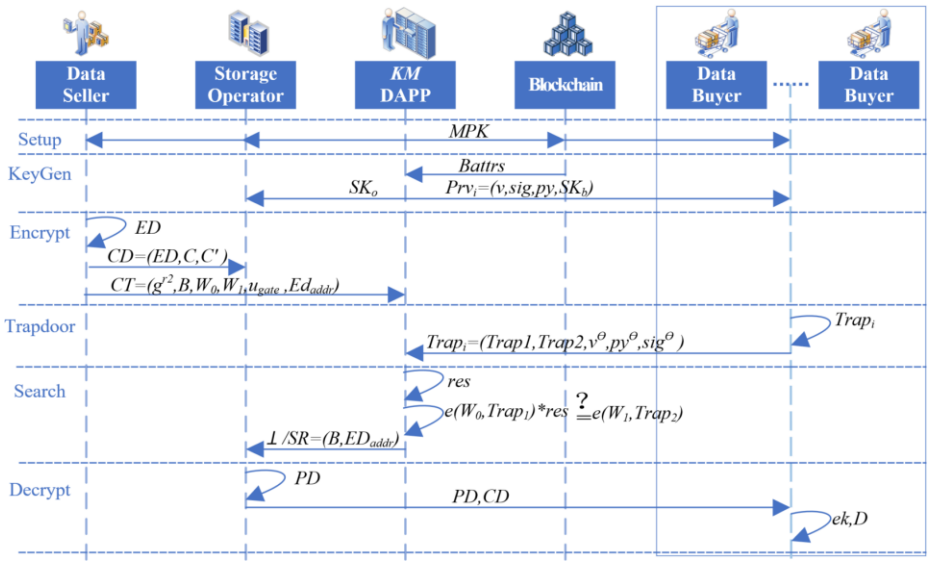


Fig. 4. Data circulation process based on attribute-based searchable encryption.

B. Detailed Scheme

Fig. 4 depicts the data circulation process using attribute-based searchable encryption. The data is transmitted securely through the *KM* DAPP, the storage operator, and the blockchain platform for end-to-end encryption. All processes will be recorded on the blockchain as part of the trading. Table II includes a list of the main notations used in this section.

1) $Setup(\lambda) \longrightarrow (MPK, MSK)$: The generation phase of system parameters, which occurs during the system initialization period, is executed by the *KM* DAPP to generate MPK and MSK. As the system's public parameters, the former is sent to other entities to provide uniform parameters for the following five phases. The latter, as the master key of the system, is primarily used in the $KeyGen$ phase.

a. According to the input security parameter λ, generate g, G, G_T, and a bilinear mapping $e: G \times G \to G_T$. G and G_T are two cyclic groups of prime order p, and g is the generator of G.

b. Define a collision-resistant hash function $H: \{0,1\}^* \to Z_p$. Randomly select three numbers $\alpha, \beta, \gamma \in Z_p$.

c. Define the attribute set $U = \{attr_1, attr_2, \cdots, attr_n\}$. Randomly select sets $\{t_1, \cdots, t_{2n}\}, t_i \in Z_p$ and $\{x_1, x_2, \cdots, x_{2n}\}$ $, xi \in G$. Set $T_i = g^{-t_i}$ and $y_i = e(x_i, g)$, where $1 \le i \le 2n$.

d. Generate the system's public parameters and master key.

$$MPK = \{u, g, g^{\alpha}, g^{\beta}, g^{\gamma}, H, (T_i, y_i)_{i \in \{1,2,\cdots,2n\}}\},$$
$$MSK = \{\alpha, \beta, \gamma, (t_i, x_i)_{i \in \{1,2,\cdots,2n\}}\}.$$

2) $Encrypt(MPK, D, AP, w) \longrightarrow (CT, CD)$: The encryption phase of the data, where the seller generates a symmetric key ek, encrypts the data D, and then generates CD and CT according to the access policy AP and data keyword index, which are sent to the storage operator and *KM* DAPP.

a. Input MPK and the access policy $pol = \{Dattr_1, Dattr_2, \cdots, Dattr_n\}$. Using a symmetric key ek *DS* encrypts the data D and gets the encrypted data $ED = SK\_Enc(D, ek)$. *DS* creates an $index$ based on the data keyword w.

b. Randomly select three numbers $s, r_1, r_2 \in Z_p$. Calculate ciphertext $C = e(g,g)^{\alpha \cdot s} \cdot ek$ and $C' = g^s$, and generate $CD = (ED, C, C')$. Send CD to the storage operator and get the returned storage address ED_{addr}.

TABLE II
NOTATIONS IN DATA CIRCULATION SCHEME

Symbol	Description
λ	A security parameter
U	The global attribute set
MPK	The system's public parameters
MSK	The system's master key
$Battrs$	The attribute set of data buyer
Prv	The attribute key of data buyer
SK_o	The partial decryption key held by the storage operator
D	The data to trade
AP	The access policy set by data seller
w	The keyword of data
ek	A symmetric key for encrypting data
ED	The encrypted data
CD	The ciphertext containing ED and the elements for obtaining ek
CT	The ciphertext used to verify trapdoors
bw	The keyword searched by data buyer
$Trap$	The trapdoor generated by data buyer
SR	The search result
PD	The partially decrypted data using SK_o

c. For each attribute $Dattr_i$ in the $policy$, if $Dattr_i \in U$, set $T_i' = T_i$, otherwise, $T_i' = T_{i+n}$. Set $u_{gate} = g^{r_2} \cdot \prod_{i=1}^{n} T_i'$.

d. Set $W_0 = g^{\gamma \cdot r_1}$, $W_1 = g^{\alpha \cdot (r_1 + r_2)} \cdot g^{\beta \cdot r_1 \cdot H(w)}$ and $B = e(g^{\gamma \cdot s}, g^{\beta})$. Generate $CT = (g^{r_2}, B, W_0, W_1, u_{gate}, ED_{addr})$ and upload it to *KM* DAPP.

3) $KeyGen(MPK, MSK, Battrs) \longrightarrow (Prv, SK_o)$: The generation phase of the buyer's attribute key occurs after the buyer completes the payment, and the *KM* DAPP generates an attribute key prv for the buyer and a partial decryption key SK_o for the storage operator.

a. Input MPK, MSK, and the buyer's attribute set $Battrs$. *KM* DAPP randomly selects a number $\alpha_1 \in Z_p$ and generates α_2, which equals $(\alpha - \alpha_1)$. Set $SK_b = g^{\alpha_1} \cdot g^{\gamma \cdot \beta}$ and $SK_o = g^{\alpha_2}$. Send SK_o to the storage operator.

b. Set $v = g^{\alpha \cdot \gamma}$. For each attribute $Battr_i$ in $Battrs$, if $Battr_i \in U$, set $py_i = y_i$ and $sig_i = x_i \cdot v^{t_i}$, otherwise, $py_i = y_{i+n}$, $sig_i = x_{i+n} \cdot v^{t_{i+n}}$.

c. Set $sig = \prod_{i=1}^{n} sig_i$ and $py = \prod_{i=1}^{n} py_i$, generate $Prv = (v, sig, py, SK_b)$ and send it to DB.

4) $Trapdoor(MPK, prv, bw) \longrightarrow (Trap)$: The generation phase of the buyer's trapdoor occurs when the buyer obtains the attribute key and prepares to obtain the data. The buyer

uses the trapdoor function to wrap the attribute key to generate the trapdoor so that the $Search$ phase can verify whether its attributes meet the requirements through the trapdoor alone.

a. DB randomly selects a number $\theta \in Z_p$. To generate the search trapdoor for keyword bw, calculate $Trap1 = \left(g^{\alpha} \cdot g^{\beta \cdot H(bw)}\right)^{\theta}$ and $Trap2 = g^{\theta \cdot \gamma}$.

b. Generate the trapdoor $Trap = (Trap1, Trap2, v^{\theta}, sig^{\theta}, py^{\theta})$and send it to *KM* DAPP.

