International Journal of Software and Informatics, ISSN 1673-7288
http://www.ijsi.org, ijsi@iscas.ac.cn, +86-10-62661048
IJSI, 2021, 11(3): 287–313, doi:10.21655/ijsi.1673-7288.00264

International Journal of Software and Informatics

Research Article

Efficient Blockchain-Empowered Data Sharing Incentive Scheme for Internet of Things

Ting Cai (蔡婷)[1,2], Hui Lin (林晖)[1,2], Wuhui Chen (陈武辉)[1,2], Zibin Zheng (郑子彬)[1,2], Yang Yu (余阳)[1,2]

[1] (School of Data and Computer Science, Sun Yat-Sen University, Guangzhou 510006, China)
[2] (National Engineering Research Center of Digital Life (Sun Yat-Sen University), Guangzhou 510006, China)
Corresponding author: Wuhui Chen, chenwuh@mail.sysu.edu.cn

Abstract In recent years, with many devices continuously joining the Internet of Things (IoT), data sharing as the main driver of the IoT market has become a research hotspot. However, the users are reluctant to participate in data sharing due to security concerns and lacking incentive mechanisms in the current IoT. In this context, blockchain is introduced into the data sharing of IoT to solve the trust problem of users and provide secure data storage. However, in the exploration of building a secure distributed data sharing system based on the blockchain, how to break the inherent performance bottleneck of blockchain is still a major challenge. For this reason, the efficient blockchain-based data sharing incentive scheme is studied for IoT. In the scheme, an efficient data sharing incentive framework based on blockchain is proposed, named ShareBC. Firstly, ShareBC uses sharding technology to build asynchronous consensus zones that can process data sharing transactions in parallel and deploy efficient consensus mechanisms on the cloud/edge servers and asynchronous consensus zones in sharding, thus improving the processing efficiency of data sharing transactions. Then, a sharing incentive mechanism based on a hierarchical data auction model implemented by a smart contract is presentedto encourage IoT users to participate in data sharing. The proposed mechanism can solve the problem of multi-layer data allocation involved in IoT data sharing and maximize the overall social welfare. Finally, the experimental results show that the proposed scheme is economically efficient, incentive-compatible, and real-time, with scalability, low cost, and good practicability.

Keywords blockchain; data sharing; sharding; incentive mechanism; IoT

Citation Cai T, Lin H, Chen WH, Zheng ZB, Yu Y. Efficient blockchain-empowered data sharing incentive scheme for Internet of Things, *International Journal of Software and Informatics*, 2021, 11(3): 287–313. http://www.ijsi.org/1673-7288/264.htm

---

This is the English version of the Chinese article “区块链赋能的高效物联网数据激励共享方案. 软件学报, 2021, 32(4): 953–972. doi: 10.13328/j.cnki.jos.006229”.
Funding items: Key-area Research and Development Program of Guangdong Province (2019B020214006); National Natural Science Foundation of China (62032025, 61802450); NSFC-Guangdong Joint Fund Project (U20A6003); Program for Guangdong Introducing Innovative and Entrepreneurial Teams (2017ZT07X355); Pearl River Talent Recruitment Program (2019QN01X130); Science and Technology Research Program of Education Commission, Chongqing Municipal (KJZD-K201802401)
Received 2020-09-13; Revised 2020-10-26; Accepted 2020-12-19; IJSI published online 2021-09-25

In the wake of advances in 5G and mobile cloud computing technology, data sharing is playing an increasingly important role in the development and application of the Internet of Things (IoT). This is because most of the underlying deployment of the IoT applications are based on data sharing[1]. According to the statistics based on big data, the number of IoT devices is expected to soar to 41.6 billion by 2025, implying that 127 new devices will be connected to the Internet per second worldwide. IoT devices generate about 500 million bytes of data per day, and the number is expected to reach 79.4 zettabytes by 2025[2]. Such an massive amount of data will be shared and analyzed between IoT devices, inevitably leading to an ultra-large data transaction market[3, 4]. However, the current IoT data market is far from meeting that expectation. On the one hand, practical data sharing usually consumes resources and increases the costs of data sharing participants[5–9]. In the absence of an effective incentive strategy for participants, it is difficult to balance interests between multiple parties. Hence, most IoT users are reluctant to share data or forward messages. On the other hand, massive sensing data (such as locations) are vulnerable to the risk of personal privacy leakage. Such a security issue hinders IoT users from joining the data sharing market.

A good way is to encourage users to take the initiative to participate in IoT data sharing through an effective incentive mechanism. So far, there has been some research on introducing an incentive mechanism to IoT application scenarios such as crowd sensing or resource transactions to encourage users to participate in data sharing[10–13]. For example, Gao *et al*.[10] proposed a truthful incentive mechanism for the vehicle-based non-deterministic ad-hoc network. Pu *et al*.[11] studied an incentive-based hybrid edge computing framework for large-scale vehicular crowd sensing applications. Petrov *et al*.[12] designed an incentive mechanism based on opportunistic crowd sensing over NarrowBand IoT (NB-IoT). However, most of these approaches are centralized, facing security challenges in IoT applications that do not guarantee data integrity and are not trusted[14]. For example, in IoT, the data server may be attacked by malicious users or service providers, and the data stored in the server may be tampered with. Unscrupulous IoT users may provide false or even malicious data for their interest or illegal purposes[15].

