Leveraging Consortium Blockchain for Secure Cross-Domain Data Sharing in Supply Chain Networks

Runqun Xiong, *Member, IEEE*, Jing Cheng, Xirui Dong, Jiahang Pu, and Feng Shan, *Member, IEEE*

***Abstract*—Supply Chain Networks (SCNs) play a vital role in achieving strategic decision-making for production and distribution facilities, aiming to meet market demands and gain competitive advantages. With the application of new-generation information technology in the supply chain, enterprises within SCNs generate a substantial volume of relevant business data. Sharing this data among SCN enterprises can effectively reduce operating costs, optimize business processes, and enhance the overall efficiency of the supply chain. However, effective data sharing among SCN participants faces challenges, such as data leakage, data quality assurance, and fair data value allocation. To address these challenges, this paper proposes a secure cross-domain data sharing model in SCNs (named SCN-CDSM) based on consortium blockchain technology. The model introduces trust, enables cross-domain data exchange, and promotes cooperation among supply chain enterprises. To ensure privacy, group signatures and access control smart contracts are designed, along with an approach to reduce blockchain throughput limitations. Furthermore, a sharing incentive mechanism utilizing the Stackelberg game model based on data value is designed to foster fairness and collaboration. Extensive numerical simulations are conducted to demonstrate the effectiveness of the proposed schemes, achieving both security and efficiency in data sharing within SCNs.**

***Index Terms*—Supply chain networks, data sharing, cross-domain, consortium blockchain, stackelberg game.**

I. Introduction

SUPPLY Chain Networks (SCNs) require strategic decision-making regarding the number, location, capacity, and functionality of production and distribution facilities within a cooperative group of enterprises. This is achieved through information sharing, resource integration, and collaboration, aiming to meet dynamic market demands and create competitive advantages for the participating enterprises. With the widespread adoption of new-generation information technologies, such as the Internet of Things (IoT), cloud computing, and artificial intelligence, SCNs generate a substantial volume of business data from upstream and downstream partners. Effective sharing of this data among SCN participants can lead to significant reductions in operational costs, including inventory redundancy, thereby benefiting all involved participants [1], [2]. Taking the fast-moving consumer goods (FMCG) industry as an example, a large retailer relies on real-time production and order information from multiple suppliers to adjust its warehousing plans promptly, adapting to seasonal changes or market trends. During this process, suppliers are required to provide up-to-date information regarding their production and distribution capabilities, enabling the retailer to make timely decisions. Real-time data sharing allows the retailer to optimize inventory levels and reduce the risks of stock-outs and surpluses. However, in practice, a well-established and effective data sharing model among SCN enterprises (referred to as "cross-domain" in this paper) is still lacking, the main challenges are as follows.

- *Trust Issues in Data Sharing:* In SCNs, enterprises are often hesitant to share sensitive data due to concerns about potential data leakage or misuse by competitors [3]. For example, a FMCG supplier may be reluctant to share inventory data with multiple retailers, fearing that competitors might exploit this information for market advantage. This lack of trust reduces transparency and responsiveness across the supply chain.
- *Disagreement on Data Value and Quality:* Data producers and consumers often have differing views on the value and utility of shared data [4]. For instance, a food supplier in FMCG-SCN may consider its sales and inventory data highly valuable for optimizing retailer inventory, but the retailer may prioritize consumer purchasing trends and see limited value in the supplier's data. This misalignment can impede data-sharing agreements and affect collaboration efficiency.
- *Lack of Proper Incentives for Data Producers:* Data producers often incur significant costs to provide real-time data, such as investments in IT systems or data maintenance, without receiving adequate compensation. This lack of financial reward discourages data producers from sharing valuable data [5]. For example, a supplier may invest in advanced IT infrastructure to update inventory data in real-time but may not receive proportional financial returns, reducing their motivation to share.

To address these challenges, we propose the Supply Chain Network Cross-Domain Data Sharing System Model (SCN-CDSM) with a Stackelberg game-based incentive mechanism.

Received 22 December 2023; revised 29 December 2024; accepted 16 February 2025. Date of publication 20 February 2025; date of current version 10 April 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB2900100, in part by the National Natural Science Foundation of China under Grant 62172091, Grant 62232004, and Grant 61602112, in part by the Jiangsu Provincial Key Laboratory of Network and Information Security under Grant BM2003201, and in part by the Key Laboratory of Computer Network and Information Integration of the Ministry of Education of China under Grant 93K-9. *(Corresponding author: Runqun Xiong.)*

Runqun Xiong, Xirui Dong, and Feng Shan are with the School of Computer Science and Engineering, Southeast University, Nanjing 211189, China (e-mail: rxiong@seu.edu.cn; sirid@seu.edu.cn; shanfeng@seu.edu.cn).

Jing Cheng and Jiahang Pu are with the College of Software Engineering, Southeast University, Nanjing 211189, China (e-mail: chengjing@seu.edu.cn; jiahangpu@seu.edu.cn).

Digital Object Identifier 10.1109/TSC.2025.3544130

This model enhances data security, builds a trust framework, and provides a fair compensation structure, ultimately improving the overall efficiency and collaboration within SCNs. The main contributions of this research are as follows.

- *Access Control Mechanism:* The proposed SCN-CDSM is based on consortium blockchain technology that integrates Access Control Lists (ACLs) and Proxy Re-Encryption (PRE) through smart contracts to improve the automatic execution and security management of access control in a cluster of stakeholder enterprises. This approach avoids the high computational and storage bottlenecks associated with Attribute-Based Encryption (ABE), while providing stronger security than ACLs alone. Additionally, we incorporate group signatures to ensure privacy, traceability, and accountability, thus strengthening user privacy protection.
- *Security and Privacy of Data Sharing:* By combining an off-chain storage solution, InterPlanetary File System (IPFS), with consortium blockchain, we optimize storage efficiency and integrate group signature and hash verification techniques to ensure data integrity and non-repudiable traceable verification. Our model strikes a better balance between security and efficiency, making SCN-CDSM more suitable for real-world SCN scenarios where both security and performance are critical.
- *Incentive Mechanisms for Data Sharing:* Our approach introduces a novel Stackelberg game-based incentive mechanism. Unlike existing studies, we consider the diverse cooperative relationships among supply chain participants (such as suppliers, manufacturers, distributors, and customers) within the consortium blockchain. These partners establish varying degrees of data sharing, influenced by factors such as transaction frequency, business confidentiality, and creditworthiness. SCN-CDSM dynamically adjusts data pricing through smart contracts based on these relationships, as well as other factors like privacy loss and data conversion capacity, maximizing the benefits for both data producers and consumers, and enhancing the incentive and monitoring capabilities in cross-domain collaborations.

II. Related Work

A. Access Control

Ravidas et al. [6] discussed the advantages and disadvantages of various access control models. Ying et al. [7] proposed an efficient fine-grained access control framework focusing on reliable policy updates in cloud data sharing to enhance the security and efficiency of data sharing. Meng et al. [8] introduced an SDN-based gateway to address identity theft attacks. It provided a firewall mechanism that ensured only legitimate service consumers or IoT devices could access the user's virtual machine. To overcome the vulnerability of storing the system private key in each registered entity and prevent bypass attacks, Zhang et al. [9] designed a secure and efficient access control protocol for video multicast in SDVN. This protocol guaranteed anonymous access to video content by multicast member vehicles and facilitated the tracking of malicious vehicles. Additionally, Fugkeaw [10] utilized Attribute Certificates (ACs) to support authorization within the CP-ABE scheme, addressing the key update issue when a large number of users accessed shared data in the cloud. This approach enabled efficient management of attribute-based revocation in large-scale systems. In recent years, the emergence of blockchain as a distributed technology has spurred research on distributed architectures. Blockchain can be leveraged to establish trusted identity authentication and authorization mechanisms, eliminate single points of failure, and enhance scalability [11], [12]. Hao et al. [13] proposed an architecture based on a lightweight consortium blockchain, leveraging smart contracts to facilitate cross-domain access control and employing a token accumulation mechanism for trust assessment of access requesting nodes. Guo et al. [14] proposed a blockchain-based attribute data sharing scheme for 6G-enabled VANETs, incorporating a multi-tree structure for efficient batch user revocation and joining. To protect privacy in multi-user access to Electronic Health Records (EHRs), Wang et al. [15] employed an Attribute-Based Encryption (ABE) scheme with constant ciphertext size on the blockchain, using searchable indexes in smart contracts to reduce storage overhead on peer nodes. However, in SCNs, which involve large amounts of data, numerous participating enterprises, and strict real-time processing requirements, the configuration of roles and permissions tends to remain relatively consistent. While ABE offers fine-grained access control and is widely applied in scenarios requiring complex access policies [16], its high computational overhead and large ciphertext sizes limit its applicability in resource-constrained environments [17], and its tree-based access structure is not suitable for scenarios in which access policies are frequently updated. In contrast, our proposed approach combines ACLs, PRE, and smart contracts to provide a lightweight and efficient solution. By circumventing the performance bottlenecks of ABE, our mechanism reduces computational and storage burdens while maintaining robust security, making it particularly well suited for the real-time data processing needs of cross-domain collaborations. Our approach ensures efficient data sharing without compromising security, offering a practical and scalable alternative in this context.

B. Security and Privacy of Data Sharing

Yugha et al. [18] explored the possible technologies and security protocols to be adopted in future IoT, and proposed key measures to ensure data security and privacy. Alwarafy et al. [19] analysed the security and privacy issues in edge computing-assisted IoT, and discussed the challenges faced in distributed computing environments as well as the potential solutions. Hathaliya et al. [20] delved into security and privacy issues in the context of Healthcare 4.0. Karati et al. [21] proposed a certificateless signature scheme based on bilinear pairings to ensure data integrity in industrial Internet of Things (IoT) systems. Li et al. [22] introduced a hierarchical multi-group data sharing framework for cloud-assisted industrial IoT, implementing group signatures and Merkle Hash Trees to protect the integrity of shared data from compromise by the cloud. In recent years, there has been a growing effort to integrate blockchain technology into industrial IoT for secure data sharing [23], [24]. Tian et al. [25] addressed the security and efficiency issues when outsourcing data to network storage services by proposing a secure duplicate data deletion and shared auditing scheme based on blockchain and a dual-server storage model. This scheme protected users from data loss due to single points of failure and duplicate forgery attacks. Considering the untrusted cloud and malicious auditing, Zhang et al. [26] achieved efficient authentication of source Personal Health Records (PHRs) from Health-IoT devices, ensuring the credibility of shared PHRs

in the cloud and the correctness of search results through the adoption of the BLS algorithm for blockchain-based verification code storage and data integrity auditing. Rehman et al. [27] introduced blockchain technology into cloud-based service provider systems, utilizing smart contracts to enhance system performance and throughput while mitigating the risks of malicious activities in the network. Jiang et al. [28] proposed an industrial IoT data sharing scheme based on the consortium blockchain and edge computing, addressing trust issues among industrial IoT entities while acknowledging the need for further improvement in privacy protection for shared data. Lu et al. [29] addressed the challenge of massive IoT data by presenting a data sharing scheme that combines Attribute-Based Encryption and blockchain. The encrypted data was stored in the IPFS network, achieving a balance between fine-grained access control and efficient file sharing. Yeh et al. [30] integrated an editable blockchain with the existing revocable IPFS mechanism and developed an enhanced proxy re-encryption scheme, effectively improving security with moderate overhead. Wang et al. [31] designed an anonymous authentication algorithm and combined broadcast encryption and proxy re-encryption techniques to achieve secure and flexible cross-domain data sharing. Li et al. [32] studied the relationship between capacity and security in blockchain from the perspective of block propagation and forking problems in on-demand distributed service scenarios. Our scheme employs IPFS for off-chain storage and integrates group signature, hash authentication, and access control smart contract based on ACLs to ensure data security and integrity while improving the scalability of the system.