5) $Search(MPK, CT, Trap) \longrightarrow (SR/\perp)$: After the buyer sends $Trap$, the *KM* DAPP checks whether his attributes meet *AP*. If they do, the *B* and ED_{addr} in the CT are sent to the storage operator; otherwise, an error message is returned.

a. Calculate $res = e\left(u_{gate}, v^{\theta}\right) \cdot e\left(sig^{\theta}, g\right)/py^{\theta}$. If $Battrs$ satisfies *AP*, the following two equations hold: $res = e(g,g)^{\alpha \cdot \gamma \cdot \theta \cdot r_2}$ and $e(W_0, Trap1) * res = e(W_1, Trap2)$.

b. If the two equations do not hold, return the error identifier "$\perp$"; otherwise, generate the search result $SR = (B, ED_{addr})$ and send it to the storage operator.

6) $Decrypt(MPK, SK_b, SK_o, SR) \longrightarrow (D)$: The storage operator, after receiving SR, accesses the encrypted data and partially decrypts it using the SK_o to generate PD, and then sends the PD and CD to the buyer. The buyer calculates the decryption key from them and further obtains the data.

a. The storage operator retrieves the encrypted data according to ED_{addr} and then calculates $PD = B/e(C', SK_o) = e(g,g)^{\gamma \cdot \beta \cdot \theta}/e(g,g)^{\alpha_2 \cdot \theta}$. Send PD and CD to DB.

b. DB uses (SK_b, PD) to calculate $ek = C \cdot PD/e(C', SK_b)$, and decrypts the data according to $D = SK\_Dec(ED, ek)$.

Fine-grained access control is achievable by attribute-based encryption, and the buyer can only get the encrypted data and decryption key through trapdoor authentication when its attributes satisfy the seller's preset conditions. The payment status, which is controlled by smart contracts, is the most crucial attribute of a buyer. Other attributes of the buyer are mainly set during registration, such as enterprise type and business scope. Meanwhile, they will not be forcibly changed to meet the policy. For example, a smart home manufacturing company only allows non-competitive companies to purchase its data, and it can set its policy to "non furniture manufacturing type and paid". Other types of users can meet the policy and obtain data after payment, such as the institute, electric company, and housekeeping APP.

It is clear from the six phases above that data is always stored and circulated in ciphertext during the trading process. It is worth mentioning that, compared to other schemes, our scheme newly introduces the function of partial decryption by setting two partial keys, requiring cooperation between the market and buyers to complete the decryption, thus improving security. Meanwhile, the buyer must obtain the key after multiple verifications using the bilinear operation to hide the decryption key in ciphertext C. Therefore, our scheme can preserve data security and uphold the fairness of trading.

C. Proof of correctness

1) *The Correctness of Trapdoor Search:* As shown in the *Search* phase, after the buyer generates a trapdoor, the *KM* DAPP performs a search match based on the trapdoor, and the result is judged according to whether both of the equations $res = e(g,g)^{\alpha \cdot \gamma \cdot \theta \cdot r_2}$ and $e(W_0, Trap1) \cdot res = e(W_1, Trap2)$ hold. The following derivation process proves the correctness of the equations.

$$
\begin{aligned}
&\text{res} \\
&= \frac{e\left(u_{gate}, v^{\theta}\right) \cdot e\left(sig^{\theta}, g\right)}{py^{\theta}} \\
&= \left[\frac{e\left(u_{gate}, g^{\alpha \cdot \gamma}\right) \cdot e(sig, g)}{py}\right]^{\theta} \\
&= \left[\frac{e(g^{r_2} \cdot \prod_{i=1}^{n} g^{-t_i}, g^{\alpha \cdot \gamma}) \cdot e(\prod_{i=1}^{n} x_i \cdot g^{\alpha \cdot \gamma \cdot t_i}, g)}{\prod_{i=1}^{n} e\,(x_i, g)}\right]^{\theta} \\
&= [e(g^{r_2}, g^{\alpha \cdot \gamma}) \cdot e(\prod_{i=1}^{n} g^{-t_i}, g^{\alpha \cdot \gamma}) \cdot e(\prod_{i=1}^{n} g^{\alpha \cdot \gamma \cdot t_i}, g^{\alpha \cdot \gamma})]^{\theta} \\
&= e(g,g)^{\alpha \cdot \gamma \cdot \theta \cdot r_2}
\end{aligned}
$$

$$
\begin{aligned}
&e(W_0,\ Trap1) \cdot e(g,g)^{\alpha \cdot \gamma \cdot \theta \cdot r_2} \\
&= e\left(g^{\gamma \cdot r_1}, \left(g^{\alpha} \cdot g^{\beta \cdot H(w)}\right)^{\theta}\right) \cdot e(g,g)^{\alpha \cdot \gamma \cdot \theta \cdot r_2} \\
&= e(g,g)^{\alpha \cdot \gamma \cdot \theta \cdot r_1} \cdot e(g,g)^{\beta \cdot \gamma \cdot \theta \cdot r_1 \cdot H(w)} \cdot e(g,g)^{\alpha \cdot \gamma \cdot \theta \cdot r_2} \\
&= e(g,g)^{\alpha \cdot \gamma \cdot \theta \cdot (r_1 + r_2)} \cdot e(g,g)^{\beta \cdot \gamma \cdot \theta \cdot r_1 \cdot H(w)}
\end{aligned}
$$

$$
\begin{aligned}
&e(W_1,\ Trap2) \\
&= e\left(g^{\alpha \cdot (r_1 + r_2)} \cdot g^{\beta \cdot r_1 \cdot H(bw)}, g^{\theta \cdot \gamma}\right) \\
&= e(g,g)^{\alpha \cdot \gamma \cdot \theta \cdot (r_1 + r_2)} \cdot e(g,g)^{\beta \cdot \gamma \cdot \theta \cdot r_1 \cdot H(bw)}
\end{aligned}
$$

When the hash values of the keywords bw and w are equal, it is easily obtained that $e(W_0, Trap1) \cdot res = e(W_1, Trap2)$.

2) *The Correctness of Obtaining the Key*: As shown in the *Decrypt* phase, after receiving the ciphertext and PD, the buyer needs to obtain the key ek for decrypting, which is obtained by computing $C \cdot PD/e(C', SK_b)$. The following derivation process proves the correctness of obtaining the key.

$$
\begin{aligned}
&\frac{C \cdot PD}{e(C', SK_b)} \\
&= \frac{e(g,g)^{\alpha \bullet \theta} \cdot ek \cdot e(g,g)^{\gamma \bullet \beta \bullet \theta}}{e(g^{\theta}, g^{\alpha_1} \cdot g^{\gamma \bullet \beta}) \cdot e(g,g)^{\alpha_2 \bullet \theta}} \\
&= \frac{e(g,g)^{\alpha \bullet \theta} \cdot ek \cdot e(g,g)^{\gamma \bullet \beta \bullet \theta}}{e(g,g)^{\alpha_1 \bullet \theta + \gamma \bullet \beta \bullet \theta + \alpha_2 \bullet \theta}} \\
&= \frac{e(g,g)^{\alpha \bullet \theta} \cdot ek \cdot e(g,g)^{\gamma \bullet \beta \bullet \theta}}{e(g^{\theta}, g^{\alpha_1 + \alpha_2}) \cdot e(g,g)^{\gamma \bullet \beta \bullet \theta}} \\
&= ek
\end{aligned}
$$

V. Zero-knowledge Proof-based Inspection Mechanism

A. Zero-knowledge Proofs and Cryptographic Commitments

Zero-knowledge proof is a probability-based verification in which the prover (*P*) can convince the verifier (*V*) that an assertion is correct without disclosing any information that would reveal a secret [35]. There are three crucial properties of zero-knowledge proof, namely completeness, reliability, and zero-knowledge.

Pedersen cryptographic commitment, which allows users to hide secrets with perfectly hiding and computationally binding properties, is applied to establish an inspection mechanism to generate proofs. To commit a secret value $x \in Z_p$, the user first selects a random number $r \in Z_p$ to hide the commitment. Then, the user computes the commitment by $Cm(x,r) = g^x \cdot h^r$, where g and h are the two random generators of the multiplicative group G.