Considering that the security challenge is extremely important, blockchain, owing to its inherent security properties, such as decentralization, anonymity, traceability, and non-tampering, is introduced into IoT data sharing to solve the trust problem of IoT users and provide secure data storage[16]. Substantial research on blockchain-based IoT data sharing has been put forward and implemented. For example, Kang *et al*.[17] proposed a blockchain-enabled Internet of vehicle data sharing scheme by optimizing consensus mechanisms. Yu *et al*.[18] proposed a crypto currency LRCoin based on Bitcoin, whose core idea is to design a leakage-resilient digital signature scheme for data transactions in the IoT to improve the security of data transactions. Yang *et al*.[19] designed a blockchain-based trust management system using the Bayesian inference model. These studies tried to solve the problem of data sharing in IoT by leveraging the blockchain technology. However, while constructing a blockchain-based secure distributed sharing system, they ignored the key performance bottleneck inherent in blockchain[20]. For example, the maximum transaction throughput of Bitcoin is about seven transactions per second, and the client that creates the transaction has to wait for at least 10 minutes on average to ensure that the transaction is chained; the maximum throughput of Ethereum is limited to 20 transactions per second, with an average latency of 12 seconds. By contrast, a centralized payment system, such as Visa, is usually able to complete transactions within several seconds, with a throughput reaching 10,000 transactions per second[21, 22]. The performance bottleneck of blockchain has become another important factor impeding IoT users' participation in data sharing. Therefore, to use the blockchain technology to facilitate the potential large-scale data sharing market comprising billions of IoT devices, we must improve its

performance as much as possible while maintaining its security and decentralization properties. It is urgent to study and propose an efficient blockchain-empowered IoT data sharing incentive scheme.

To solve the above problem, this paper first proposes an efficient blockchain-empowered IoT data sharing incentive framework, called ShareBC. In this framework, ShareBC introduces sharding technology[23] to divide the IoT devices into several Asynchronous Consensus Zones (ACZs). Specifically, it processes, in parallel, transaction verification in various sharding ACZs, which usually requires the joint action of all nodes in the system. The above steps aim to enhance the transaction processing ability of the blockchain-empowered data sharing system. In addition, according to the properties of consortium blockchain and IoT data sharing[24], this paper designs an efficient consensus process for ShareBC. The consortium blockchain committee relies on a set of distributed ACZs while maintaining complete control over data sharing transactions. Such a consensus mechanism is open and can be audited, with superiority in computation cost and scalability. Second, to encourage IoT users to participate in data sharing, this paper also proposes a sharing incentive mechanism based on the hierarchical data auction model implemented by a smart contract to maximize the overall social welfare of all participants. In practice, data sharing among IoT devices is usually subjected to the limitations caused by a multi-layer communication network[25]. Therefore, our mechanism designs a three-layer data auction model including data agents and corresponding data allocation and pricing rules, and it considers the impact of data transmission cost on social welfare. Finally, the mechanism is enforced in the form of a smart contract, which ensures the non-repudiation and execution efficiency of auction rules in data sharing transactions.

The main contributions of this paper are summarized as follows:

(1) It proposes an efficient blockchain-empowered IoT data sharing incentive scheme framework, i.e., ShareBC. To improve the transaction processing capability of the system, ShareBC introduces sharding and outlines the sharding construction steps of the blockchain-empowered IoT data sharing. In addition, ShareBC deploys an efficient consensus mechanism in the ACZs and cloud/edge servers, avoiding the high computational cost caused by the traditional block generation based on the workload proof and improving the efficiency of consensus-generated blocks.

(2) It introduces a sharing incentive mechanism based on hierarchical data auction model implemented by a smart contract. To ensure that the maximum number of IoT devices can participate in data sharing, we propose a three-layer data auction model based on the proposed ShareBC framework. In the framework, data sharing resources can be accessed indirectly by communication-constrained low-level devices via data agents to maximize social welfare.

(3) It develops a prototype system. For the simpler logic of the auction mechanism, the smart auction contract is designed in layers and deployed in the three layers of the data auction model respectively. The test results show that the smart contract has low computing cost and good practicability. Finally, massive simulation experiments prove the economic feasibility, incentive compatibility, real-time performance, and scalability of the data sharing incentive mechanism.