C. Incentive Mechanisms

The research on incentive mechanisms aims to encourage data sharing among participants, facilitating the effective utilization and rational flow of data. The design of incentive mechanisms using technologies such as blockchain and game theory to promote data sharing is currently a popular research area [33], [34], [35], [36], [37]. Bi et al. [38] introduces a Bayesian game-based privacy framework for protecting personalised services for social IoT environments. Zheng et al. [39] designed a joint online pricing and reward sharing mechanism aiming to enhance the efficiency and fairness of the mobile data market. Lu et al. [40] explored the implementation of decentralised and automated incentive mechanisms in ride-hailing services in order to prevent price discrimination and improve service transparency. Gao et al. [41] models the interaction between IoT service providers and data analytics providers using game theory and explores the potential for its application to decision making in Big Data analytics. To achieve fast and reliable information retrieval in edge environments, Huang et al. [42] proposed a Stackelberg game to simulate the interaction between producers and distributors/relays. However, their solution for verifying rogue distributors disguising themselves as data producers still has limitations. Liu et al. [43] proposed a blockchain-enhanced data market framework and an optimal pricing mechanism for the IoT, utilizing a two-stage Stackelberg game to maximize the profits of data consumers and market institutions. Wang et al. [44] employed contract theory and optimal control theory to consider different privacy compensations under information asymmetry and the Data Privacy as a Service (DPaaS) model, formulating optimal data pricing strategies in smart contracts. Yang et al. [45] introduced social externality in the subgame at the user level, making the market closer to reality, and discussed in detail the impact of market demand and price, social influence graph, and the number of buyers on the game outcomes. Unlike the application scenarios studied above, we consider the existence of different levels of cooperation among enterprises within the supply chain. Partners at varying levels can share data with corresponding degrees of confidentiality, enabling differentiated data sharing among SCN participants. Our scheme dynamically adjusts data pricing based on these relationships and other factors such as privacy loss and data conversion capacity, maximizing benefits for both data producers and consumers, and effectively incentivizing data sharing.

III. System Model

In the field of SCNs, inter-enterprise cross-domain data sharing plays a crucial role in enabling efficient collaboration and optimizing business processes. However, challenges such as data leakage, inadequate assessment of data value, and insufficient incentives for data sharing hinder the implementation and advancement of data sharing practices. To address these issues and ensure secure inter-enterprise data sharing, this paper proposes a model called Supply Chain Network Cross-Domain Data Sharing System (SCN-CDSM). The SCN-CDSM incorporates mechanisms such as encryption, access control, and smart contracts to guarantee the security of inter-enterprise data sharing. Building upon the SCN-CDSM, we design a sharing incentive mechanism based on the Stackelberg game. This mechanism takes into account the cooperation relationship of both participants involved and request information to make optimal data sharing decisions that maximize the benefits for both participants. The combined effect of the SCN-CDSM model and the incentive mechanism contributes to the improvement of inter-enterprise data sharing practices by addressing security concerns and promoting mutually beneficial collaborations.

Based on the operational characteristics of the supply chain, cross-domain data sharing in the SCN involves various participants, including suppliers, manufacturers, distributors, transporters, and customers. The level of cooperation between SCN participants is determined by factors such as the frequency of business transactions, business secrets, and credibility. Differential data sharing occurs among SCN participants, where different levels of data privacy can be shared among different participants. This concept is illustrated in the left half of Fig. 1. Meanwhile, the proposed SCN-CDSM encompasses multiple roles, such as department of enterprises (DoE), key generation center (KGC), blockchain server (BS), encryption authentication server (EAS), blockchain distributed ledger, and off-chain storage center. Similar to [46], these roles are categorized into three layers based on their functionalities, as depicted in the right half of Fig. 1. The following provides specific descriptions of each layer.

A. Entity Layer

The *Entity Layer* primarily consists of DoEs within an enterprise and their corresponding sets of data acquisition devices (DADs). These DADs (such as IoT sensors) are integrated into physical assets such as goods, containers, vehicles, etc., enabling the DoEs to monitor and track real-time location, status, and transportation conditions. This monitoring generates a significant volume of data. The data is then categorized into different datasets with varying levels of privacy for processing and

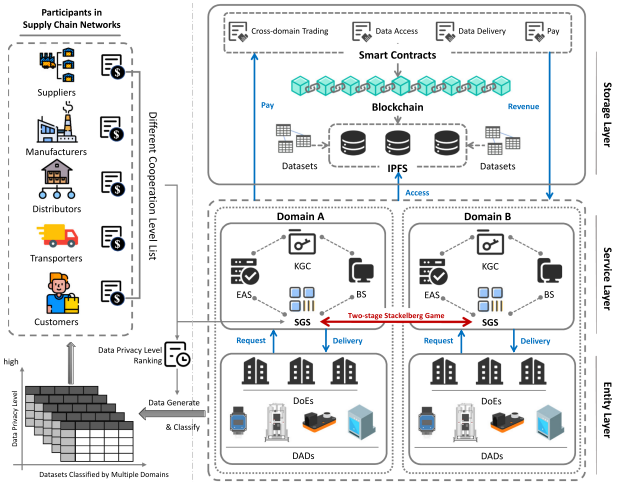


Fig. 1. System model of SCN-CDSM.

storage. Within an enterprise, multiple DoEs are interconnected, forming a "Domain". When a department within one domain needs to request data from a department in another domain, it must communicate with its affiliated domain *Service Layer* to facilitate data transactions and access.

B. Service Layer

The *Service Layer* consists of three types of servers that are responsible for cross-domain data access control.

Key Generation Center(KGC): Each domain has a unique KGC, which serves as a key management component. During domain initialization, the KGC generates domain public parameters and manages the keys of enterprise departments to ensure communication confidentiality and security. Additionally, the KGC collaborates with the Blockchain Server (BS) and the Encryption-Authentication Server (EAS) to facilitate authentication processes across domains.

Blockchain Server(BS): Each domain requires a node to maintain the blockchain ledger. The BS encapsulates specific data into transactions and writes them to blocks.

Encryption Authentication Server(EAS): In the cross-domain data sharing scenario of the SCN, DoEs need to prove their operating privileges for data delivery and access. The EAS is responsible for generating and verifying group signatures. It also performs proxy re-encryption to generate ciphertexts when the data contributed by DoEs needs to be encrypted and stored. These operations are typically performed collaboratively between the BS and the EAS.

C. Storage Layer

This layer is responsible for storing data and access control related information. Its main components are as follows.

Blockchain Distributed Ledger: The blockchain distributed ledger utilizes encryption techniques to safeguard data and enables distributed storage and processing. As a distributed technology, it offers advantages in terms of trust, reliability, and CIA properties (Confidentiality, Integrity, and Availability). In the context of cross-domain data sharing in the SCN, the consortium blockchain model is more suitable, considering requirements such as system decentralization and efficiency. The consortium blockchain involves collaborating peer nodes that collectively maintain the distributed ledger. Nodes in a consortium blockchain undergo verification before joining the network. Although nodes in a consortium blockchain do not fully trust each other, they collaborate under specific contracts and regulations. Trust relationships can be established among different enterprise domains through the consortium blockchain, with representative nodes assigned to each domain responsible for maintaining the global distributed ledger. These nodes participate in consensus to ensure data consistency and security. Key management information, transaction information, rules for data access, and access logs from each domain are encapsulated in blocks of the distributed ledger. These pieces of information are utilized in the cross-domain data sharing process.

Off-chain Storage Server: The storage limitations of blockchain technology need to be addressed, as the block size in a blockchain is limited, leading to throughput bottlenecks. To mitigate this issue, we introduces the Object Storage Service (OSS) to reduce the amount of data written on the blockchain. Real data is stored in OSS, while the blockchain distributed ledger stores the hash values of the data. In this paper, Inter-Planetary File System (IPFS) server nodes are employed for off-chain storage.

The SCN-CDSM model serves as the foundation for inter-enterprise collaboration and cross-domain data sharing within the SCN. When a department u^A from enterprise domain A initiates a cross-domain data sharing request with department u^B in the target data-producing enterprise domain B, u^B employs the Stackelberg game framework to simulate their interaction. This simulation takes into account the cooperation level between A and B as well as the request information. It determines the optimal decisions for data privacy level, unit price, and quantity, maximizing utility for both participants and incentivizing data sharing (see Section V for details). Based on the simulation results, u^B uploads the relevant data to the *Service Layer* within B. The data is encrypted and stored off-chain in IPFS. At this stage, the *Service Layer* invokes the cross-domain trading contract. The system access control module executes the corresponding access control process through the smart contract based on the data access request information. The necessary information is retrieved from the distributed ledger for access control verification. If authorization is granted, the system returns the content identifier (CID) of the data to the *Service Layer* of the data consumer department u^A in A. The *Service Layer* device retrieves the ciphertext of the data resource based on the CID, decrypts it, and returns it to u^A. After verifying the data information, u^A completes the signing of the cross-domain trading contract and pays the corresponding data fees.

Moreover, in the SCN-CDSM model, we assume that all departments are self-interested and rational, seeking to maximize their profits in the process of data sharing. Most departments follow the established incentive rules to obtain reasonable profits. However, there are a few rogue departments that seek to gain unfair advantages by cheating to obtain more profits. For example, data producers may unilaterally increase the price of data for sharing during the decision-making process. In our proposed model, the verification of data information by data consumers is crucial in the cross-domain data sharing process of the SCN. We assume that data consumers voluntarily or mandatorily inspect whether the data's privacy level, unit price, and quantity align with the optimal sharing decisions generated

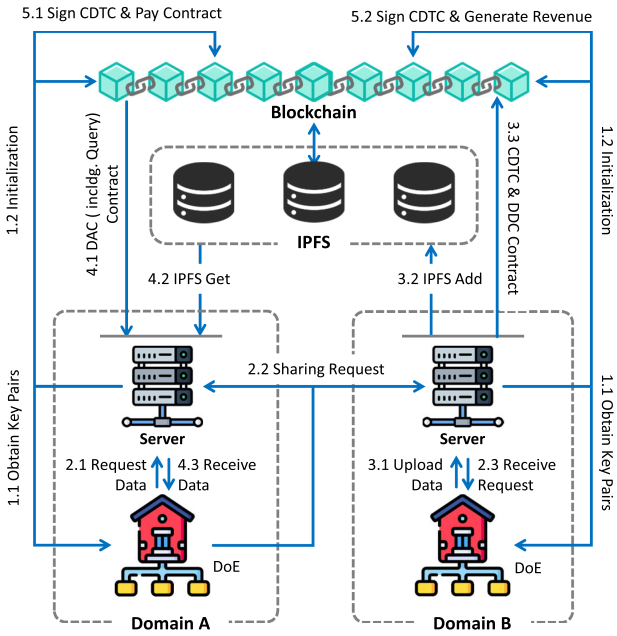


Fig. 2. Detailed processes of cross-domain data sharing in SCN.