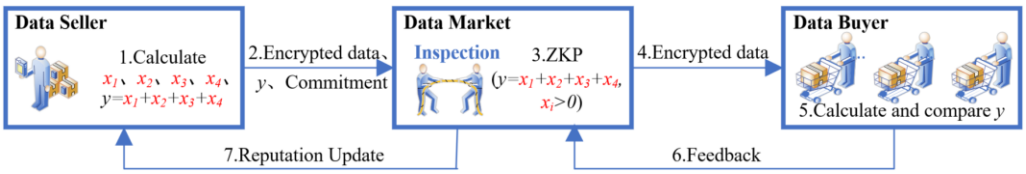


Fig. 5. Data market inspection mechanism based on the zero-knowledge proof.

B. Inspection Mechanism

In *DM*, *DS* may be a dishonest user seeking more profit by selling data with a lower actual value than the claimed value. Although reputation incentives can assist *DBs* in selecting *DS* with higher scores, the reputation metrics described in existing works depend mainly on buyers' post-sale evaluations. On the one hand, there may be dishonest buyers submitting low scores or sellers hiring buyers to submit high scores, and on the other hand, due to the unique commodity form of data, *DB* cannot return the data once they receive it.

Therefore, a fair and trustworthy third party is needed to inspect the data before delivery. Due to concerns over privacy protection, sellers are hesitant to allow *DM* to perform the inspection by obtaining the data. So, the inspection mechanism is constructed based on zero-knowledge proofs to address this problem, as shown in Fig. 5, where *DM* can accurately inspect the goods without having access to the data and then update the reputation score of sellers based on the inspection results.

In the zero-knowledge proof-based inspection mechanism, *DS* acts as the prover, *DM* acts as the verifier, and the knowledge to be verified is the value of the trading data. Data value depends on the relative value index of data assets, which is primarily reflected in four aspects: multidimensionality, activity, information entropy [36], and acquisition cost. The four values are weighted, processed, and recorded as x_1, x_2, x_3, and x_4. Meanwhile, let the total value of data be $y = x_1 + x_2 + x_3 + x_4$. The sellers need to publish the total data value to get strong competitiveness in *DM*. However, the specific values of the data in four different aspects are business secrets that cannot be made public. Dishonest sellers may publish false total data values to gain more profits. Therefore, *DM* can use zero-knowledge proofs to verify whether the total data value published by sellers indeed originates from the specific values of the four aspects. Besides, it regards any data with a negative x_i as inferior data that needs to be tested for non-negativity.

The inspection mechanism is shown in Algorithm 1. According to the principle used for non-negativity verification, if x is not negative, $x = \sqrt{x^2}$. Three crucial properties of the zero-knowledge proof are elaborated separately to prove the viability of the inspection mechanism. ① *Completeness*: if the data seller is honest, x_i is not negative, and $y = \sum_{i=1}^{4} x_i$. The inspection paradigm is automatically triggered according to the smart contract, which assures fair and trustworthy inspection results. It makes the market as the verifier can trust the seller's proof; ② *Reliability*: if the seller is dishonest and incorrectly publishes the value, it cannot pass the checking paradigm, and the seller cannot deceive the market; ③ *Zero-knowledge*: according to the hidden nature of Pedersen commitment, the market will only get the commitment generated by the secret x_i and random numbers.

Algorithm 1 Inspection Mechanism

Input: $x_i (i = 1,2,3,4)$, y
Output: inspection result

1: *DS* squares x_i, takes the square root, and records $\sqrt{x_i{}^2}$;
2: *DS* randomly generates r_i, r^*, and $r_i' \in Z_p$, and sends the commitments $Cm\left(\sqrt{x_i{}^2}, r_i'\right)$, $Cm(0, r^*)$, and $Cm(x_i, r_i)$ to *DM*;
3: *DM* sends a random challenge $\beta \in Z_p$ to *DS*;
4: *DS* replies with $Z_r = r^* + \beta \cdot \sum_{i=1}^{4} r_i$ and $Z_{r_i} = \beta \cdot r_i - \beta \cdot r_i'$;
5: *DM* checks whether the equations $g^{\beta \cdot y} \cdot h^{z_r} = Cm(0, r^*) \cdot \left(\prod_{i=1}^{4} Cm(x_i, r_i)\right)^{\beta}$ and $Cm\left(\sqrt{x_i{}^2}, r_i'\right)^{\beta} \cdot h^{Z_{r_i}} = Cm(x_i, r_i)^{\beta}$ hold;
6: If the equations hold, set the result as a success and increase the seller's reputation score. Otherwise, set the result as a failure and deduct the reputation score;
7: return result.

C. Proof of correctness

The Correctness of Inspection: As shown in step 5 of Algorithm 1, the market needs to check whether two equations hold. The correctness derivation and proof of the first equation are shown below, and the other equation proof is similar.

$$\begin{aligned} & g^{\beta \bullet y} \cdot h^{z_r} \\ &= g^{\beta \cdot \sum_{i=1}^{4} x_i} \cdot h^{r^* + \beta \cdot \sum_{i=1}^{4} r_i} \\ &= g^0 \cdot g^{\beta \cdot \sum_{i=1}^{4} x_i} \cdot h^{r^*} \cdot h^{\beta \cdot \sum_{i=1}^{4} r_i} \\ &= \left(g^0 \cdot h^{r^*}\right) \cdot \left(g^{\sum_{i=1}^{4} x_i} \cdot h^{\sum_{i=1}^{4} r_i}\right)^{\beta} \\ &= Cm(0,\ r^*) \cdot \left(\prod_{i=1}^{4} Cm(x_i, r_i)\right)^{\beta} \end{aligned}$$

VI. Pricing Mechanism Based on Stackelberg Game

In one-to-many data trading, multiple buyers should pay equal amounts to obtain the same data. However, the data value is uncertain. *DM* needs to earn the difference between the selling price and the purchase price, which affects the revenue of *DM* and its attractiveness to users. This section constructs a two-stage, three-layer Stackelberg game pricing architecture to establish a reasonable and fair price for the data. We add a parameter representing the number of buyers, and redesign the expected return function for all parties.

A. Revenue Formulation

The Stackelberg game is an information-dynamic game in which the main idea is that the leader and the follower continuously adjust their decisions according to each other's strategies until the game reaches the Nash equilibrium. As shown in Fig. 6, an optimal strategy can be obtained for data pricing through an initial strategy and two-stage subgames.

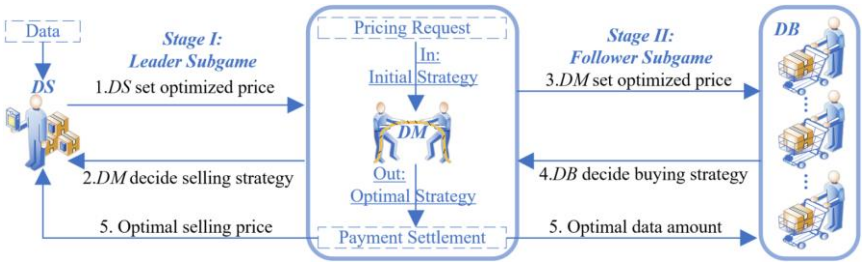


Fig. 6. Data market pricing architecture based on Stackelberg game.

TABLE III
NOTATIONS IN PRICING MECHANISM

Symbol	Description
n	The number of data buyers
x, x^*	The quantity and optimal quantity of data in trading
$p_s, p_s{}^*$	The unit selling price and optimal unit selling price set by *DS*
$p_b, p_b{}^*$	The unit buying price and optimal unit buying price set by *DM*
C_s	The cost of unit data set by data seller
C_m	The cost of unit data set by data market in one-to-many trading
C_{m1}	The cost of unit data set by data market in one-to-one trading
C_b	The cost of unit data set by data buyer
b_i	The revenue parameter of data buyer i
V_i	The revenue obtained by data buyer i
SU, MU, BU	The revenue of *DS*, *DM*, and *DB*

The fundamental condition of the game is to derive the expected revenue function of all users. The primary notations used in this section are listed in Table III.