1 Literature Review

1.1 Centralized IoT data sharing incentive

With the exponential growth in IoT data, the research on IoT data sharing has attracted broad attention from the academic community[26–33]. For example, Wang *et al.*[26] proposed a vehicle-based recruitment strategy for ad hoc data sharing participants, which is applicable to vehicle trajectory prediction, minimizing the total recruitment cost. Ni *et al.*[27] proposed a fog-based

vehicular crowd sensing framework to address the security and privacy problems between the task requester and the worker. Xiao *et al*.[28] studied the problem of data sharing of the Internet of vehicles based on game theory and used the Q-Learning algorithm to implement a vehicle payment strategy. However, there is still a lack of effective incentive mechanisms in these studies. In most schemes, the data sharing behavior of IoT players is voluntary and proactive and does not conform with the objective reality. Concerning the incentive problem of data sharing among IoT devices, various single-layer auction mechanisms have been proposed. For example, Jin *et al*.[31] proposed an incentive-compatible auction mechanism to determine resource quotation in line with the demands of mobile devices and implemented resource-sharing transactions between mobile devices (buyers) and cloud service providers (sellers). Wen *et al*.[32] proposed a quality-driven auction-based incentive mechanism, which can compute participants' payment in accordance with the quality of sensing data to increase users' motivation to participate in acquiring and sharing sensing data.

In large-scale IoT data sharing incentive scenarios, the application of a single-layer auction mechanism is limited[30]. In a wireless network, the geographic locations of smart IoT devices are decentralized, and their data sharing service covers a limited range. Due to the limitations of communication and services, some terminal devices cannot access the data market. In this case, other IoT devices have to serve as intermediary agents to assist these terminal devices in accessing shared data resources. The single-layer auction model is not suitable for such hierarchical structure scenarios, nor can it solve the maximal social welfare. In this context, the hierarchical auction mechanism has been proposed and considered as a promising solution to the maximization of social welfare among IoT sharing devices. For example, Kiani and Ansari[29] studied and proposed a three-layer resource allocation model based on dynamic programming problems. Wang *et al*.[30] proposed a hierarchical auction-based mechanism for multi-robot real-time communication and efficient data retrieval. However, under conventional incentive mechanisms, most data sharing models are centralized and usually rely on a trusted third-party centralized mechanism, thus being vulnerable to attacks and a Single Point Of Failure (SPOF). In addition, unscrupulous users may provide false or even malicious data out of self-interest, further intensifying the trust crisis of data sharing[34].

1.2 Application of blockchain-empowered IoT data sharing and incentive mechanism

The blockchain-based distributed system is an effective technology for establishing secure and trusted data sharing[2, 35–38]. For example, Li *et al*.[36] implemented a blockchain-based mobile crowdsensing system that enables task requesters to send tasks directly to workers, avoiding the involvement of traditional centralized trusted third-party platforms. Cai *et al*.[16] developed a blockchain-assisted trust access authentication system for the Solid (Social linked data) project using the threshold RSA signature technology. This system, focusing on data sharing and privacy security, is also applicable to data sharing applications in IoT. However, most of the existing studies only apply blockchain to IoT to build a secure data sharing system, ignoring the performance bottleneck of the blockchain itself. In this paper, we will focus on the performance bottleneck of blockchains. In terms of the incentive mechanism, the blockchain technology has collaborated with various single-layer resource allocation protocols. For example, He *et al*.[37] proposed a truthful incentive mechanism that can meet the diverse resource allocation requirements of IoT users in dynamic and distributed P2P (Peer-to-Peer) environments. Kang *et al*.[38] proposed a localized P2P electricity trading model for locally buying and selling electricity among Plug-in Hybrid Electric Vehicles (PHEVs). Yao *et al*.[2] established a decentralized autonomous trading platform for Industrial IoT (IIoT) networks and modeled the interaction between the cloud provider and miners as a Stackelberg game. However,

few of these existing studies explored the blockchain-empowered multi-layer auction incentive mechanism. Our paper studies the hierarchical auction model and solves the security, efficiency, and incentive problems of multi-layer data sharing based on a blockchain framework, which will set our paper apart from existing research.

2 Efficient Blockchain-Empowered IoT Data Sharing Incentive Framework (ShareBC)

In its nature, ShareBC integrates the incentive mechanism and blockchain technology. To improve the consensus efficiency of a blockchain system, ShareBC proposes to enable some nodes to work in parallel through the sharding technology based on consortium blockchains to replace conventional network-wide consensus, thus avoiding the high computational cost caused by the Proof-of-Work (PoW) based consensus mechanism in public unlicensed blockchains[22, 23]. In this section, we first describe the constituent entities of ShareBC. Then, we introduce the process and key steps of data sharing implemented by ShareBC.

2.1 Framework description

As indicated by Figure 1, ShareBC includes three types of IoT sharing players: data providers, data agents, and data users.