TABLE I
DESCRIPTION OF SYMBOLS

Symbol	Description
DID	the identity of an enterprise domain
$(dsk,\ dpk)$	the public-private keypair of the enterprise domain
u	a department in the enterprise domain
$(upk,\ usk)$	the public-private keypair of the department
N	number of datasets held by the data producer
$\boldsymbol{DS_i}$	the i-th dataset held by the data producer
λ_i	privacy factor of the i-th dataset held by the data producer
λ_{min}	minimum shareable data privacy factor corresponding to cooperation level
λ_{max}	maximum shareable data privacy factor acceptable to data consumer
p_{max}	maximum data unit price acceptable to data consumer
D_{min}	minimum quantity of data acceptable to the data consumer
SELECT	SELECT statements for target data
$\boldsymbol{DS}^*$	shared dataset
D^*	quantity of data in shared dataset
p^*	data unit price of shared dataset
λ^*	privacy factor of shared dataset
CID	the identity of shared dataset
ACL	access control list
T	smart contract transaction
rk	the transform key of proxy re-encryption

by the incentive rules. The scenario where data consumers refuse to pay for eligible data is beyond the consideration of this paper.

IV. Detailed Design and Implementation

The detailed process of a DoE (data consumer) in Domain A requesting data sharing with a DoE (data producer) in Domain B is depicted in Fig. 2. Initially, the domain initialization process establishes a secure and encrypted environment, ensuring trustworthy and transparent information exchange. Subsequently, the sharing request process allows data consumers to request specific data tailored to their needs, enabling personalized data sharing and transactions while safeguarding data privacy. Following that, the data delivery process involves data producers uploading data meeting optimal sharing criteria to the *Service Layer* and triggering the cross-domain trading contract, while defining and managing access rules to ensure data resource security and privacy protection. Next, the data access process decrypts the data after verifying access permissions, making it available in its raw form, and delivers it to the data consumer. Finally, the data payment process involves finalizing the cross-domain trading contract through mutual agreement, validating the data, and executing the necessary payment. This process upholds fairness in data transactions, unlocks the value of the data, and fosters trust and cooperation among SCN enterprises. Detailed implementations of the major steps are provided below, and the symbols used are presented in Table I.

A. Domain Initialization

Service Layer devices within the domain need to generate public parameters and assign the necessary public-private key pairs for the domain itself and the departments it manages.

1) All departments and their *Service Layer* devices within an enterprise constitute a domain and determine its *DID*. The KGC within the domain can be regarded as the group manager in group signatures, responsible for generating domain public parameters related to group signatures. The EAS within the domain is responsible for generating domain public parameters related to proxy re-encryption.
2) The KGC runs the key generation algorithm to assign public-private key pairs $(dpk,\ dsk)$ and $(upk,\ usk)$ to the domain and its internal departments. The public-private key pair $(upk,\ usk)$ of each department is also used by the EAS in executing the proxy re-encryption scheme for encrypting and decrypting data.
3) The BS within the domain generates initialization transactions sent to the blockchain and stored in the distributed ledger, so that miners can later verify the domain's public key.

B. Data Sharing Request

We assume the data producer possesses N datasets divided by privacy factors: $\mathbf{DS} = \{\mathbf{DS}_1, \mathbf{DS}_2, \ldots, \mathbf{DS}_i, \ldots, \mathbf{DS}_N\}$, where the privacy factor of dataset $\boldsymbol{DS_i}$ is λ_i, $\lambda_i \in (0, 1)$. A smaller λ_i indicates a higher data security level but lower data utility value due to more anonymization. The minimum privacy factor $\lambda_{\min}$ that enterprises of different cooperation levels can share is different. When requesting cross-domain data, data consumers can specify requirements for the upper limits of unit price $p_{\max}$, low limits of data quantity $D_{\min}$, upper limits of privacy factor $\lambda_{\max}$, and data attributes SELECT statement, respectively. The request will be handled by the *Service Layer* of the requester's domain.

1) When department u^A from domain A requests data from department u^B in domain B, u^A generates a data request $Request = \langle u^B, p_{\max}, D_{\min}, \lambda_{\max}, \mathtt{SELECT} \rangle$ and uploads it to EAS^A. EAS^A performs encryption to encrypt the transaction request and obtains the ciphertext $Ttrade = Enc(dpk_B, Request)$. In addition, u^A needs to generate a group signature $\sigma \leftarrow Sign(dpk_A, usk_{u^A}, msg)$, where msg is the shared request information.

2) The blockchain server BS^A within domain A generates a request transaction $T = (DID^B, DID^A, Ttrade, Timestamp)$. The miner of the consortium blockchain needs to check the group signature σ, if $Verify(dpk_A, msg, \sigma)$ is $true$, then T will be packaged into a block.
3) Domain B decrypts $Ttrade$ using dsk_B and verifies the data request *Request* information. If verified successfully, BS^B sends *Request* and the minimum privacy factor threshold $\lambda_{\min}$ that domain B can share with domain A to u^B, and generates a shared request confirmation transaction $T = (DID^B, DID^A, Ttrade, VerifyResult, Timestamp)$ to be packaged into a block written to the distributed ledger.

C. Data Delivery

After the shared request confirmation, the data producer derives the optimal data sharing decision through simulating interactions with the data consumer (see Section V for details), and delivers the selected dataset $\boldsymbol{DS}^*$, data quantity D^*, data unit price p^* and privacy factor λ^* to the *Service Layer* of its domain. The ciphertext is sent to IPFS for off-chain storage. The *Service Layer* invokes the cross-domain trading contract, and the data access rules are written to the blockchain.

1) Data producer u^B derives the optimal data sharing decision through simulating interactions with the data consumer according to the set incentive rules, selecting the dataset with privacy factor λ^*. By querying with `SELECT` statement, it selects the data subset $\boldsymbol{DS}^*$ with data quantity D^* from it, along with the corresponding data quantity D^*, data unit price p^*, privacy factor λ^* and data attributes `SELECT` and uploads them to EAS^B. EAS^B then performs encryption $CT \leftarrow \text{Enc}(upk_{u^B}, \boldsymbol{DS}^*)$.
2) EAS^B uploads CT to the off-chain storage server OSS, and OSS stores the ciphertext in the IPFS, obtaining the hash of the data ciphertext corresponding to $\boldsymbol{DS}^*.CID$.
3) u^B determines which domain can access the data. Except the accessible *DID* in the access control list, EAS also generates a domain-specific conversion key $rk_{u^B \to DID^A} \leftarrow \text{ReKeyGen}(usk_{u^B}, dpk_A)$. In addition, u^B needs to generate a group signature $\sigma \leftarrow Sign(dpk_B, usk_{u^B}, msg)$, where msg is the information on data delivered.
4) BS^B invokes the smart contract CDTC (Cross Domain Trading Contract) to generate a two-party contract involving BS^A and BS^B after checking the σ, including the data hash of the shared dataset, data quantity, unit price and privacy factor. The miner of the consortium blockchain needs to check the group signature σ, if $Verify(dpk_B, msg, \sigma)$ is $true$, then the cross-domain data trading logic in the CDTC will continue to be executed, and BS^B generates a cross-domain data trading transaction $T = (DID^B, DID^A, Ttrade, DS^*.$ *CID*, $D^*, p^*, \lambda^*,$ *Timestamp*) to be packaged into a block.
5) BS^B invokes the smart contract DDC (Data Delivery Contract) to record the data hash and access rules on the distributed ledger. The miner of the consortium blockchain needs to check the group signature σ, if $Verify(dpk_B, msg, \sigma)$ is $true$, then the data delivery logic in the smart contract DDC will continue to be executed, and BS^B generates a data delivery transaction $T = (DID^B, \boldsymbol{DS}^*.CID, ACL, Timestamp)$ to be packaged into a block.

D. Data Access

The *Service Layer* of the data consumer's domain verifies the cross-domain data sharing transaction information and its own access permission for the required dataset, and obtains the conversion key rk. It performs re-encryption and decryption operations on the ciphertext to obtain the plaintext of the data, and sends the data plaintext to the data consumer.

1) u^A generates a group signature $\sigma \leftarrow Sign(dpk_A, usk_{u^A}, msg)$, where *msg* is the access request information. It then sends σ and *msg* to BS^A, BS^A checks whether the data privacy level, unit price and data quantity contained in the cross-domain data sharing transaction generated by the data producer conform to the optimal data sharing decision derived from the incentive rules; if they conform, the data access process continues; otherwise, BS^A will try to reinitiate the data sharing request process and refuse the data payment process, causing the data producer to suffer losses for delivering the data.
2) BS^A invokes the smart contract DAC (Data Access Contract) to access the data. The miner of the consortium blockchain needs to check the group signature σ, if $Verify(dpk_A, msg, \sigma)$ is $true$, then the data access logic in the smart contract DAC can continue.
3) The smart contract DAC invokes *Query* contract to query whether DID^A is in the *ACL* of $\boldsymbol{DS}^*$. If the query succeeds, the service devices of domain A obtain the corresponding conversion key $rk_{u^B \to DID^A}$. Then, BS^A generates a data access transaction $T = (DID^A, Ttrade, \boldsymbol{DS}^*.CID, ConfirmResult, Timestamp)$ to be packaged into a block.
4) EAS^A downloads the data ciphertext $\boldsymbol{DS}^*.CT$ from the off-chain storage server using $\boldsymbol{DS}^*.CID$, and then performs re-encryption using the conversion key $rk_{u^B \to DID^A}$ to obtain the ciphertext EAS^A can decrypt, that is, $\boldsymbol{DS}^*.CT^A \leftarrow ReEnc(rk_{u^B \to DID^A}, \boldsymbol{DS}^*.CT)$.
5) EAS^A decrypts $\boldsymbol{DS}^*.CT^A$ using its domain private key dsk_A to obtain the data $\boldsymbol{DS}^* \leftarrow Dec(dsk_A, \boldsymbol{DS}^*.CT^A)$, and returns it to the u^A within the domain.

E. Data Payment

After data delivery and access are completed, the data consumer verifies the accuracy of the data information and pays for the data.

1) u^A compares the received data with the information in the CDTC created by the u^B for verification. If the CDTC contains false data quantity, false unit price, false data attributes, etc., u^A has the right to refuse to sign, meaning the data producer will not get any income. u^A generates a group signature $\sigma_{u^A} \leftarrow Sign(dpk_A, usk_{u^A}, msg)$, where *msg* is the transaction confirmation information. It then sends σ_{u^A} and *msg* to BS^A. The miner of the consortium blockchain verifies the group signature by checking $Verify(dpk_A, msg, \sigma_{u^A})$ if is $true$.