The data seller gathers the data using its IoT devices, then processes it and uploads it to the storage operator servers for a unit cost denoted by C_s. p_s is the initial selling unit price set by the seller, n is the number of buyers, and the seller will consider selling only when $n \cdot p_s > C_s$. Using x to denote the number of units of trading data quantity, the data seller's expected revenue can be expressed as

$$SU(x, p_s) = n \cdot p_s \cdot x - C_s \cdot x \quad (1)$$

At the intermediate layer, the data market must host the data uploaded by sellers at the storage operator. The storage and trading costs per unit of data are denoted as C_m. p_b is the initial purchase unit price paid by buyers to the data market, so the expected revenue of the data market can be obtained by

$$MU(x, p_b, p_s) = n \cdot p_b \cdot x - n \cdot p_s \cdot x - C_m \cdot x \quad (2)$$

Data buyers profit from mining and exploiting data; the profit margin is usually related to x and is positively logarithmic [37]. The revenue that various buyers derive from the same data is different, and the revenue obtained by buyer $i(i = 1,2,\cdots n)$ are defined as $V_i = b_i \cdot ln(1 + x)$, where b_i denotes the revenue parameter of buyer i. The cost of acquiring a unit of raw data is denoted as C_b so that the expected total revenue of n buyers can be written as

$$BU(x, p_b) = \sum_{i=1}^{n} V_i(x) - p_b \cdot x \cdot n - C_b \cdot x \cdot n \quad (3)$$

B. Stackelberg Equilibrium Points

Using backward induction to analyze the three-layer Stackelberg game, an optimal strategy can be obtained to reach Stackelberg equilibrium by solving the subgame between different layers. All parties will apply this optimal pricing strategy to maximize their revenue. First, the game between data buyers and the data market is analyzed to form the subgame of buyers, which can be represented as followed.

Problem 1 (Buyers' Subgame):

$$\underset{x,p_b}{maximize}\ BU(x, p_b)$$
$$subject\ to\ x > 0, p_b > 0$$

Given p_b, buyers determine their optimal buying strategy x^* to maximize revenue. Derive the first-order and second-order derivatives of the buyers' revenue in formula (3) with respect to x, which can be written as follows:

$$\frac{\partial BU(x,p_b)}{\partial x} = \frac{\sum_{i=1}^{n} b_i}{1+x} - (p_b + C_b) \cdot n \quad (4)$$

$$\frac{\partial^2 BU(x,p_b)}{\partial x^2} = -\sum_{i=1}^{n} b_i \cdot (1 + x)^{-2} < 0 \quad (5)$$

Since the second-order derivative is constantly negative, BU is a strictly convex function. BU obtains its maximum value when $\partial BU(x, p_b)/\partial x = 0$, as follows:

$$x^* = \frac{\sum_{i=1}^{n} b_i}{n \cdot (p_b + C_b)} - 1 \quad (6)$$

Based on the optimal buying strategy of *DB*s, *DM* can adapt p_b to maximize revenue. The subgame of *DM* can be written as followed.

Problem 2 (Market's Subgame):

$$\underset{x,p_b}{maximize}\ MU(x, p_b, p_s)$$
$$subject\ to\ x > 0, p_b > p_s + C_m/n > 0$$

The market's subgame aims to set the optimal buying price under the condition that *DB*s use the optimal strategy. Therefore, substituting formula (6) into formula (2) obtains the revenue of *DM* to be rewritten as

$$MU(x^*, p_b, p_s) = (n \cdot p_b - n \cdot p_s - C_m) \cdot \left(\frac{\sum_{i=1}^{n} b_i}{n \cdot (p_b + C_b)} - 1\right) \quad (7)$$

Derive the first-order and second-order derivatives of the market's revenue in (7) with respect to p_b, which can be written as follows:

$$\frac{\partial MU(x^*,p_b,p_s)}{\partial p_b} = \frac{\sum_{i=1}^{n} b_i}{n \cdot (p_b + C_b)^2} \cdot (n \cdot C_b + n \cdot p_s + C_m) - n \quad (8)$$

$$\frac{\partial^2 MU(x^*,p_b,p_s)}{\partial {p_b}^2} = -2\sum_{i=1}^{n} b_i \cdot \frac{n \cdot C_b + n \cdot p_s + C_m}{n \cdot (p_b + C_b)^3} < 0 \quad (9)$$

Similar to $BU(x, p_b)$, $MU(x^*, p_b, p_s)$ is also a strictly convex function. Based on $\partial MU/\partial p_b = 0$, MU obtains the optimal pricing strategy, which is as follows:

$${p_b}^* = \frac{\sqrt{\sum_{i=1}^{n} b_i \cdot (n \cdot C_b + n \cdot p_s + C_m)}}{n} - C_b \quad (10)$$

Due to the second-order derivatives of $BU(x, p_b)$ and $MU(x^*, p_b, p_s)$ being both negative, x^* and ${p_b}^*$ are both globally and uniquely optimal, and they are the Stackelberg game equilibrium points.

In order to maximize revenue, *DS* can dynamically adapt p_s according to the optimized strategy of *DM*. The subgame of *DS* can be described as followed.

Problem 3 (Seller's Subgame):

$$\underset{x,p_b}{maximize}\ SU(x, p_s)$$

$$subject\ to\ x > 0, n \cdot p_s > C_s > 0$$

According to these subgames, the optimal strategy points for reaching the Stackelberg equilibrium can be defined as

$$p_s{}^* = arg \underset{x,p_s}{maximize}\ SU(x^*, p_s)$$
$$subject\ to\ (x^*, p_b{}^*) = arg \underset{x,p_b,p_s}{maximize}\ MU(x^*, p_b, p_s)$$
$$x^* = arg \underset{x,p_b}{maximize}\ BU(x, p_b)$$

Substituting formulas (6) and (10) into formula (1) gains the revenue of *DS* to be expanded as

$$SU(x^*, p_s) = (p_s - C_s) \cdot \left(\sqrt{\frac{\sum_{i=1}^{n} b_i}{n \cdot C_b + n \cdot p_s + C_m}} - 1 \right) \quad (11)$$

Derive the first-order and second-order derivatives of SU with respect to p_s , which can be expressed as follows:

$$\frac{\partial SU(x^*, p_s)}{\partial p_s} = n \cdot \sqrt{\sum_{i=1}^{n} b_i} \cdot \frac{n \cdot C_b + \frac{1}{2} p_s + C_m + \frac{1}{2} C_s}{(n \cdot C_b + n \cdot p_s + C_m)^{\frac{3}{2}}} - 1 \quad (12)$$

$$\frac{\partial^2 SU(x^*, p_s)}{\partial {p_s}^2} = n^2 \cdot \sqrt{\sum_{i=1}^{n} b_i} \cdot \frac{-n \cdot C_b - \frac{1}{4} n \cdot p_s - C_m - \frac{3}{4} C_s}{(n \cdot C_b + n \cdot p_s + C_m)^3} < 0 \quad (13)$$

Therefore, $SU(x^*, p_s)$ is also a strictly convex function. $p_s{}^*$, which is globally unique and optimal, is the solution to equation $\partial SU(x^*, p_s)/\partial p_s = 0$.

In conclusion, by applying backward induction in the constructed Stackelberg game, the optimal strategy x^* is first calculated based on the low-level game, and the optimal strategy $p_b{}^*$ for the market is generated based on x^*. When the low-level game achieves equilibrium, the optimal strategy $p_s{}^*$ can be obtained based on the subgame of the data seller, at which time the global Stackelberg equilibrium is reached. Meanwhile, $(x^*, p_b{}^*, p_s{}^*)$ is the global equilibrium point.