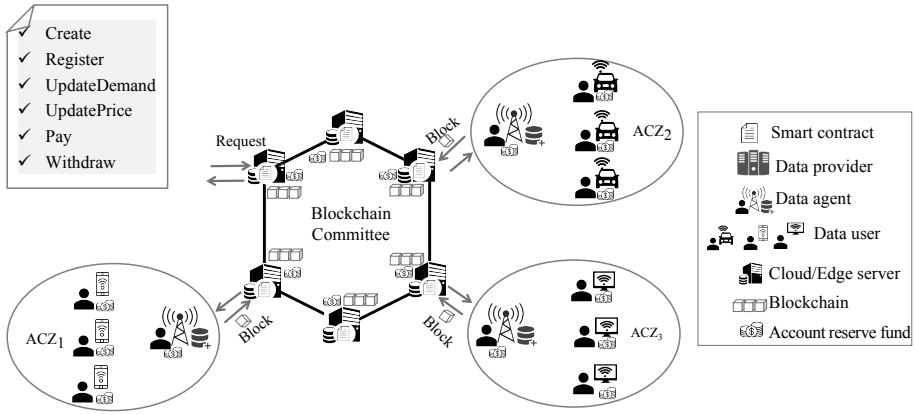


Figure 1 Efficient blockchain-based data sharing incentive framework for IoT (ShareBC)

The data provider refers to an IoT device with shared data resources. Both the data agent and the data user are the IoT devices with a demand for data. In a hypothetical scenario, each IoT data user connects directly to a nearby data agent and communicates with a cloud/edge server through a data agent. Each IoT data user can only connect to a unique data agent, so all data users connecting to one data agent will be assigned to the same zone, which is called ACZ. In addition, each data agent and each data user are configured with a separate data sharing transaction account, the address of which is required to be set as the information independent of the user's privacy, such as a public key. In the ShareBC mechanism, the data provider requests data sharing transactions through a cloud/edge server. Then, after the data provider's registration information is received by the auction platform consisting of cloud/edge servers, the amount and the payment of sharing data for the data agent and the data user respectively are determined by the hierarchical data auction mechanism in line with purchase demand, auction prices, and transmission data cost. Then, the data agent and the data user access the data resources obtained by their respective auctions and complete the corresponding payment. Finally, these data-sharing transactions are packaged into blocks in sharding ACZs, and the final audit is completed

by the pre-selected consortium blockchain committee and then added to the blockchain after a consensus has been reached.

2.2 Key steps of data sharing implemented by ShareBC

2.2.1 *System initialization*

System initialization includes two steps, i.e., the registration of IoT devices and the deployment of smart contracts. First, after the system is established, the Certificate Authority (CA) initializes the system parameters and generates the public and private keys for the newly registered IoT devices using asymmetric cryptography. For security concerns, the private key, once sent to the user, will be destroyed immediately, while the public key exists as the unique identity of the IoT device in the system. Considering the regulatory requirements for anonymous transactions in the consortium chain, CA implements a supervised anonymous authentication scheme[39] by storing the association table of the mapping relationship between the truthful identity information and the public key of the IoT device. As such, when there is a dispute over the identity of the IoT data user, its data agent can request the CA to arbitrate the dispute and track the truthful identity. Second, the compilation and deployment of the smart contract are completed. ShareBC automatically executes data-sharing transactions through the smart contract. After the smart contract is initialized in the blockchain network, the data provider can participate in customizing a data sharing incentive mechanism. Once deployed, the smart contract will have an independent ID and be permanently logged in the blockchain.

2.2.2 *Construction of sharding*

ShareBC introduced sharding[23] to divide the IoT devices in the network into several sub-networks to process the transaction verification, which needs to be performed by all the network nodes, in parallel in each sharding network area to enhance the transaction processing capability of the blockchain system. The specific steps for constructing the sharding are detailed as follows:

(1) Network sharding: According to a key property value of the IoT device (e.g. geographic coordinates), IoT devices are divided into different ACZs, each of which is homogeneous and has the same function and equal status. In light of fault tolerance, there is a threshold for the number of nodes in each ACZ.

(2) Work division of nodes: Each ACZ authenticates data transactions through the IoT devices within it. In sharding ACZs, the work of IoT device nodes will be divided. One of the nodes will be selected as a leader while some as followers. For the security of sharding, the next round of leaders will be re-selected after one round of work division is executed. In light of the property of consortium blockchains, the leader of each ACZ in the first round can be designated in advance. Afterward, the leader to be selected in each round can be determined randomly through a random-number-based computation[22].

(3) Increase or decrease in shard number: Given the entry of new IoT devices and the motion of original nodes, ShareBC requires that the threshold of the number of nodes in each ACZ is fixed. The number of nodes in each ACZ is proportional to the weight of the ACZ, thereby ensuring that ACZs can still keep balanced with each other after the increase or decrease in the number of sharding. For example, when the workload of the current ACZ is high, the system's throughput can be improved by adding shards. If the number of nodes in the current ACZ is lower than the security threshold value, the ACZ is canceled and the nodes within the zone are transferred to other ACZs.