2) u^B generates a $\sigma_{u^B} \leftarrow Sign(dpk_B, usk_{u^B}, msg)$, where *msg* is the transaction confirmation information. It then sends σ_{u^B} and *msg* to BS^B. The miner of the consortium blockchain verifies the group signature by checking $Verify(dpk_B, msg, \sigma_{u^B})$ if is $true$.
3) BS^A and BS^B respectively use σ_{u^A} and σ_{u^B} to sign the CDTC, making the contract take effect and stored on the blockchain for public verification, with other nodes confirming the change of credit of each node.
4) BS^A invokes the smart contract PC (Pay Contract) to pay for the data. The miner of the consortium blockchain needs to verify the group signature by checking $Verify(dpk_A, msg, \sigma_{u^A})$ if is $true$, where *msg* is the transaction confirmation information. When σ_{u^A} is valid, the data payment logic in the smart contract PC can continue. BS^A generates a data payment transaction $T = (DID^B, DID^A, Ttrade, \boldsymbol{DS}^*.CID, Timestamp)$ to be packaged into a block.

F. Security and Privacy Threats

The inter-enterprise collaboration and data sharing in SCNs are promising work scenarios, where openness and interconnectivity between multiple enterprise domains can significantly reduce administrative costs and greatly increase productivity. However, as information flows between enterprises, security and privacy concerns become important.

Sybil Attack: A Sybil attack is when a malicious participant creates a large number of false identities (Sybil identities) and attempts to use these false identities to send misleading information to the SCN cross-domain data sharing system in order to disrupt the normal operations and decision-making process of the supply chain.

Message Replacement Attacks: A replacement attack is when a malicious attacker may try to intercept data in the SCN, tamper with its content, and pass the forged data to the receiver. This behavior may cause the receiver to make a wrong decision that the forged messages are from a legitimate sender, which can negatively impact the entire supply chain.

Single Point of Failure: A single point of failure means that if one service node fails, the entire system will be inoperative, which does not meet the goal of high availability and reliability. A malicious attacker may launch a large-scale DoS/DDoS attack against a service node to make its services unavailable by exceeding its processing capacity in terms of the number of requests, a situation that may have a serious impact on data exchange between SCN participants, leading to delays, errors, or disruptions in the supply chain.

Privacy Protection: In some application scenarios that require high privacy protection, the identities of the participants should be protected. And in the process of data sharing among SCN enterprises, high-classified data may only be shared with partners with high trust and close cooperation, while low-classified data can be shared with a wider range of partners.

V. Stackelberg Game Based Incentive Mechanism

The cross-domain data sharing in SCNs suffers from inefficiency and fairness due to isolation and data permission restrictions between different enterprise domains. By establishing a data sharing incentive mechanism, it can help each department to consider its own profit while integrating the requirements and restrictions of the other departments to optimize decision making, enhance transparency, and establish a sustainable collaborative relationship. In this section, we first formulate the differential data sharing problem and propose a Stackelberg game to model the interaction between data producers and data consumers. We then derive analytical solutions to the proposed model. The parameters used in the expressions are listed in Table II.

TABLE II
Parameters Used in the Expressions

Parameter	Description
$D_i^{\max}$	the quantity of data owned by the i-th dataset after SELECT retrieval
c	a single data transmission cost
D_i	the quantity of data purchased by the data consumer in the i-th dataset
D_i^*	optimal quantity of data purchased by the data consumer in the i-th dataset
p_i	the unit price of data sold by the data producer in the i-th dataset
p_i^*	optimal unit price of data sold by the data producer in the i-th dataset
i^*	Serial number of shared dataset
r	data conversion capacity factor
μ	privacy leakage threat factor
$\mathbf{N}$	the set of datasets held by the data producer
PoP	profit function of the data producer
PoC	profit function of the data consumer

A. Overview of the Problem

In the scenario of cross-domain data sharing in SCNs, when a department in Domain A is interested in the data of another department in Domain B, it can initiate a proxy transaction through the *Service Layer* of its domain. In this process, the department in A can propose requirements for the upper limits of data unit price $p_{\max}$, low limits of data quantity $D_{\min}$, upper limits of privacy factor $\lambda_{\max}$, and data attributes SELECT, respectably. And the lower bound of privacy factor for data privacy levels that B can share with A is $\lambda_{\min}$. The data producer has N datasets divided by privacy factors, represented by the set $\mathbf{N} = \{1, 2, \ldots, i, \ldots, N\}$. The data producer hopes to select a suitable dataset with appropriate privacy level for sharing and determine the unit price of data, while the data consumer hopes to decide the quantity of data to purchase. To determine the optimal decision of data privacy level, data unit price and data quantity to be shared, we formulate the problem as a two-stage Stackelberg game, where the optimal data sharing decision derived from the incentive rules is encoded in the smart contract, and the transaction and data information can be publicly verified and accepted, with the credit of each participant being updated accordingly.

For each data producer, we define its profit from sharing the ith dataset as follows:

$$\mathrm{PoP}_i\left(p_i, D_i\right) = p_i D_i - c D_i - \mu \ln\left(\frac{1}{\lambda_i}\right) D_i, \qquad (1)$$

where, p_i and D_i represent the unit price of data sold by the data producer and the quantity of data purchased by the data consumer in the ith dataset, respectively, and $p_i D_i$ is the revenue. cD_i is the transmission cost when delivering data with a data quantity of D_i, c is a single data transmission cost. Inspired by [42], we represent the privacy loss as $\mu \ln(\frac{1}{\lambda_i}) D_i$, where μ measures the threat of data leakage.

For each data consumer, its profit comes from the benefits gained by converting the data minus the cost of paying for the data, which is defined as follows:

$$\mathrm{PoC}_i\left(p_i, D_i\right) = r\ln\left(1+\lambda_i D_i\right) - p_i D_i, \tag{2}$$

where, due to the differences in D_i and λ_i, the benefit that can be obtained after processing the data is also different, inspired by [43], we define the benefit of data conversion as $r\ln(1+\lambda_i D_i)$, and r measures the data conversion capability. $p_i D_i$ denotes the cost of paying for the data.

In our hypothetical data sharing scenario, the data producer determines the unit price p_i for each dataset $\boldsymbol{DS_i}$ in order to maximize the shared profit of each dataset. Furthermore, in the selection of the final shared dataset, the data producer strives to maximize its own profit, resulting in the emergence of a subgame among different datasets. Following this, the data consumer determines the quantity of data to purchase from the shared dataset of the data producer, denoted as D_i, with the objective of maximizing its own profit.

B. Stackelberg Game Model

We assume the participating departments are self-interested and rational agents that aim to maximize their individual profits from data sharing transactions. To achieve win-win outcomes under this setting, we conceptualize the problem as a two-stage Stackelberg game with the following definition:

- *Leader:* Data producer.
- *Follower:* Data consumer.
- *Strategies:* The data producer determines a suitable dataset with appropriate privacy factor and the data unit price: i^* and p^*. The data consumer determines the data quantity D^*.
- *Payoff:* The objective is to maximize the profits for the data producer and consumer, denoted as $\mathrm{PoP}_{i^*}(p^*, D^*)$ and $\mathrm{PoC}_{i^*}(p^*, D^*)$, respectively.

Specifically, the data producer first chooses the optimal dataset and pricing strategy, and the data consumer formulates the optimal data purchase strategy based on the data producer's decision. The data producer updates its strategy after receiving the response from the data consumer, and then provides it to the data consumer again. This process repeats until the data producer and data consumer are unwilling to change the equilibrium. The solution of the game can be the optimal strategy combination of the leader and follower.

Definition 1: Stackelberg Equilibrium. The outcome $\{i^*,\ p^*,\ D^*\}$ of this two-stage Stackelberg game reach the equilibrium if the data producer and data consumer simultaneously satisfy the following conditions:

$$\mathrm{PoP}_{i^*}\left(p^*, D^*\right) \geq \mathrm{PoP}_j\left(p_j, D_j^*\right),\ \forall j \in \mathbf{N}. \tag{3}$$

$$\mathrm{PoC}_{i^*}\left(p^*, D^*\right) \geq \mathrm{PoC}_{i^*}\left(p^*, D_{i^*}\right). \tag{4}$$

where i^*, p^*, D^* are the optimal value for i, p_i, D_i respectively.

The profit of the data producer is maximized while the data consumer adopts the optimal strategy, which represents the equilibrium of the Stackelberg game in this specific scenario.

C. Equilibrium Analysis

When analyzing the equilibrium of the Stackelberg game, we employ the backward induction method, which is key to solving sequential games by analyzing each stage of the game in reverse order. This game consists of a sequence of subgames between the data producer and the data consumer. We represent this non-cooperative game as $G = \{\mathbf{N},\ \{p_i\}_{i\in\mathbf{N}},\ \{\mathrm{PoP}_i(p_i, D_i)\}_{i\in\mathbf{N}}\}$. Each subgame corresponds to a dataset and is played independently by data producers and data consumers. Data producers set prices for datasets with different privacy levels and data consumers respond. The backward induction method allows us to begin by determining the optimal response of the data consumer (Stage II) to a given price, and then use this to derive the data producer's pricing decision (Stage I). By working backward from the consumer's decision, we can ensure that each player's strategy is logically consistent with the others, systematically addressing the equilibrium of the game. This method provides a clear, step-by-step modeling of the interactions between the leader (data producer) and the follower (data consumer).