C. Comparison of The Revenue of Different Schemes

In the one-to-one data trading, n buyers purchase data from one seller, and the number of generated transactions is n. Therefore, the seller needs to encrypt and upload the data n times, and the storage operator needs to store n copies of the data. In this case, the cost required by *DM* to complete storing and trading unit data is denoted as C_{m1}, at which point the expected revenue of *DM* can be expressed as

$$MU(x, p_b, p_s)_{1v1} = n \cdot p_b \cdot x - n \cdot p_s \cdot x - C_{m1} \cdot x \cdot n \quad (14)$$

When only one buyer is involved in the trading, attribute-based searchable encryption has more computational overhead. However, the computational overhead is lower than the encryption algorithm for the one-to-one trading scheme when multiple buyers are involved. As a result, when n is large enough, it satisfies $C_{m1} < C_m < n \cdot C_{m1}$ and $MU(x, p_b, p_s)_{1v1} < MU(x, p_b, p_s)$. In the multi-buyer scenarios, the one-to-many trading improves the market's revenue; however, buyers' and seller's trading prices and costs still do not change, so their revenue is not affected. To increase the revenue of buyers and sellers while improving the revenue of *DM* and maximizing revenue for each party, this scheme takes the derived optimal strategy. The comparison of the revenue of buyers and sellers in each scheme is shown as follows:

$$BU(x, p_b)_{1v1} = BU(x, p_b) < BU(x^*, p_b{}^*) \quad (15)$$
$$SU(x, p_s)_{1v1} = SU(x, p_s) < SU(x^*, p_s{}^*) \quad (16)$$

According to formulas (15) and (16), the Stackelberg game can increase revenue for buyers and sellers by decreasing some additional revenue obtained by the data market in our one-to-many data trading scheme. Under the premise of security and fairness, buyers and sellers prefer to participate in the data market where they can obtain higher revenue, so the pricing mechanism can attract more users to take part in data trading. The increase in the number of users will also promote the increase in trading profits, achieving a dynamic virtuous cycle. In the pricing mechanism, relevant parameters include the number of buyers, storage costs, encryption costs, etc. For the same data, except for the number of buyers, other parameters will not change significantly in a similar period, therefore, the prices of data in the market tend to be stable.

VII. System Evaluation and Performance Analysis

A. Experiment Setup

The experiment was set up on a virtual machine, and the Docker container was installed for configuring Fabric and VerneMQ networks. A test tool recorded the runtime of each test trade by simulating different numbers of buyers buying data from one seller. The Fabric network was configured with three organizations, and the number of blockchain nodes was controlled by adjusting the number of peer nodes in each organization. The number of storage operator servers was controlled by adjusting the number of nodes in VerneMQ, and there was one storage operator server by default. In addition, the parameters involved in the experiment include the number of buyers and the quantity of purchased data, which are assigned by manual input. After the test tool has been run to simulate trading, a file recording the results of the test trading runtime is generated. The source code is released publicly[1].

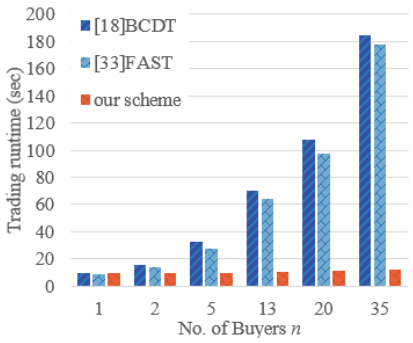
B. Evaluation of Trading Performance

To demonstrate the efficiency of SFOM-DT, BCDT [18] and FAST [33] are simulated and implemented as one-to-one data trading schemes for comparison with our one-to-many scheme. BCDT uses a symmetric encryption algorithm, while FAST uses asymmetric encryption. The two schemes are aiming to one-to-one data trading and use runtime and resource overhead as experimental indicators. To ensure the results are fairly evaluated, we only compared our scheme with the encryption schemes used in BCDT and FAST to avoid interference from other factors.

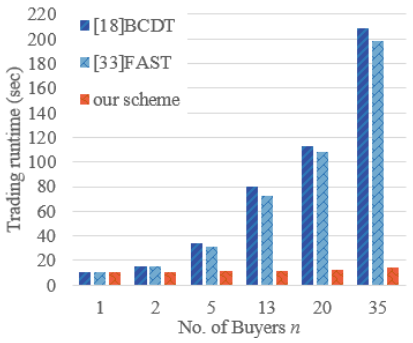
Under the condition of fixing one storage operator node and 1KB of data quantity, the comparison of trading runtime for the various numbers of buyers in the Fabric network with 3 and 6 nodes, respectively, is shown in Fig. 7. When the number of buyers (n) is 1, the difference between the runtime of our scheme and BCDT is small, and they are slightly higher than the runtime of FAST. However, starting from n=2, the trading runtime of BCDT and FAST increases rapidly with the number of buyers, but our trading runtime grows slowly and is significantly lower than others. Regardless of the number of buyers in our scheme, the seller only needs to encrypt and upload the data once, and the market only needs to generate a single purchase transaction, thus improving trading efficiency.

As can be seen from Fig. 7(a) and (b), the trading runtime increases slightly for all three schemes as the number of blockchain nodes increases. Since writing transactions that

[1] https://github.com/cpc99/one-to-many-data-trading.git



(a) 3 blockchain nodes



(b) 6 blockchain nodes

Fig. 7. Trading runtime with varying number of buyers.

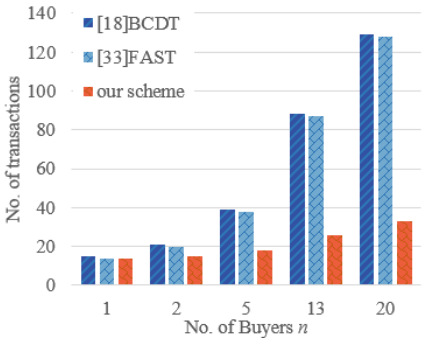


Fig. 8. The number of transactions generated after a trade.

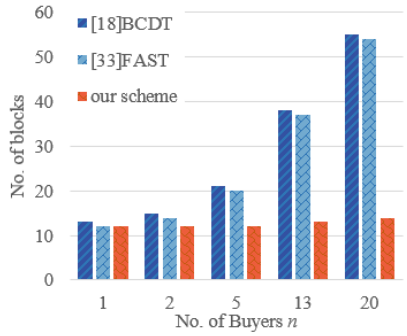
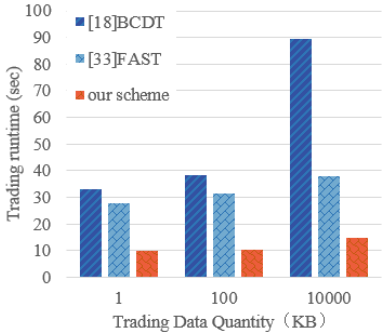
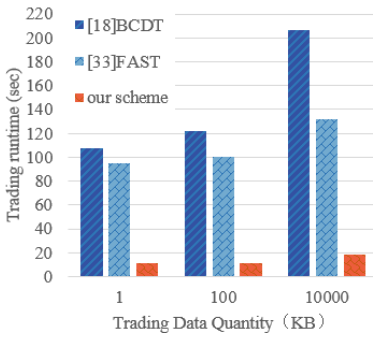


Fig. 9. The number of blocks generated in the blockchain after a trade.



(a) 5 buyers



(b) 20 buyers

Fig. 10. Trading runtime of different trading data quantity.

TABLE IV
THE EXECUTION TIME OF ENCRYPTION ALGORITHMS IN DIFFERENT SCHEMES (SEC.)