2.2.3 *Role setting of IoT devices*

In the simulated data sharing incentive scenario shown in Figure 1, the data provider can broadcast its data transaction requests using the blockchain network and get paid by sharing

the data. There are mainly two types of data demanders, i.e. the data agent and the data user. Considering that the node communication within the sharding ACZ is affected by the geographical location and the coverage of the data sharing service is limited, ShareBC requires that the data user cannot directly request data sharing transactions from the data provider, and the data user has to rely on the data agent in the sharding ACZ to access the shared data; the data agent can directly trade data with the data provider to access shared data resources.

2.2.4 *Data sharing incentive mechanism implemented by smart contracts*

Figure 1 shows the six main Functional Interfaces (FI) of the smart contract. Critical events of IoT data sharing are automatically executed by calling these interfaces. The data provider first calls the Register interface of the smart contract to register the service of data sharing resources, and then the data agent calls the Register interface to join data sharing transactions. At the Register interface, the smart contract defines all the corresponding variables of the data sharing mechanism, such as the quantity of data resource sets D, initial data quotation p, data price increment C, and data demand r. After both the data provider and the data agent complete their registration, the smart contract creates the top-level market for data transactions between the two. Next, the data agent creates a sub-contract, $\mathcal{H}'$, through the Create interface of the smart contract and establishes the low-level market for data transactions between data users in the sharding ACZ where the data agent is located. Data users join the data sharing transactions through the Register interface of the sub-contract $\mathcal{H}'$.

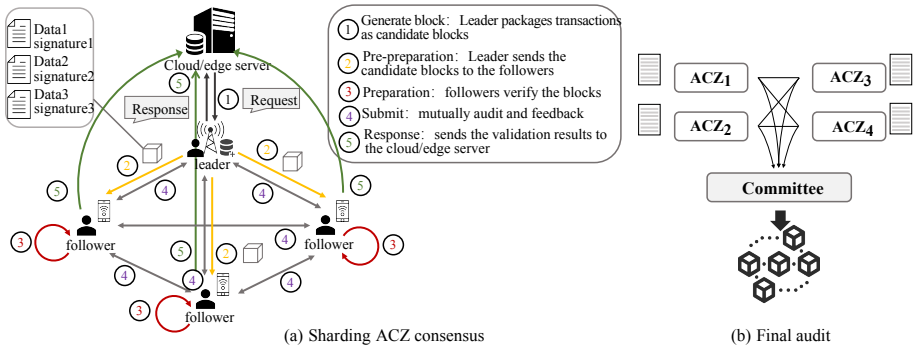
After the above steps are completed, the data provider, the data agent, and the data user can begin data sharing. To maximize social welfare and encourage IoT devices to actively participate in data sharing transactions, the smart contract provides an UpdateDemand interface, and the data users in sharding ACZ updates their data demand through the UpdateDemand interface of the contract $\mathcal{H}'$. When collecting enough low-level data demands, the data agent calls the UpdateDemand interface of the smart contract $\mathcal{H}$ to update its data demand in the top-level market. When the supply and the demand of data transactions are equal, the auction is completed. Before that, the smart contract provides the UpdatePrice interface for the data provider to update its data quotation, and then a new auction round begins. For each winner device (data agent and data user), payment is made to the data provider through the Pay interface. After the auction is completed, the data agent and the data user can withdraw their remaining account funds through the Withdraw interface.

2.2.5 *Access to shared data and transactions*

The winner data agent (data user) downloads the corresponding shared data resources from the data provider (data agent), completes the decryption, and accesses the shared data. To ensure the security of data resources during resale, the data provider can encrypt shared data using the One-Time Password (OTP)[40] technology. In this way, it can prevent the data agent from receiving repeated proceeds by reselling the data resources accessed through the auction of the data provider to the data user. After being broadcast by the smart contract, the data sharing transaction is completed. Each data sharing transaction consists of the transaction information and the digital signature[35]; the transaction information includes the payment log, the transaction cost, and the transaction-generated timestamps. Considering the limited storage capability of the blockchain system, the transaction data often contains an index to log the storage location of the encrypted shared data outside the chain; the digital signature is generated by the private keys of both parties of the transaction. Finally, after a certain number of transaction logs are collected by the nodes in the sharding ACZ, the transactions will be packaged into a block and enter the next process of consensus.

2.2.6 *Process of consensus*

A ShareBC-based blockchain consensus process mainly consists of two stages. First, transactions are authenticated within the sharding ACZ, and blocks are identified through consensus. Each sharding can select the consensus algorithm[20] within the zone (for example, PoW, PoS, PoB, or PBFT). Then, based on a certain pre-set agreement, a consensus will be reached between the ACZs to implement a global system in which ACZs are interconnected. Figure 2 shows the blockchain consensus process of the IoT data sharing framework; the two stages of the consensus process adopt the Practical Byzantine Fault Tolerance (PBFT) algorithm. In Figure 2(a), for example, the consensus process for nodes within each sharding ACZ is divided into five steps: block generation, pre-preparation, preparation, confirmation, and response.