Stage II: Follower Gaming. Given the data unit price p_i set by the data producer for the ith dataset, the data consumer aims to maximize its profit by determining the optimal purchasing strategy D_i^*. Specifically, we define the formulation of the data consumer as follows.

$$\max_{i\in\mathbf{N}}\ \mathrm{PoC}_i\left(p_i, D_i\right), \tag{5}$$

$$\text{subject to } D_i \in [D_{\min},\ D_i^{\max}]. \tag{6}$$

We derive the first-order and second-order derivatives of the data consumer's profit function in (2) with respect to D_i, which can be written as follows.

$$\frac{\partial \mathrm{PoC}_i}{\partial D_i} = \frac{r\lambda_i}{1+\lambda_i D_i} - p_i, \tag{7}$$

$$\frac{\partial^2 \mathrm{PoC}_i}{\partial {D_i}^2} = -\frac{r{\lambda_i}^2}{\left(1+\lambda_i D_i\right)^2} < 0. \tag{8}$$

The derivative (8) indicates that $\mathrm{PoC}_i(p_i, D_i)$ is a strictly convex function, so the optimal solution can be directly obtained. By solving $\frac{\partial \mathrm{PoC}_i}{\partial D_i} = 0$, we derive the optimal response function of the data consumer as follows.

$$D_i^* = \frac{r}{p_i} - \frac{1}{\lambda_i}. \tag{9}$$

Stage I: Leader Gaming. Based on the optimal purchasing strategies determined by the data consumer for each dataset in the second stage, the data producer strives to determine a pricing strategy that maximizes its own profit. Specifically, we define the formulation of the data producer as follows.

$$\max_{i\in\mathbf{N}}\ \mathrm{PoP}_i\left(p_i, D_i\right), \tag{10}$$

$$\text{subject to } p_i \in [c,\ p_{\max}], \lambda_i \in [\lambda_{\min},\ \lambda_{\max}]. \tag{11}$$

Substituting (9) into (1), the profit of the data producer from sharing the ith dataset can be rewritten as follows.

$$\begin{aligned}\mathrm{PoP}_i\left(p_i, D_i^*\right) &= p_i D_i^* - cD_i^* - \mu\ln\left(\frac{1}{\lambda_i}\right) D_i^* \\ &= \left[p_i - c - \mu\ln\left(\frac{1}{\lambda_i}\right)\right]\left(\frac{r}{p_i} - \frac{1}{\lambda_i}\right).\end{aligned} \tag{12}$$

We derive the first-order and second-order derivatives of the data producer's profit function in (12) with respect to p_i, which can be written as follows.

$$\frac{\partial \mathrm{PoP}_i}{\partial p_i} = -\frac{1}{\lambda_i} + \frac{\mathrm{rc}}{{p_i}^2} + \mu\ln\left(\frac{1}{\lambda_i}\right)\frac{r}{{p_i}^2}, \tag{13}$$

$$\frac{\partial^2 \mathrm{PoP}_i}{\partial {p_i}^2} = -2r\frac{\mu\ln\left(\frac{1}{\lambda_i}\right) + c}{{p_i}^3} < 0. \tag{14}$$

The derivative (14) indicates that $\mathrm{PoP}_i(p_i, D_i^*)$ is a strictly convex function, so the optimal solution can be directly obtained. By solving $\frac{\partial \mathrm{PoP}_i}{\partial p_i} = 0$, we derive the optimal pricing strategy of the data producer as follows.

$$p_i^* = \sqrt{r\lambda_i \left[c + \mu ln\left(\frac{1}{\lambda_i}\right)\right]}. \tag{15}$$

Based on the proposed data sharing incentive mechanism in this paper, the data producer performs a subgame of N datasets to select the i^*th dataset that maximizes its profit. We are able to infer the profit attained by data producer, which is calculated as follows:

$$\begin{aligned}\mathrm{PoP}_{i^*}(p^*, D^*) &= \max_{i\in\mathbf{N}} \mathrm{PoP}_i(p_i^*, D_i^*) \\ &= \left[\left(\sqrt{r\lambda_{i^*}\left(c + \mu \ln\left(\frac{1}{\lambda_{i^*}}\right)\right)} - c - \mu \ln\left(\frac{1}{\lambda_{i^*}}\right)\right)\right. \\ &\quad \left.\times \left(\frac{r}{\sqrt{r\lambda_{i^*}\left(c + \mu \ln\left(\frac{1}{\lambda_{i^*}}\right)\right)}} - \frac{1}{\lambda_{i^*}}\right)\right]. \end{aligned} \tag{16}$$

The data consumer determines its optimal data purchase strategy based on the i^*th dataset provided by the data producer and its profit is calculated as follows:

$$\begin{aligned}\mathrm{PoC}_{i^*}(p^*, D^*) &= r \ln\left(1 + \lambda_{i^*}\left(\frac{r}{p_{i^*}^*} - \frac{1}{\lambda_{i^*}}\right)\right) \\ &\quad - \sqrt{r\lambda_{i^*}\left(c + \mu \ln\left(\frac{1}{\lambda_{i^*}}\right)\right)}\left(\frac{r}{p_{i^*}^*} - \frac{1}{\lambda_{i^*}}\right). \end{aligned} \tag{17}$$

After the above game equilibrium analysis, the data producer determines to share the data subset of data quantity D^* selected by SELECT statement from dataset $\boldsymbol{DS}^*$ (i.e., $\boldsymbol{DS}_{i^*}$), then uploads data quantity D^* (i.e., $D_{i^*}^*$), data unit price p^* (i.e., $p_{i^*}^*$) and privacy factor λ^* (i.e., λ_{i^*}) into the smart contract CDTC (Cross-Domain Trading Contract) for analysis and verification by the data consumer. The impact of data transmission cost c, privacy leakage threat factor μ and data conversion capability factor r on the game equilibrium will be analyzed in the next section.

VI. Experimental Results and Analysis

A. Security and Privacy Analysis

The experimental analysis begins with a detailed security evaluation, followed by performance evaluation and scalability analysis. This order is chosen to reflect the foundational importance of security in SCN data sharing systems. Security is a critical prerequisite for ensuring the confidentiality, integrity, and authenticity of shared data, particularly in cross-enterprise collaborations. By establishing the security robustness of the proposed SCN-CDSM model, we provide a solid foundation for subsequent evaluations of computational efficiency and scalability. This sequence mirrors the logical development process of the model, ensuring that performance improvements are assessed in the context of a secure and reliable system.

First, we employ game-based proofs and other techniques to demonstrate the overall security of our proposed scheme.

Theorem 1: The proposed scheme can prevent the identities of SCNs cross-domain data sharing participants from being maliciously obtained by other non-participating domains.

Proof: This scheme employs group signature technology [47] to ensure the anonymity of the signer, allowing other domains to verify the signature using the domain's public key dpk without revealing the actual identity of the signing department.

During the *Domain Initialization* phase, the *Service Layer* server KGC generates the domain's public key $dpk = (n, e, G, g, a, \lambda, \epsilon)$, where (n, e) is the RSA-generated public key pair, G is a cyclic group, g is its generator, n is the order of G, a is a large multiplicative factor of the two prime factors of n, λ represents the private key length of departments, and ϵ is a constant greater than 1. The KGC retains the domain's private key pair $dsk = (n, d)$ generated by RSA.

Department i uses its private key $usk_i \in_R \{0, \ldots, 2^\lambda - 1\}$ and calculates $(y = a^{usk_i} \bmod n, z = g^y)$, and then send it to the EAS within the domain. Using a zero-knowledge proof, the department demonstrates that it knows the discrete logarithm of y to the base a, thereby proving ownership of the private key usk_i. As a result, the department obtains a membership certificate $v_i = (y+1)^d \bmod n$. The EAS stores (y, z) for future verification and tracking of group signatures.

Department i signs a message m to produce the group signature $\sigma(m) = (\tilde{g} = g^r, \tilde{z} = \tilde{g}^y, V_1 = \mathrm{SKLOGLOG}[\alpha : \tilde{z} = \tilde{g}^{a^\alpha}](m), V_2 = \mathrm{SKROOTLOG}[\beta : \tilde{z}\tilde{g} = \tilde{g}^{\beta^*}](m), r \in_R \mathbb{Z}_n^*)$. Any *Service Layer* servers or miners in the consortium blockchain can verify the validity of the group signature using proofs of knowledge for double discrete logarithms (SKLOGLOG) and eth roots of discrete logarithms (SKROOTLOG). However, they cannot identify the specific department who signed the message, thereby protecting the anonymity of participating departments.

In addition, the proxy re-encryption technology in this scheme only requires the servers in the *Service Layer* to participate in the generation of the conversion key, which can protect the identity of the department in the *Entity Layer* during the encryption calculation and data transmission process and prevent any possible identity leakage. □

Theorem 2: If the shared participant identities in the proposed scheme are unforgeable, accessing data purchased by other departments and tampering with data uploaded to the *Service Layer* by other departments becomes difficult, and large-scale DoS/DDoS attacks do not cause system crashes, then no malicious node can pass the system's security verification with a non-negligible probability.

Proof: We prove this theorem through a series of games as follows.

Game 0: In the initialization process, a challenger department $\mathcal{C}$ and the in-domain *Service Layer* devices have established long-term trust relationships, obtaining key pairs distributed from the in-domain KGC to enter the SCNs cross-domain data sharing system. All public parameters generated when the domain is initialized are registered through the distributed ledger of the consortium blockchain to ensure the legitimacy of the identity. The security verification of the system is lossless.

Game 1: This game is the same as *Game* 0, except that the challenger department $\mathcal{C}$ attempts to impersonate as a legitimate department using its own generated key pair. EAS discovers that the department has not registered a member certificate, the honest service layer refuses to interact, and $\mathcal{C}$ will terminate the game and declare failure.

Game 2: This game is the same as *Game* 1, except that the challenger department $\mathcal{C}$ attempts to intercept access to data purchased by other departments, infringing on the interests of data consumer departments. According to the public-private keypair generation rules: $sk \leftarrow \mathbb{Z}_p, \quad pk = g^{sk}$, g is the generator of the elliptic curve G, p is a large prime number and the order of G. EAS uses the proxy re-encrypted conversion key $\mathrm{rk}_{\mathrm{uB}\rightarrow DID^{\mathrm{A}}} = usk_{u^B}/Hash((dpk_A,)^{usk_{u^B}}$ to re-encrypt the data $\boldsymbol{DS}^*.CT = (g^r, CT \cdot g^{r\cdot usk_{u^B}})$ downloaded from IPFS and obtain the $\boldsymbol{DS}^*.CT^A = ((g^r)^{\mathrm{rk}_{\mathrm{uB}\rightarrow DID^{\mathrm{A}}}}, CT \cdot g^{r\cdot usk_{u^B}})$, where r is a random number. The honest EAS will use the domain secret key dsk_A to decrypt the data, and send the data ciphertext that can only be decrypted by the private key usk_{u^A} to the data consumer department u^A according to the *Request* generated in the *Data Sharing Request* phase. Although $\mathcal{C}$ cannot decrypt the data ciphertext after intercepting it, $\mathcal{C}$ will terminate the game and declare failure.

Game 3: This game is the same as *Game* 2, except that the challenger department $\mathcal{C}$ attempts to tamper with data uploaded to the *Service Layer* by other departments to interfere with the decision-making of data consumer departments and also maliciously reduce the credibility of its affiliated domain. $\mathcal{C}$ intercepts and modifies data uploaded to the *Service Layer* by other departments in its affiliated domain during the data delivery process. *Service Layer* of its affiliated domain check the group signatures generated by the data-producing departments and find that the information on data delivered does not match the uploaded data, so it terminate the invocation of smart contract CDTC, and $\mathcal{C}$ will terminate the game and declare failure.

Game 4: This game is the same as *Game* 3, except that the challenger department $\mathcal{C}$ attempts to massively tamper with or access data to cause a single point of failure in the system. The partially decentralized architecture of the consortium chain means that a single point of failure in the *Service Layer* server BS will not affect the entire system. Moreover, the *Service Layer* server EAS can be distributed on different network nodes, and the services it provides, such as key verification and proxy re-encryption, can be completed by calling smart contracts with the assistance of BS, which means that even if $\mathcal{C}$ successfully prevents a server from providing services, it cannot prevent the normal operation of the service layer. Therefore, $\mathcal{C}$ will terminate the game and declare failure." □

Theorem 3: The proposed scheme can realize measurable and customized data sharing while protecting data privacy.