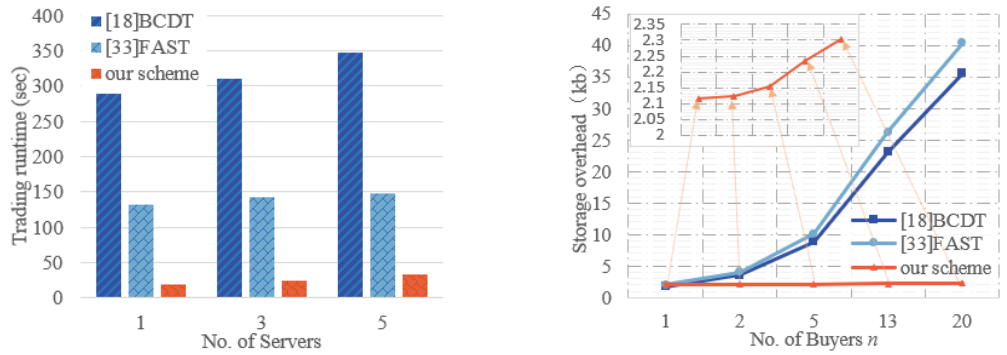
Algorithms	1-1kb	1-10mb	20-1kb	20-10mb	35-1kb	35-10mb
AES	0.00559	0.68178	0.01394	12.62188	0.01917	19.93487
RSA	0.02482	4.04396	0.75649	70.40594	2.41812	120.03579
ABSE	0.30927	0.73676	1.12649	8.11653	2.27100	13.43612

change the state of the blockchain affects the synchronization time, the more nodes, the longer the transaction synchronization will take. Fig. 8 and Fig. 9 present the variation in the number of transactions and blocks generated in the blockchain after completing a trade under different buyer numbers. These numbers can be obtained from the deployed blockchain browser Hyperledger Explorer. In BCDT and FAST, as the number of buyers grows, the seller needs to generate more trading and upload them to the blockchain. As a result, both the number of transactions and the blocks responsible for recording transaction information increase on the blockchain. As transaction synchronization time in our scheme has significantly decreased, the impact of a change in the number of nodes on transaction execution time has also been reduced.

Table IV provides the execution time of the encryption algorithms in FAST, BCDT, and our scheme, which correspond to AES, RSA, and attribute-based searchable encryption (ABSE), respectively. "1-1kb" indicates that one buyer purchases 1 kb of data. As the number of buyers increases, the number of runs of AES, RSA, and the phases involving buyers in ABSE, such as $KeyGen$, also increases, leading to a growth in the execution time. Among them, ABSE needs to execute all six phases, so its runtime is the longest under the condition of a smaller number of buyers or data quantity. However, the millisecond time difference does not significantly impact the trading runtime. Due to our scheme always needing to execute the $Setup$ and $Encrypt$ phases only once, its runtime efficiency is higher than others when the number of buyers or data quantity exceeds a specific size.

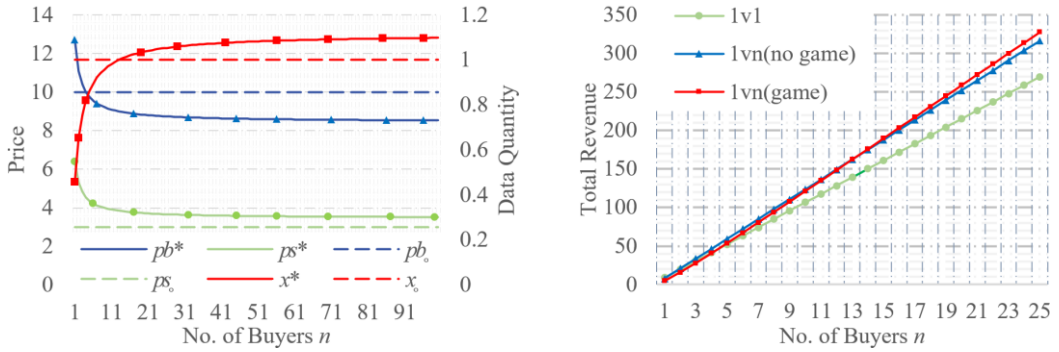
To demonstrate the effect of data quantity on trading runtime, the runtime required for 5 buyers and 20 buyers to purchase 1 KB, 100 KB, and 10 MB of data under different schemes is given in Figs. 10(a) and 10(b). Combined with TABLE IV, it can be seen that the encryption algorithm execution time and trading runtime of BCDT vary most significantly as the quantity changes. Although the RSA algorithm used in BCDT is more secure, it is only applicable to a small amount of critical data, and encrypting and decrypting a large amount of data will consume more time. The runtime of our scheme does not change significantly with the increase in quantity. Therefore, SFOM-DT is more suitable for large-scale data trading.

As shown in Fig. 11(a), the trading runtime grows as the number of storage operator servers increases. Despite having the most significant rise in trading runtime, the BCDT's ratio of addition to runtime increases less. In our scheme, the storage operator servers are required to participate in the data flow process for several times. Therefore, the ratio of trading runtime increases the most. However, the absolute value of runtime is much lower than in other schemes. Fig. 11(b) depicts the overhead of storage resources required for various numbers of buyers to trade 1 KB of data. Due to the necessity to store various keys, ciphertexts, attributes, and other



(a) Impact of the number of servers on runtime (b) The storage overhead of different schemes

Fig. 11. The impact of the number of storage operator servers on trading runtime and the storage overhead of different schemes.



(a) Optimal pricing strategy (b) Total social revenue

Fig.12. Optimal pricing strategy and total social revenue.

information, the storage resources needed for our scheme are slightly higher than those of other schemes when n=1. In the multi-buyer scenario, the one-to-one data trading scheme requires storage operators to store multiple copies of data, and as the number of buyers increases, the storage overhead becomes significantly higher than our scheme.

C. Evaluation of Pricing Revenue

To evaluate the pricing revenue, the simulation experiment sets the initial trading data quantity $x_0 = 1.0$, purchase price $p_{b_0} = 10.0$, sale price $p_{s_0} = 3.0$, and the remaining initial parameters are $\sum_{i=1}^{n} b_i/n = 20.0$, $C_b = 1.0$, $C_s = 2.0$, $C_{m1} = 2.0$, and $C_m = 3.0$. We calculated the optimal pricing strategy for a different number of buyers under the initial parameters, including the optimal trading data quantity x^*, and the optimal purchase/sale price $p_b{}^*/p_s{}^*$. Then, we calculated the tripartite revenue obtained using the optimal strategy.

The optimal strategy for a different number of buyers is shown in Fig. 12(a), where x^* is slightly lower than x_0 when n < 13 and slightly higher than x_0 when n > 13, and the difference is satisfied by the actual controllable range of trading data quantity. Meanwhile, $p_s{}^*$ is always higher than p_{s_0}. When n > 4, $p_b{}^*$ becomes lower than p_{b_0}. In order to gain more revenue, buyers want to reduce the purchase price, and sellers want to increase the sale price, so the market can attract more users to join by using the optimal strategy when n > 4.

Fig. 13 and Fig. 12(b) depict the detailed revenue and total social revenue of buyers, seller, and the market in the one-to-one data trading scheme (1v1), the one-to-many data trading scheme using initial pricing (1vn (no game)), and the one-to-many data trading scheme using an optimal pricing strategy (1vn (game)), respectively. As shown in Fig. 13, SU and BU in 1vn (no game) are equal to SU and BU in 1v1, and MU is always higher than MU in other schemes at n > 2, while MU in 1vn (game) is higher than MU in 1v1 at n > 15. By using the optimal strategy, 1vn (game) has higher SU and BU at n > 3 and n > 4, respectively, than those two schemes. As seen in Fig. 12(b), the total revenue of 1vn (game) is higher than that of 1vn (no game) at n > 12, and higher socioeconomic benefits can be obtained. When 4<=n<=12, both BU and SU are larger. As a result, 1vn (game) can attract more buyers and sellers to participate in trading, which is more beneficial to development of the data market.

VIII. Security Analysis

In this section, we will analyze the security of SFOM-DT according to the problems and security threats. The following theorems and lemmas are proven to demonstrate the security of the SFOM-DT.

Theorem 1. SFOM-DT can guarantee data security through ensuring that it is not stolen by any malicious attacker.

Proof 1. This theorem can be demonstrated through the following three lemmas. ■

Lemma 1.1. SFOM-DT can prevent Forking and Sybil attacks.