(a) Sharding ACZ consensus (b) Final audit

Figure 2 ShareBC based blockchain consensus process

- Block generation: The completed data sharing transactions will be authenticated by all nodes in the network. The transactions, once authenticated to be true, will be signed by these authentication nodes and submitted to the leader node in their respective sharding ACZ. In each sharding ACZ, the leader node is responsible for packaging the collected and confirmed transactions as candidate blocks.
- Pre-preparation: Each leader node manages a unique list that logs the information of all the follower nodes inside the sharding ACZ in the current round. Based on this list, the leader node in the current round will forward the candidate blocks to the follower nodes in the sharding ACZ where the leader node is located for consensus.
- Preparation: Each follower node that receives the message will authenticate the validity of the candidate blocks.
- Confirmation: Each follower node completes the authentication of the candidate blocks and broadcasts the feedback with its own signature to the other nodes in the sharding ACZ. If more than a certain number of follower nodes, such as two-thirds of the total, reach consensus, the process will enter the submission phase and the request for submission is broadcast. Otherwise, the leader node will, based on the feedback results, consider whether to initiate the next round of consensus.
- Response: The leader node in the sharding ACZ submits the consensus-reached candidate blocks to the committee for final audit.

Upon the completion of the above steps, the leader node in the consensus-reached sharding ACZ will submit the candidate blocks to the consortium blockchain committee for final audit via a nearby cloud/edge server, as shown in Figure 2(b). In ShareBC, the committee consists of a set of cloud/edge servers provided by the consortium blockchain players. The final audit of the candidate blocks will be completed between the committee nodes by running the PBFT consensus protocol, and the blocks approved by the final audit will be added as new ones to the

blockchain and then broadcast synchronously to the other sharding ACZs in the network. In addition, the Committee is also responsible for generating a computational random number in each round to select the leader node in the sharding ACZ. In such a consensus mechanism, the Committee relies on distributed sharding ACZs and maintains complete control over data sharing transactions. This mechanism is highly open and can be audited, with advantages in computation cost and scalability. The consensus within the sharding will be detailed in Section 4.1.

3 Data Sharing Incentive Mechanism

How to design an effective incentive mechanism to drive IoT users to actively participate in data sharing is another focus of this paper. As a core part of ShareBC, this paper proposes a data sharing incentive mechanism based on the hierarchical data auction model implemented by the smart contract. This mechanism is able to maximize the participants' social welfare and ensure the efficiency of data sharing transactions. In this section, the data sharing problem in IoT is abstracted. Then, the formal representation is given. Afterward, the paper defines the research problem and puts forward the mathematical model of the hierarchical data auction. On this basis, a three-layer data auction algorithm based on the smart contract is proposed. Finally, relevant theorems about the algorithm are proved.

3.1 Problem description

The problem of IoT data sharing based on blockchain can be abstracted into a hierarchical data transaction market, as shown in Figure 3. The transaction market mainly consists of the data provider $\mathcal{P}$, the data agents $\mathcal{M} = \{1, 2, \cdots, M\}$, and the data user $\mathcal{N} = \{1, 2, \cdots, N\}$. $\mathcal{P}$ and $\mathcal{M}$ form the top-level market of data sharing transactions, while $\mathcal{M}$ and $\mathcal{N}$ constitute the low-level market. It is assumed that the shared data resources are divisible and homogeneous, with $\mathcal{D} = \{1, 2, \cdots, D\}$ being the shared data resource possessed by $\mathcal{P}$, where D denotes an integer. The utility vector of each data agent $j \in \mathcal{M}$ to the data set $\mathcal{D}$ is defined as $\boldsymbol{u}_j$, which is the same as the proceeds obtained by the data agent in its sharding ACZ through data auction. $\mathcal{N}_j$ is the set of data users in the sharding ACZ of the data agent j, $j \in \mathcal{N}_j$. This is because the data agent j can also participate in data transactions as a data user in the low-level market. The utility vector of each data user $i \in \mathcal{N}_j$ to the data set $\mathcal{D}$ is defined as $\boldsymbol{v}_i$, and the elements in $\boldsymbol{v}_i$ are ranked in a descending way according to the principle of diminishing marginal utility. Considering that data resources can be divided, it is assumed that the size of the k-th data resource accessed by the data user i is $D_i[k]$.