Proof: This scheme leverages differential privacy techniques to protect the privacy of data-generating departments while retaining statistical features and removing personal characteristics. Given a dataset D and a query mechanism $\mathcal{M}$, the privacy guarantee for any output set S is defined as: $\Pr[\mathcal{M}(D_1) \in S] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(D_2) \in S] + \delta$, where ϵ represents the privacy budget, and δ denotes the failure probability. Graded privacy perturbation is applied to the dataset, where each data point D_i is perturbed by adding noise $Noise(\lambda_i)$. Based on the cooperation levels within the SCNs, shareable data is categorized into different privacy levels. The intensity of the noise is proportional to the privacy level λ_i, as expressed by: $PrivacyPerturbation(D_i) = D_i + Noise(\lambda_i)$.

Additionally, this scheme allows data consumer departments to evaluate the data transformation benefits of datasets at different levels according to their specific needs, balancing data utility and privacy protection levels. Departments can select datasets with higher utility value that meet high-level expectations. Furthermore, by enabling payment after the completion of data verification, the scheme prioritizes the interests of data consumer departments. □

Second, specific potential vulnerabilities related to smart contracts are critical to the functionality of the system and require a more thorough examination of potential attack vectors and the development of a strong mitigation strategy. Based on our analysis, we have identified two primary vulnerabilities:

Timestamp Dependency Issue. Due to the distributed nature of the blockchain platform, achieving perfect time synchronization across all nodes is nearly impossible. This could allow an attacker to manipulate the '`block.timestamp`' variable, leading to logical attacks on contracts that rely on time-sensitive operations. To address this issue, we recommend using an on-chain oracle or other more reliable time sources for obtaining time information. Additionally, contract design should minimize time-sensitive operations or implement fault-tolerant mechanisms to handle potential timestamp inconsistencies.

Information and Function Exposure. Given the transparent nature of the blockchain, smart contract code and data are publicly visible, which can allow attackers to analyze and exploit vulnerabilities. To avoid abuse or misuse, we recommend following the principle of least privilege, granting only the minimum permissions necessary for contract execution. Additionally, visibility modifiers (e.g., '`public`', '`private`', '`internal`') should be used to restrict access to sensitive information and functions, limiting their exposure.

B. Experiment Setup

The experimentation is conducted in an Ubuntu 18.04 test environment, utilizing hardware comprising a 13th Gen Intel(R) Core(TM) i5-13500H 2.60 GHz processor and 16 GB RAM. Simulation experiments are conducted using a multidomain system model based on Hyperledger Fabric v2.3.3 [48]. In this model, each organization (org) within the Fabric network represents a domain, and the Peer node within an org represents the intra-domain blockchain server (BS) in the multidomain system model.

In this study, we conduct blockchain performance testing using Caliper [49], a service designed to generate workloads for a specific system under test (SUT) while continuously monitoring their responses. Caliper enables users to utilize customized benchmark artifacts as inputs, including transaction definitions, load patterns, chaincode deployment, blockchain network configurations, and more, which are essential for interacting with the SUT. By leveraging Caliper, users can define realistic benchmark artifacts aligned with their business logic and performance metrics, facilitating accurate and repeatable performance measurements.The key parameters in the Benchmark artifacts are as follows.

- test.workers.number: This parameter specifies the number of worker processes utilized to execute the workload. Caliper consists of two distinct services/processes: a manager process and multiple worker processes, ensuring scalability. Worker processes independently generate the workload for the system under test (SUT). Even if a worker process reaches the capacity of its host machine, employing additional worker processes can further augment the workload rate of Caliper.

TABLE III
CONFIGURATION AND SIMULATION PARAMETER

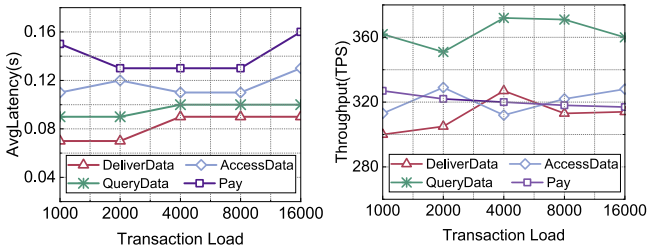
Parameter	Value [Unit]	Parameter	Value [Unit]
workers.number	2	$p_{\max}$	10
txDuration	20 [s]	$D_{\min}$	20
rateControl	fixed-load	$D_i^{\max}$	500
Endorsement	major	λ_i	$\sim \mathcal{N}(0,1)$
N	50000	μ	0.01
$\lambda_{\max}$	0.9	c	0.1
$\lambda_{\min}$	0.1	r	300

- test.rounds[i].txDuration: This parameter defines the duration of the ith round of testing in seconds. It allows for specifying the duration of each test round, thereby controlling the time range and duration of the performance testing.
- test.rounds[i].rateControl: This parameter denotes the rate control policy employed during the ith test round, with the option to specify it as fixed-load. Fixed-load rate controllers are utilized to drive the test with a target load, maintaining a defined backlog of transactions within the system. This controller adjusts the number of transactions executed per second (TPS) to sustain the desired workload. The achieved maximum TPS reflects the system's capability to maintain the load of pending transactions.

We conducted two sets of comparison experiments. In the first set, we tested the average latency and TPS for data delivery, access, query, and payment under different fixed load numbers in a 2-domain environment. In the second set, we extended the consortium blockchain network from 2 to 3, 4, 5, and 6 domains. We then tested the average latency and TPS for data delivery, access, query, and payment under different fixed load numbers for each configuration. For the experimental setup, each domain included 1 Peer node. Transaction validation required the participation of at least half of the members from different channels to execute and validate the transaction. This allowed organizations joining the channel to automatically adhere to the chain code endorsement policy. The configuration parameters of the Caliper and the simulation parameter settings of the cross-domain data sharing incentive mechanism based on Stackelberg game are shown in Table III.

C. Model Performance Evaluation

To evaluate the computational efficiency of the proposed ACLs+PRE access control mechanism, we conducted comparative experiments with ABE [50]. The experiments were performed on a server with a 4-core CPU and 8 GB of RAM. The results indicate that ABE incurs substantial computational overhead, particularly during key generation. For instance, generating a user key with one attribute requires 582 ms, while generating a key with four attributes increases this time to 1287 ms. Additionally, creating a three-layer access control tree for four attributes takes 941 ms, and the decryption process requires 683 ms. In contrast, the ACLs+PRE mechanism is significantly more efficient. The data producer encrypts the data in 550 ms, while the EAS generates the proxy re-encryption key in 384 ms. Re-encryption by the data consumer's service layer takes 44 ms, and decryption requires only 36 ms. The data consumer only needs to use their private key to decrypt the data, without the need for generating additional keys.



(a) Average Latency in 2-domain (b) TPS in 2-domain

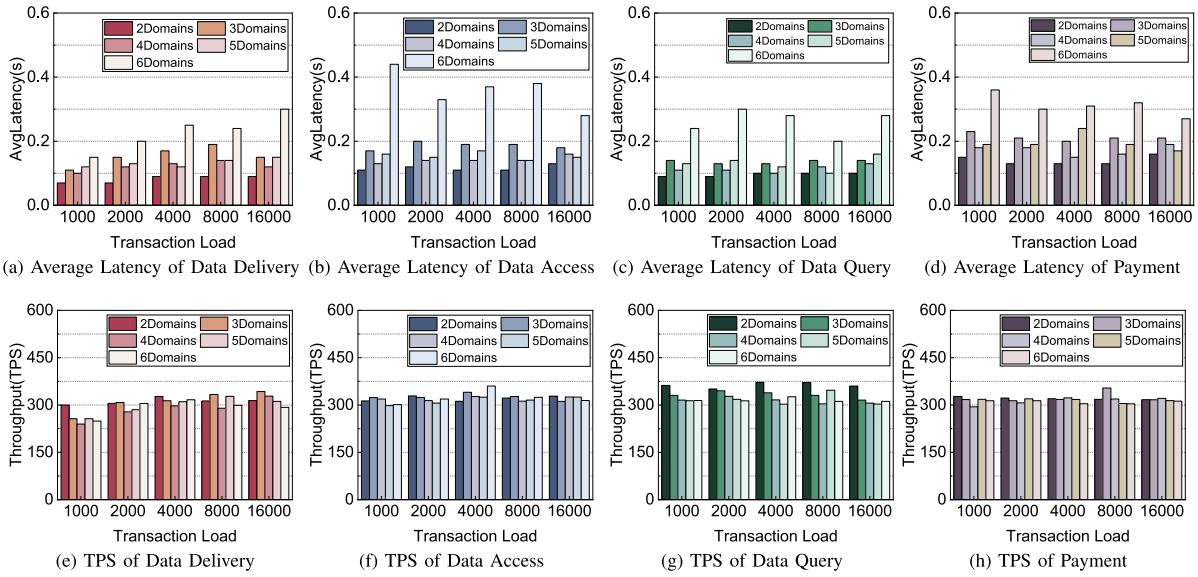
Fig. 3. Performance of chain code operations under different loads in two domains.

Fig. 3 displays the average latency and TPS of data delivery, data access, data query and payment under different transaction loads (transaction loads are 1000, 2000, 4000, 8000, 16000 respectively) in an environment of 2 domains. We can see that with the increase of transaction load, the average latency and TPS of the four types of chaincode operations maintain good stability. In addition, we can find that the TPS of data query operations is higher and the average latency is lower than that of data access operations. This is reasonable because query operations do not involve updating the state of the ledger. The current experimental results demonstrate that the system can be scaled up nearly linearly under increasing loads, i.e., maintaining sufficient and stable processing power while guaranteeing low response latency.

Fig. 4 exhibits the average latency and TPS for data delivery, data access, data query, and payment under varying transaction loads (1000, 2000, 4000, 8000, 16000) when extending the consortium blockchain network from 2 domains to 3, 4, 5, and 6 domains. It is evident that as the transaction load increases, the average latency for the four chaincode operations also increases significantly. This is due to the "major" endorsement policy set in the experiment. In a network with N nodes, the "major" policy requires at least $N/2+1$ nodes to endorse a transaction, meaning that a transaction is only acknowledged and enters the blockchain consensus process when endorsed by a majority of nodes. The consensus time in Hyperledger Fabric increases with the number of endorsing nodes. Hence, the observed latency increase is a result of executing the endorsement policy. It is noteworthy that although there is a clear upward trend in latency, the decline in throughput for the four chaincode operations is minimal and remains relatively stable. This indicates that the system can mitigate the impact of increased consensus time resulting from network expansion by allocating more computing resources. It is important to note that the performance of Hyperledger Fabric is influenced by various factors, including hardware, network environment, and block parameters. We believe that with high-performance hardware support and fine-tuning of underlying chain parameters, both throughput and latency can be further improved to meet the performance requirements of future SCN applications based on blockchain.

D. Incentive Mechanism Assessment

To examine the impact of various parameters on the effectiveness of the proposed Stackelberg game-based incentive mechanism, we construct the following hypothetical scenario: the data producer possesses N datasets with varying levels of privacy, and the data consumer submits data requests based on their



(a) Average Latency of Data Delivery (b) Average Latency of Data Access (c) Average Latency of Data Query (d) Average Latency of Payment

(e) TPS of Data Delivery (f) TPS of Data Access (g) TPS of Data Query (h) TPS of Payment

Fig. 4. Performance of chain code operations under different loads in multi-domain environment.