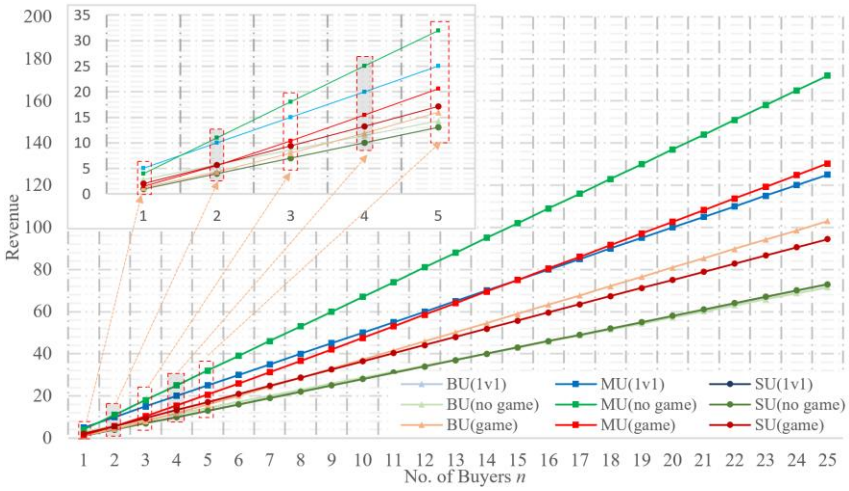


Fig. 13. Tripartite revenue of three different schemes.

Proof 1.1. Since the blockchain in our framework is a permissioned network, only trusted nodes with a unique identity will be added. Because the consensus process in Fabric is a three-step process including execute-order-validate, forking attacks are immune. Public blockchains are often vulnerable to Sybil attacks because the same node can fake multiple IDs. ID faking is difficult in the permissioned blockchain; no single node has absolute power. Therefore, SFOM-DT has an advantage in resisting Sybil attacks. ■

Lemma 1.2. SFOM-DT can prevent MITM and DDoS attacks.

Proof 1.2. Users can only enter the network with permission and will be assigned a unique identity. A user must sign the message with his private key to publish or query transactions on the network at any time. The recipient also verifies the signature when receiving the message. Therefore, SFOM-DT can prevent MITM and DDoS attacks. ■

Lemma 1.3. SFOM-DT can prevent an adversary (a user whose attributes do not meet the policy) from obtaining data.

Proof 1.3. We consider a game between a challenger and an adversary as follows:

Setup: The challenger inputs security parameters and generates the system's public parameters and master key.

Query: The adversary generates his attributes. The challenger executes $KeyGen()$ and sends the key Prv to the adversary.

Challenge: The challenger executes $Encrypt()$ and sends the ciphertext CT (including u_{gate}) to the adversary.

Guess: The adversary inputs Prv and gets a trapdoor $Trap$ (including $v^{\theta}, sig^{\theta}, py^{\theta}$). The probability of the adversary obtaining data is:

$$\epsilon = Pr[e(u_{gate}, v^{\theta}) \cdot e(sig^{\theta}, g)/py^{\theta} = e(g,g)^{\alpha \cdot \gamma \cdot \theta \cdot r_2}]$$

Because smart contracts control attributes, users cannot freely change their attributes. If the Computational Diffie-Hellman problem in the bilinear group is hard and an adversary's attributes do not meet the policy, ϵ is negligible. Therefore, SFOM-DT can prevent a user whose attributes do not meet the policy from obtaining data. ■

Theorem 2. SFOM-DT can guarantee that data trading is fair.

Proof 2. First, the buyer will receive a new attribute after successful payment for an order. If all other attributes meet the access policy, the buyer can successfully obtain the decryption key to get the data. According to Theorem 1, the data is secure, so the buyer can get complete and valid data. If the buyer does not make payment, they will not receive the attribute of successful payment and cannot obtain data. The buyer's payment will be made to the data market. If the buyer's other attributes do not meet the access policy, their payment will be refunded. Otherwise, the data market will pay the seller after the buyer obtains the data. The acquisition of data and the settlement are performed based on smart contracts, and the execution is autonomous, tamper resistant, and unbiased. No malicious parties can affect buyers' ability to obtain data after successful payment, and sellers will receive the payment they deserve. Therefore, SFOM-DT can guarantee buyer fairness and seller fairness.

Secondly, the inspection mechanism can effectively prevent data release with a false value, and the pricing mechanism can provide unified and fair pricing. Therefore, SFOM-DT can prevent the unfair exchange of unequal values.

Thirdly, when the seller uploads data, a unique integrity credential will be generated for the data based on the hash algorithm. The credential will be sent to the data buyer along with the data, and the buyer will generate the hash value of the obtained data based on the same hash algorithm. If the data is complete, the hash value will be equivalent to the content of the integrity certificate. On the other hand, this paper designs an end-to-end encryption protocol, where data always exist in the form of ciphertext during the trading process, which has extremely high security and can ensure the integrity of the data. Therefore, SFOM-DT can ensure the integrity of data.

Finally, if baleful buyers deliberately give negative feedback, SFOM-DT can guarantee the fairness of the reputation mechanism. In our one-to-many data trading, false feedback can be found according to the comprehensive opinions of other honest buyers. Besides, the inspection mechanism designed in our scheme can effectively judge the value of data, which is also the basis for evaluating the authenticity of buyers' feedback. In addition, in our scheme, only successful buyers can give feedback. It is unrealistic for malicious users to manipulate enough buyers to affect comprehensive feedback results at a high cost. ■

IX. Conclusion

In this paper, we proposed a consortium blockchain-based one-to-many data trading scheme that improves the attribute-based searchable encryption to accomplish one-to-many trading, guaranteeing strong security and efficient access. Meanwhile, an inspection mechanism was constructed to maintain trading fairness. Finally, we designed a new data pricing mechanism based on Stackelberg game to provide an optimal pricing strategy for one-to-many trading. According to security analysis, these components undertake different functions to wholly construct a more secure and fair data trading platform. The data market without any one above will face significant risks. We did extensive experiments, and the results showed that our trading scheme is efficient, and the pricing mechanism can maximize the revenue. In future research, some functions like payment modules will be implemented to deploy a real data trading market, and a strict post-sale monitoring mechanism will be established to prevent data resale. Meanwhile, we will further consider the scalability of the trading system and the impact of increasing data trading volumes on the performance of proposed scheme.

References

[1] C. Lin, D. He, S. Zeadally, X. Huang, and Z. Liu, "Blockchain-based data sharing system for sensing-as-a-service in smart cities," *ACM Trans. Internet Technol.*, vol. 21, no. 2, pp. 1-21, Jun. 2021.

[2] F. Loukil, C. Ghedira-Guegan, K. Boukadi, A-N. Benharkat, and E. Benkhelifa, "Data privacy based on IoT device behavior control using blockchain," *ACM Trans. Internet Technol*., vol. 21, no. 1, pp. 1-20, 2021.

[3] E. Y. Song, G. J. FitzPatrick, K. B. Lee and E. Griffor, "A methodology for modeling interoperability of smart sensors in smart grids," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 555-563, Jan. 2022.

[4] A.A. Abdellatif, L. Samara , A. Mohamed, et al., "MEdge-chain: leveraging edge computing and blockchain for efficient medical data exchange," *IEEE Internet of Things J.*, vol. 8, no. 21, pp. 15762-15775, 1 Nov. 2021.

[5] F. Cerruto, S. Cirillo, D. Desiato, et al. "Social network data analysis to highlight privacy threats in sharing data," *J Big Data*, vol. 9, no. 19, pp. 1-26, Feb. 2022.

[6] Y. Miao, Q. Huang, M. Xiao and W. Susilo, "Blockchain assisted multi-copy provable data possession with faults localization in multi-cloud storage," *IEEE Trans Inf. Forensics Secur.*, vol. 17, pp. 3663-3676, 2022.