The system requires that after a data sharing transaction takes effect, the user which wants to access data has to rely on a data agent to connect to the data provider to download resources. The IoT data user cannot access the shared data from the data provider unless the user is assisted by the data agent in the sharding ACZ. The network channel capacity on both sides of communication is represented by $H_{i,j}$, and the transmission time required for the data user (or the data agent) $i \in \mathcal{N} \cup \mathcal{M}$ to access the k-th data resource from the data agent (or the data provider) $j \in \mathcal{M} \cup \mathcal{P}$ is

$$T_{i,j}(D_i[k]) = D_i[k]/H_{i,j} \tag{1}$$

According to the computation method of Hong *et al.*[9], transmission power consumption is defined as the product of transmission power and time of the data user or the data agent. The transmission power between the two parties is $P_{i,j}$, and the transmission power consumed by the data user (or the data agent) $i \in \mathcal{N} \cup \mathcal{M}$ to access the k-th data resource via the data agent (or the data provider) $j \in \mathcal{M} \cup \mathcal{P}$ is

$$E_{i,j}(D_i[k]) = P_{i,j} T_{i,j}(D_i[k]) \tag{2}$$

According to Formulas (1) and (2), the transmission cost required for the data user (or the data agent) $i \in \mathcal{N} \cup \mathcal{M}$ to access the k-th data resource from the data agent (or the data provider) $j \in \mathcal{M} \cup \mathcal{P}$ is represented by

$$C_{i,j}(D_i[k]) = f^E E_{i,j}(D_i[k]) + f^T T_{i,j}(D_i[k]) \tag{3}$$

where f^E and f^T are two cost factors, $f^E > 0$ and $f^T > 0$.

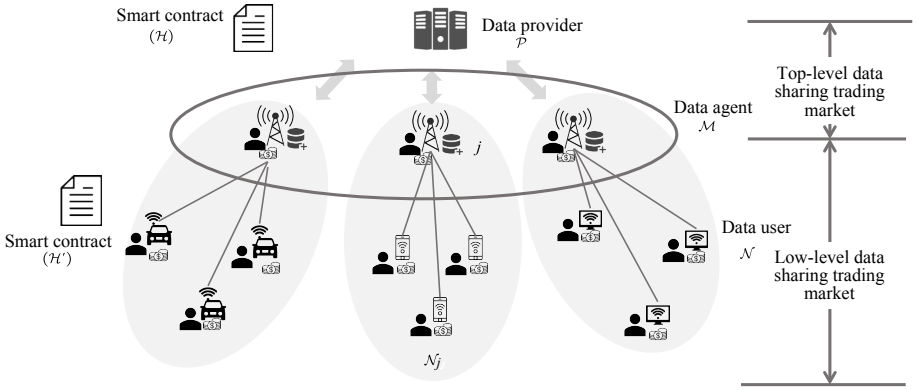


Figure 3 Hierarchical data sharing trading market based on blockchain

3.2 A formal definition of problems

It is assumed that the network topology of the three-layer data transaction market is fixed in the process of auction. In other words, it is required that the data provider, the data agent, and the data user cannot alter their current data transaction market during the auction. The objectives of data sharing participants conflict with each other: The data provider seeks to maximize their proceeds from sharing data; the data agent expects to maximize their proceeds from data resale; the data user hopes to minimize the cost of accessing data. In this case, the auction model should solve, to the greatest extent, the social welfare problem of all participants in data sharing to achieve an effective market equilibrium. The objective function is to maximize the difference between the data user's utility of shared data resources and the transmission cost for accessing data resources, which is formally defined as

$$\begin{cases} SW(\boldsymbol{q}) = \max\limits_{\boldsymbol{q}} \left\{ \sum\limits_{i \in \mathcal{N}_j, j \in \mathcal{M}} \sum\limits_{k=1}^{\boldsymbol{q}_i} (\boldsymbol{v}_i[k] - C_{i,j}(D_i[k]) - C_{j,\mathcal{P}}(D_i[k])) \right\} \\ \text{s.t.} \sum\limits_{i \in \mathcal{N}} \boldsymbol{q}_i = \mathcal{D} \end{cases} \tag{4}$$

where $\boldsymbol{q}$ is the allocation vector of data resources in the three-layer data transaction market; $\boldsymbol{v}_i[k]$ is the k-th element in the utility vector $\boldsymbol{v}_i$ of the data user $i \in \mathcal{N}$, namely the utility of i to the k-th data resource; $C_{i,j}$ is the transmission cost to the data user i accessing data via the data agent j; $C_{j,\mathcal{P}}$ is the transmission cost to the data agent j accessing the data resources of the data provider $\mathcal{P}$; and $D_i[k]$ is the size of the k-th data resource accessed by the data user i. The optimal data resource allocation vector $\boldsymbol{q}$ can be obtained by maximizing social welfare.

For the above objectives, it is necessary to provide the utility and cost of data-sharing individual participants. However, given the limitations on communication and services in the hierarchical structure of the data sharing market, the auction information between the top-level

market formed by $\mathcal{P}$ and $\mathcal{M}$ and the low-level market formed by $\mathcal{M}$ and $\mathcal{N}$ is incomplete. In the top-level market, the data provider cannot directly access the data user's data request in the low-level market located in the zones not covered by its data sharing services. Similarly, in the low-level market, the amount of data that the data agent can supply to the data user is not clear at the beginning of the auction, because at that time the data agent has not yet accessed the data quotation from the data provider. Therefore, this paper proposes a hierarchical data auction mechanism to solve the problem of maximizing the social welfare when the auction information in the multi-layer market is incomplete.