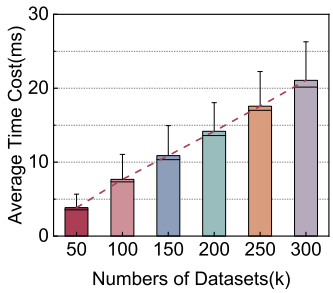
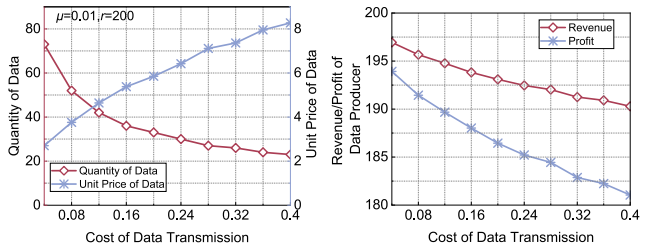


Fig. 5. Average time cost to finalize the optimal shared decision for different number of datasets.



(a) Data Quantity and Unit Price (b) Revenue & Profit of Data Producer

Fig. 6. Impact of data transmission costs c.

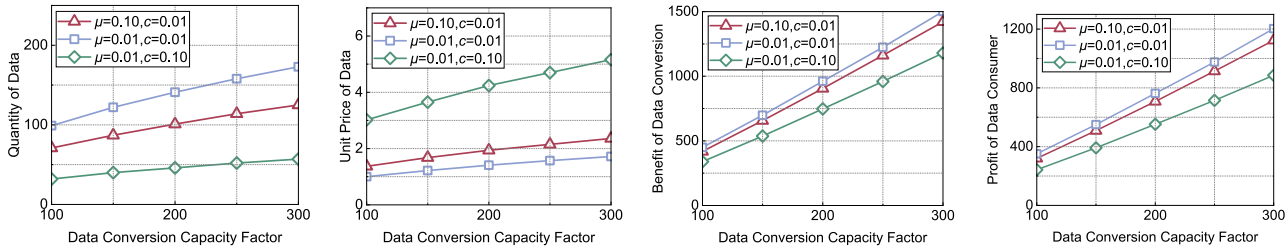
requirements. The default parameter values for the numerical experiments are as follows: $\lambda_{\max} = 0.9, \lambda_{\min} = 0.1, p_{\max} = 10, D_{\min} = 20, N = 50000, \{D_i^{\max} = 500\}_{i \in \boldsymbol{N}}$. We assume that $\{\lambda_i\}_{i \in \boldsymbol{N}}$ follow a normal distribution within (0,1). All numerical simulations are repeated 500 times, and the mean values are used as the final results.

Fig. 5 shows the impact of the data producer's dataset quantity on the efficiency of the sharing incentive mechanism. As the number of datasets available for sharing increases, the time required for the data producer to identify the dataset with the highest benefit exhibits a linear growth trend. This can be attributed to the rise in computational complexity for value mining and decision-making as the dataset scale expands.

Fig. 6 demonstrates the influence of data transmission cost c. We set $\mu = 0.01$ and $r = 200$. As the transmission costs are initially low, data consumers have access to a large quantity of data. However, as the costs increase, the available quantity of data for consumers decreases significantly. Once the cost surpasses a certain threshold, consumers only purchase the minimum amount of data required. Furthermore, as the costs continue to rise, the rate of unit price increase slows down. Meanwhile, the revenue of data producers declines as transmission costs escalate. Notably, the decline in producer profits is nearly twice as rapid as the decline in revenues. This trend highlights the importance of strategic measures to mitigate costs and safeguard economic interests, as lower transmission costs enable greater data acquisition for consumers and higher revenue for producers.

Fig. 7 depicts the utility of data consumers. A higher μ results in a reduced quantity of data acquired by data consumers, along with higher data unit price, which negatively impacts their total profits. The increase in c can also significantly decrease the quantity of data obtained by data consumers while simultaneously increasing the data unit price. The r of data consumers plays a critical role in maximizing their profits from data sharing. Data consumers with strong data conversion capabilities can effectively leverage shared data resources to generate higher profits. Enhancements in security and privacy measures for data sharing, reductions in the system's data transmission costs, and improvements in data conversion capabilities are essential for increasing the total profits of data consumers and fostering sustainable and mutually beneficial supply chain collaboration.

Fig. 8 reveals the utility of data producers. While the increase in the μ has a relatively minor impact on the revenue of data producers, it significantly reduces their total profits. Hence, data producers must prioritize and uphold data security in sharing to ensure privacy protection. The escalation of c can affect the revenue of data producers and diminish their total profits. To ensure economic benefits, data producers should strive to reduce

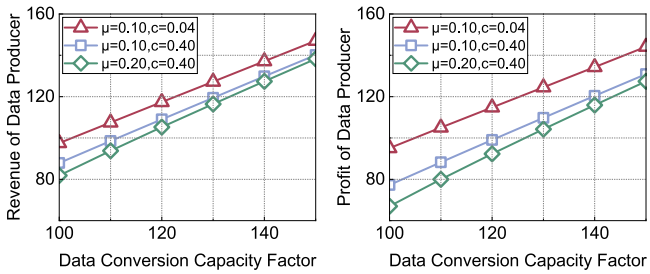


(a) Data Quantity (b) Unit Price (c) Benefit of Data Conversion (d) Profit of Data Consumer

Fig. 7. Trends of data consumer's utility.

TABLE IV
SYSTEM FEATURE COMPARISON AMONG VARIOUS MECHANISMS

	[11]	[14]	[29]	[30]	[31]	[42]	Our Scheme
Decentralisation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Access Control	KP-ABE	CP-ABE	CP-ABE	TA+PRE	BPRE	No	ACL+ PRE
Storage Scalability	Weak	Strong	Strong	Strong	Strong	No Sure	Strong
Shared Incentives	No	No	No	No	No	Yes	Yes
Experimental Setup	Fabric&3 Org	Fabric&5 PN	Fabric&2 Org	Ethernet&3 PC	Fabric&3 Org	Numerical Simulation	Fabric&(2-6) Org
Limitations	ABE settings use a lot of on-chain storage	Lack of analysis of attacks by bad nodes in the changing VANETs environment	Lack of solution for distributing, updating, or revoking ABE keys	No analysis of new security risks from editable blockchains	No analysis of different security policies between domains	No testing of the on-chain system's performance	Incentives based on business partnerships need a risk assessment



(a) Revenue of Data Producer (b) Profit of Data Producer

Fig. 8. Trends of data producer's utility.

costs in the data transmission process, thereby achieving higher profit margins and maintaining competitiveness. The revenue and profits of data producers are positively correlated with the r of data consumers. Therefore, collaboration between data producers and data consumers can lead to a mutually beneficial outcome and foster the advancement of supply chain data markets.

E. Comparison With Existing Approaches

In light of our research and analysis, directly comparing the experimental results of our proposed model with those of existing studies presents significant challenges. These challenges stem from several factors, such as differences in experimental environments, system architectures, and the performance metrics employed in the literature. Many existing approaches are designed for specific domains or application scenarios, such as vehicular ad-hoc networks (VANET), cloud computing, or industrial Internet of Things (IIoT), which differ substantially from SCNs environment considered in our study. Furthermore, some solutions rely on simulation-based evaluations with varying assumptions, making direct performance comparisons infeasible. To facilitate a meaningful comparison, we provide a qualitative analysis of the key features and design choices across relevant studies [11], [14], [29], [30], [31], [42]. Table IV summarizes the main distinctions between our proposed SCN-CDSM and existing blockchain-based data sharing schemes. This comparison highlights the differences in aspects such as decentralization, access control mechanisms, storage scalability, and incentive structures. As shown in Table IV, our approach offers a more efficient and scalable solution for cross-domain data sharing in SCNs, especially in terms of real-time data processing and security. Unlike many existing studies that focus on specific domains or employ more computationally-intensive mechanisms like ABE, our model strikes a balance between security and computational efficiency, making it particularly well-suited for dynamic, large-scale SCNs. While direct experimental comparisons may not be feasible due to differences in system design and evaluation methods, this qualitative comparison underscores the practical advantages of our scheme in addressing the unique challenges posed by SCNs.

VII. DISCUSSION

The experimental results demonstrate that our proposed scheme is highly scalable and capable of addressing the complexities of future supply chain environments. However, this study has certain limitations that warrant further exploration in future research.

Real-World Validation: While our experiments focus on system performance in a simulated environment, future work should validate the model's practicality and scalability in real-world supply chain environments.

Handling Malicious Behavior: The current model assumes that data consumers will adhere to incentive rules; however, real-world scenarios may involve malicious behaviors, such as refusal to pay or tampering with data. Future research should explore solutions such as penalty mechanisms, reputation systems, or smart contract-based arbitration mechanisms to address these issues.

Dynamic Pricing Models: Our current incentive mechanism classifies data into different privacy levels based on cooperative relationships. Future work could incorporate market dynamics in real supply chain environments, such as dynamic pricing models that account for data sensitivity, legal regulations, and market supply and demand.

VIII. Conclusion

This paper presents a cross-domain data sharing scheme for SCNs utilizing consortium blockchain technology. The scheme aims to enhance supply chain collaboration by establishing trust relationships among enterprises and enabling seamless cross-domain data sharing. Data storage is implemented outside the blockchain to improve efficiency and protect privacy, and privacy protection measures such as group signatures, access control lists, and differential sharing are employed. Additionally, a two-stage Stackelberg game is devised to balance SCN data sharing participants' interests and provide incentives for data sharing. The proposed scheme is evaluated through extensive numerical simulations, demonstrating its security and efficiency.

References

[1] J. Peng, L. Chen, and B. Zhang, "Transportation planning for sustainable supply chain network using Big Data technology," *Inf. Sci.*, vol. 609, pp. 781–798, 2022.

[2] T. Jiang, Y.-S. Lin, and T. Nguyen, "Market equilibrium in multi-tier supply chain networks," *Nav. Res. Logistics*, vol. 69, no. 3, pp. 355–370, 2022.

[3] V. Hassija, V. Chamola, V. Gupta, S. Jain, and N. Guizani, "A survey on supply chain security: Application areas, security threats, and solution architectures," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6222–6246, Apr. 2021.

[4] W. Powell et al., "From premise to practice of social consensus: How to agree on common knowledge in blockchain-enabled supply chains," *Comput. Netw.*, vol. 200, 2021, Art. no. 108536.

[5] W. Klibi, A. Martel, and A. Guitouni, "The design of robust value-creating supply chain networks: A critical review," *Eur. J. Oper. Res.*, vol. 203, no. 2, pp. 283–293, 2010.

[6] S. Ravidas, A. Lekidis, F. Paci, and N. Zannone, "Access control in internet-of-things: A. survey," *J. Netw. Comput. Appl.*, vol. 144, pp. 79–101, 2019.