[7] X. Liu, G. Chen, Y. Li, L. Chen, Q. Meng and C. Mehdi-Souzani, "Sampling via the aggregation value for data-driven manufacturing," *Natl. Sci. Rev.*, vol. 9, no. 11, pp. 1-11, Nov. 2022.

[8] D. Hu, Y. Li, L. Pan, M. Li, S. Zheng, "A blockchain-based trading system for big data, " *Comput. Networks*, vol. 191, pp. 1-13, Mar. 2021.

[9] Z. Cai, X. Zheng and J. Wang, "Efficient data trading for stable and privacy preserving histograms in Internet of Things," in *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, 2021, pp. 1-10.

[10] R. C. Fernandez, P. Subramaniam, and M. J. Franklin, "Data market platforms: trading data assets to solve data problems," *Proc. VLDB Endow.*, vol. 13, no. 12, pp. 1933–1947, Aug. 2020.

[11] G. Gao, M. Xiao, J. Wu, S. Zhang, L. Huang and G. Xiao, "DPDT: a differentially private crowd-sensed data trading mechanism," *IEEE Internet of Things J.*, vol. 7, no. 1, pp. 751-762, Jan. 2020.

[12] C. Niu, Z. Zheng, F. Wu, X. Gao and G. Chen, "Achieving data truthfulness and privacy preservation in data markets," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 1, pp. 105-119, 1 Jan. 2019.

[13] P. Gupta, V. Dedeoglu, K. Najeebullah, S. S. Kanhere and R. Jurdak, "Energy-aware demand selection and allocation for real-time IoT data trading," in *2020 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2020, pp. 138-147.

[14] D. S. Sergi, P. S. Cristina, N. A. Guillermo, H. J. Jordi, "A fair protocol for data trading based on Bitcoin transactions," *Future Gener. Comput. Syst.*, vol 107, pp. 832-840, 2020.

[15] Y. Li, L. Li, Y. Zhao, N. Guizani, Y. Yu and X. Du, "Toward decentralized fair data trading based on blockchain," *IEEE Network.*, vol. 35, no. 1, pp. 304-310, Feb. 2021.

[16] Y. Zhao., Y. Yu, Y. Li, G. Han, and X. Du, "Machine learning based privacy-preserving fair data trading in big data market," *Information Sciences*, vol. 478, pp. 449-460, April 2019.

[17] X. Zhang, X. Li, Y. Miao, et al., "A data trading scheme with efficient data usage control for industrial IoT," *IEEE Trans. Ind. Inf.*, vol. 18, no. 7, pp. 4456-4465, Jul. 2022.

[18] B. An, M. Xiao, A. Liu, Y. Xu, X. Zhang and Q. Li, "Secure Crowdsensed Data Trading Based on Blockchain," in *IEEE Transactions on Mobile Computing*, vol. 22, no. 3, pp. 1763-1778, 1 March 2023.

[19] C. Chen, J. Wu, H. Lin, W. Chen and Z. Zheng, "A secure and efficient blockchain-based data trading approach for Internet of Vehicles," *IEEE Trans. Veh. Technol*., vol. 68, no. 9, pp. 9110-9121, Sept. 2019.

[20] S. Khezr, A. Yassine, R. Benlamri and M. S. Hossain, "An edge intelligent blockchain-based reputation system for IIoT data ecosystem," *IEEE Trans. Ind Inform.*, vol. 18, no. 11, pp. 8346-8355, Nov. 2022.

[21] J. Zhang, B. Guo, X. Ding, D. Hu, and Y. Jiang, "Distributed supervision model for enterprise data asset trading based on blockchain multi-channel in industry alliance," *Sensors*, vol. 22, no. 20, pp. 7842-7864, Oct. 2022.

[22] T. Li, H. Wang, D. He and J. Yu, "Blockchain-based privacy-preserving and rewarding private data sharing for IoT," *IEEE Internet of Things J.*, vol. 9, no. 16, pp. 15138-15149, Aug. 2022.

[23] B. G. Jeong, T. Y. Youn, N. S. Jho, S. Shin. "Blockchain-based data sharing and trading model for the connected car," *Sensors*, vol. 20, no. 11, pp. 3141-3160, Jun. 2020.

[24] D. Liu, C. Huang, J. Ni, X. Lin and X. S. Shen, "Blockchain-cloud transparent data marketing: consortium management and fairness," *IEEE Trans. Comput.*, vol. 71, no. 12, pp. 3322-3335, 1 Dec. 2022.

[25] E. Tedeschi, T. A. S. Nordmo, D. Johansen, and H. D. Johansen, "On optimizing transaction fees in bitcoin using AI: investigation on miners inclusion pattern," *ACM Trans. Internet Technol.*, vol. 22, no. 3, pp. 1-28, Aug 2022.

[26] J. Jeong, D. Kim, S. Y. Ihm, Y. Lee, and Y. Son, "Multilateral personal portfolio authentication system based on Hyperledger Fabric," *ACM Trans. Internet Technol.*, vol. 21, no. 1, pp. 1-17, Feb 2021.

[27] L. Tian, J. Li, W. Li, B. Ramesh and Z. Cai, "Optimal Contract-Based Mechanisms for Online Data Trading Markets," IEEE *Internet of Things J.*, vol. 6, no. 5, pp. 7800-7810, Oct. 2019.

[28] G. S. Ramachandran, R. Radhakrishnan, and B. Krishnamachari, "Towards a decentralized data marketplace for smart cities," in *2018 IEEE International Smart Cities Conference (ISC2)*, 2018, pp. 1-8.

[29] H. Oh, S. Park, G. M. Lee, J. K. Choi and S. Noh, "Competitive data trading model with privacy valuation for multiple stakeholders in IoT data markets," *IEEE Internet of Things J.*, vol. 7, no. 4, pp. 3623-3639, Apr 2020.

[30] L. D. Nguyen, I. Leyva-Mayorga, A. N. Lewis and P. Popovski, "Modeling and analysis of data trading on blockchain-based market in IoT networks," *IEEE Internet of Things J.*, vol. 8, no. 8, pp. 6487-6497, 2021.

[31] W. Dai, C. Dai, K. K. R. Choo, C. Cui, D. Zou and H. Jin, "SDTE: a secure blockchain-based data trading ecosystem," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 725-737, 2020.

[32] J. Abdella, Z. Tari, A. Anwar, A. Mahmood and F. Han, "An architecture and performance evaluation of blockchain-based peer-to-peer energy trading," *IEEE Trans. Smart Grid*, vol. 12, no. 4, pp. 3364-3378, 2021.

[33] A. Dixit, A. Singh, Y. Rahulamathavan and M. Rajarajan, "FAST DATA: A Fair, Secure, and Trusted Decentralized IIoT Data Marketplace Enabled by Blockchain," in *IEEE Internet of Things J.*, vol. 10, no. 4, pp. 2934-2944, 15 Feb.15, 2023.

[34] Y. Ren, J. Qi, Y. Liu, J. Wang, and G. J. Kim, "Integrity verification mechanism of sensor data based on bilinear map accumulator," *ACM Trans. Internet Technol.*, vol. 21, no. 1, pp. 1-19, Feb 2021.

[35] S. Xu, X. Cai, Y. Zhao, Z. Ren, L. Du, Q. Wang, et al. "zkrpChain: towards multi-party privacy-preserving data auditing for consortium blockchains based on zero-knowledge range proofs," *Future Gener. Comput. Syst.*, vol 128, pp. 490-504, 2022.

[36] W. Wang, C. Chen, W. Yao, K. Sun, W. Qiu and Y. Liu, "Synchrophasor data compression under disturbance conditions via cross-entropy-based singular value decomposition," *IEEE Trans. Ind. Inf.*, vol. 17, no. 4, pp. 2716-2726, Apr 2021.

[37] K. Liu, X. Qiu, W. Chen, X. Chen and Z. Zheng, "Optimal pricing mechanism for data market in blockchain-enhanced Internet of Things," *IEEE Internet of Things J.*, vol. 6, no. 6, pp. 9748-9761, Dec. 2019.