[7] Z. Ying, W. Jiang, X. Liu, S. Xu, and R. H. Deng, "Reliable policy updating under efficient policy hidden fine-grained access control framework for cloud data sharing," *IEEE Trans. Serv. Comput.*, vol. 15, no. 6, pp. 3485–3498, Nov./Dec. 2022.

[8] Y. Meng, Z. Huang, G. Shen, and C. Ke, "SDN-based security enforcement framework for data sharing systems of smart healthcare," *IEEE Trans. Netw. Serv. Manage.*, vol. 17, no. 1, pp. 308–318, Mar. 2020.

[9] X. Zhang, H. Zhong, J. Cui, C. Gu, I. Bolodurina, and L. Liu, "AC-SDVN: An access control protocol for video multicast in software defined vehicular networks," *IEEE Trans. Mobile Comput.*, vol. 22, no. 10, pp. 5657–5674, Oct. 2023.

[10] S. Fugkeaw, "Secure data sharing with efficient key update for industrial cloud-based access control," *IEEE Trans. Serv. Comput.*, vol. 16, no. 1, pp. 575–587, Jan./Feb. 2023.

[11] M. Xiao, Q. Huang, Y. Miao, S. Li, and W. Susilo, "Blockchain based multi-authority fine-grained access control system with flexible revocation," *IEEE Trans. Serv. Comput.*, vol. 15, no. 6, pp. 3143–3155, Nov./Dec. 2022.

[12] S. Pal, A. Dorri, and R. Jurdak, "Blockchain for IoT access control: Recent trends and future research directions," *J. Netw. Comput. Appl.*, vol. 203, 2022, Art. no. 103371.

[13] X. Hao, W. Ren, Y. Fei, T. Zhu, and K.-K. R. Choo, "A blockchain-based cross-domain and autonomous access control scheme for Internet of Things," *IEEE Trans. Serv. Comput.*, vol. 16, no. 2, pp. 773–786, Mar./Apr. 2023.

[14] Z. Guo, G. Wang, Y. Li, J. Ni, and G. Zhang, "Attribute-based data sharing scheme using blockchain for 6G-enabled VANETs," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 3343–3360, Apr. 2024.

[15] M. Wang, Y. Guo, C. Zhang, C. Wang, H. Huang, and X. Jia, "MedShare: A privacy-preserving medical data sharing system by using blockchain," *IEEE Trans. Serv. Comput.*, vol. 16, no. 1, pp. 438–451, Jan./Feb. 2023.

[16] A. Miglani and N. Kumar, "BloomACS: Bloom filter-based access control scheme in blockchain-enabled V2G networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 10636–10651, Sep. 2024.

[17] Q. Zhang, Y. Fu, J. Cui, D. He, and H. Zhong, "Efficient fine-grained data sharing based on proxy re-encryption in IIoT," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 6, pp. 5797–5809, Nov./Dec. 2024.

[18] R. Yugha and S. Chithra, "A survey on technologies and security protocols: Reference for future generation IoT," *J. Netw. Comput. Appl.*, vol. 169, 2020, Art. no. 102763.

[19] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, "A survey on security and privacy issues in edge-computing-assisted Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4004–4022, Mar. 2021.

[20] J. J. Hathaliya and S. Tanwar, "An exhaustive survey on security and privacy issues in healthcare 4.0," *Comput. Commun.*, vol. 153, pp. 311–335, 2020.

[21] A. Karati, S. H. Islam, and M. Karuppiah, "Provably secure and lightweight certificateless signature scheme for IIoT environments," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3701–3711, Aug. 2018.

[22] T. Li, J. Zhang, Y. Shen, and J. Ma, "Hierarchical and multi-group data sharing for cloud-assisted industrial Internet of Things," *IEEE Trans. Serv. Comput.*, vol. 16, no. 5, pp. 3425–3438, May 2023.

[23] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, and Y. Bian, "Blockchain security: A survey of techniques and research directions," *IEEE Trans. Serv. Comput.*, vol. 15, no. 4, pp. 2490–2510, Jul./Aug. 2022.

[24] L. D. Xu, Y. Lu, and L. Li, "Embedding blockchain technology into IoT for security: A survey," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10452–10473, Jul. 2021.

[25] G. Tian et al., "Blockchain-based secure deduplication and shared auditing in decentralized storage," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 6, pp. 3941–3954, Nov./Dec. 2022.

[26] J. Zhang, Y. Yang, X. Liu, and J. Ma, "An efficient blockchain-based hierarchical data sharing for healthcare Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 7139–7150, Oct. 2022.

[27] M. Rehman, N. Javaid, M. Awais, M. Imran, and N. Naseer, "Cloud based secure service providing for IoTs using blockchain," in *Proc. 2019 IEEE Glob. Commun. Conf.*, 2019, pp. 1–7.

[28] Y. Jiang, Y. Zhong, and X. Ge, "IIoT data sharing based on blockchain: A multileader multifollower stackelberg game approach," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4396–4410, Mar. 2022.

[29] X. Lu, S. Fu, C. Jiang, and P. Lio, "A fine-grained IoT data access control scheme combining attribute-based encryption and blockchain," *Secur. Commun. Netw.*, vol. 2021, pp. 1–13, 2021.

[30] L.-Y. Yeh, W.-H. Hsu, and C.-Y. Shen, "GDPR-compliant personal health record sharing mechanism with redactable blockchain and revocable IPFS," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 4, pp. 3342–3356, Jul./Aug. 2024.

[31] F. Wang, J. Cui, Q. Zhang, D. He, and H. Zhong, "Blockchain-based secure cross-domain data sharing for edge-assisted industrial Internet of Things," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 3892–3905, 2024.

[32] Y. Li, L. Liang, Y. Jia, W. Wen, C. Tang, and Z. Chen, "Blockchain for data sharing at the network edge: Trade-off between capability and security," *IEEE/ACM Trans. Netw.*, vol. 32, no. 3, pp. 2616–2630, Jun. 2024.

[33] M. Shen, J. Duan, L. Zhu, J. Zhang, X. Du, and M. Guizani, "Blockchain-based incentives for secure and collaborative data sharing in multiple clouds," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1229–1241, Jun. 2020.

[34] P. K. R. Maddikunta et al., "Incentive techniques for the Internet of Things: A survey," *J. Netw. Comput. Appl.*, vol. 206, 2022, Art. no. 103464.

[35] C. Chi, Y. Wang, X. Tong, M. Siddula, and Z. Cai, "Game theory in Internet of Things: A survey," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12125–12146, Jul. 2022.

[36] S. Arisdakessian, O. A. Wahab, A. Mourad, H. Otrok, and M. Guizani, "A survey on IoT intrusion detection: Federated learning, game theory, social psychology, and explainable AI as future directions," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 4059–4092, Mar. 2023.

[37] H. Si, C. Sun, Y. Li, H. Qiao, and L. Shi, "IoT information sharing security mechanism based on blockchain technology," *Future Gener. Comput. Syst.*, vol. 101, pp. 1028–1040, 2019.
[38] R. Bi, Q. Chen, L. Chen, J. Xiong, and D. Wu, "A privacy-preserving personalized service framework through Bayesian game in social IoT," *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–13, 2020.
[39] Z. Zheng, Y. Peng, F. Wu, S. Tang, and G. Chen, "ARETE: On designing joint online pricing and reward sharing mechanisms for mobile data markets," *IEEE Trans. Mobile Comput.*, vol. 19, no. 4, pp. 769–787, Apr. 2020.
[40] Y. Lu, Y. Qi, S. Qi, Y. Li, H. Song, and Y. Liu, "Say no to price discrimination: Decentralized and automated incentives for price auditing in ride-hailing services," *IEEE Trans. Mobile Comput.*, vol. 21, no. 2, pp. 663–680, Feb. 2022.
[41] Y. Gao, L. Chen, G. Wu, Q. Li, and T. Fu, "A game theory study of Big Data analytics in Internet of Things," *IEEE Trans. Netw. Serv. Manage.*, vol. 20, no. 2, pp. 1707–1716, Jun. 2023.
[42] Y. Huang, Y. Zeng, F. Ye, and Y. Yang, "Profit sharing for data producer and intermediate parties in data trading over pervasive edge computing environments," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 429–442, Jan. 2023.
[43] 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 Things J.*, vol. 6, no. 6, pp. 9748–9761, Dec. 2019.
[44] Y. Wang et al., "SPDS: A secure and auditable private data sharing scheme for smart grid based on blockchain," *IEEE Trans. Ind. Inform.*, vol. 17, no. 11, pp. 7688–7699, Nov. 2021.
[45] Z. Yang, K. Liu, Y. Chen, W. Chen, and M. Tang, "Two-level stackelberg game for IoT computational resource trading mechanism: A smart contract approach," *IEEE Trans. Serv. Comput.*, vol. 15, no. 4, pp. 1883–1895, Jul./Aug. 2022.
[46] M. Shen et al., "Blockchain-assisted secure device authentication for cross-domain industrial IoT," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 942–954, May 2020.
[47] B. Libert, S. Ling, K. Nguyen, and H. Wang, "Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors," in *Proc. 35th Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2016, pp. 1–31.
[48] Hyperledger. Hyperledger-Fabric-2.3.3. Accessed: Jul. 05, 2023. [Online]. Available: https://github.com/hyperledger/fabric/releases/tag/v2.3.3
[49] Hyperledger. Hyperledger-Caliper-0.4.2. Accessed: Jul. 05, 2023. [Online]. Available: https://github.com/hyperledger/caliper/tree/v0.4.2
[50] H. Wee, "Optimal broadcast encryption and CP-ABE from evasive lattice assumptions," in *Proc. 41st Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2022, pp. 217–241.

Runqun Xiong (Member, IEEE) received the PhD degree in computer science from Southeast University. He was with the European Organization for Nuclear Research as a research associate for the AMS-02 experiment from 2011 to 2012. He is currently an associate professor with the School of Computer Science and Engineering, Southeast University, China. His research interests include Internet of Things, Cyber-Physical Systems, and Wireless Networks. He is a member of the ACM and CCF.

Jing Cheng received the BS degree in software engineering from the Hefei University of Technology, in 2022. She is currently working toward the master's degree majoring in Software Engineering with Southeast University. Her research interests include Blockchain, Internet of Things, and Game Theory.

Xirui Dong received the BS degree in computer science and technology from Southeast University, in 2019. She is currently working toward the PhD degree with the School of Computer Science and Engineering, Southeast University. Her research interests include Internet of Things, Wireless Networks, and Low-Power Wide-Area Network.

Jiahang Pu received the BS degree in software engineering from Southeast University, in 2023. He is currently working toward the master's student majoring in software engineering with Southeast University. His research interests include Network Security and Privacy, and the Internet of Things.

Feng Shan (Member, IEEE) received the PhD degree in computer science from Southeast University, China, in 2015. He was a visiting student with the School of Computing and Engineering, University of Missouri-Kansas City, Kansas City, MO, USA, from 2010 to 2012. He is currently an associate professor with the School of Computer Science and Engineering, Southeast University. His research interests include the areas of Internet of Things, Wireless Networks, Swarm Intelligence, and Algorithm Design and Analysis. He is a member of the ACM, and CCF.