3.3 Hierarchical data auction mechanism

In this section, the SW problem is first transformed into the optimal data allocation problem in the hierarchical data transaction market. Next, the mathematical expression of the hierarchical data auction mechanism is given. Finally, the theorems and proof are given.

Definition 1 (Optimal data allocation problem in the top-level market)**.** In the top-level market, the data provider shares the data with the data agent. The optimal data allocation problem is to maximize the difference between the data agent's utility of sharing data resources and its transmission cost of accessing data. The problem is formally defined as

$$\begin{cases} \max\limits_{\boldsymbol{q}^m} \sum\limits_{j\in\mathcal{M}} \sum\limits_{k=1}^{\boldsymbol{q}_j^m} (\boldsymbol{u}_j[k] - C_{j,\mathcal{P}}(D_j[k])) \\ \text{s.t.} \sum\limits_{j\in\mathcal{M}} \boldsymbol{q}_j^m = \mathcal{D} \end{cases} \tag{5}$$

where $\boldsymbol{q}^m$ is the data allocation vector of all data agents in the top-level market; $\boldsymbol{q}_j^m$ is the amount of data allocated to the data agent j by the data provider; $\boldsymbol{u}_j[k]$ is the utility of the data agent j for the k-th data resource, $\boldsymbol{u}_j[k] \in \boldsymbol{u}_j$; $D_j[k]$ is the size of the k-th data resource accessed by the data agent j; and $C_{j,\mathcal{D}}$ is the transmission cost to the data agent j accessing the data resource of the data provider.

Definition 2 (Optimal data allocation problem in the low-level market)**.** The vector $\boldsymbol{q}^{m*}$ denotes the optimal data allocation solution for the top-level market. In the low-level market, the data agent j forwards the data resource $\boldsymbol{q}^{m*}$ accessed from the top market to the data users within its sharding ACZ. The optimal data allocation problem is to maximize the difference between the data user's utility for data resources and the transmission cost to the data user accessing data resources. The optimal data allocation problem is formally defined as

$$\begin{cases} \max\limits_{\boldsymbol{q}_j^e} \sum\limits_{i\in\{\mathcal{N}_j\}} \sum\limits_{k=1}^{\boldsymbol{q}_j^e[i]} (\boldsymbol{v}_i[k] - C_{i,j}(D_i[k])) \\ \text{s.t.} \sum\limits_{i\in\{\mathcal{N}_j\}} \boldsymbol{q}_j^e[i] = \boldsymbol{q}_j^{m*}, \sum\limits_{j\in\mathcal{M}} \boldsymbol{q}_j^{m*} = \mathcal{D} \end{cases} \tag{6}$$

where $\boldsymbol{q}_j^e$ is the data allocation vector for all data users in the sharding ACZ where the data agent j is located; $\boldsymbol{q}_j^e[i]$ is the amount of data allocated to the data agent i by the data agent j; $\boldsymbol{v}_i[k]$ is the data user i's utility for the k-th data resource, $\boldsymbol{v}_i[k] \in \boldsymbol{v}_i$; $D_i[k]$ is the size of the k-th data resource accessed by the data user i; and $C_{i,j}$ is the transmission cost to the data user i accessing the data resource via the data agent j.

The process of data allocation is described below: First, the data provider provides $\boldsymbol{q}_j$ units of data resources to the data agent j. Then, the data agent j resells $\boldsymbol{q}_j$ to the data users $\mathcal{N}_j$ in the sharding ACZ where the data agent j is located. If the data demand of $\mathcal{N}_j$ in the low-level market

of the sharding ACZ equates to the data supply $\boldsymbol{q}_j$ of the data agent j, then the optimal data allocation vector $\boldsymbol{q}_j^{e*}$ of the data users $\mathcal{N}_j$ can be obtained. In other words, when data supply and demand are equal between the top-level and the low-level markets, the vectors of solutions to optimal data allocation problems (5) and (6), $\boldsymbol{q}^{m*}$ and $\boldsymbol{q}^{e*}$, can be obtained, which is equivalent to the maximization problem (4) of SW. However, given the limitations on the hierarchical structure of the transaction market, the utility vector $\boldsymbol{u}_j$ of the data agent $j \in \mathcal{M}$ is unknown at the beginning of the auction. Accordingly, $\boldsymbol{q}^{m*}$ and $\boldsymbol{q}^{e*}$ cannot be directly computed. In this case, the hierarchical data auction mechanism has to obtain complete information to solve the maximization problem of SW.

This paper introduces a synchronization mechanism into the hierarchical data auction to solve the problem of maximizing the social welfare of the data provider forwarding the data sharing set $\mathcal{D}$ via M data agents to N data users. Specifically, on the one hand, in the top-level market, the Ascending Clock auction with Clinching (ACC) mechanism put forward by Ausubel *et al*.[39] is used to solve the optimal data allocation Problem (5). The data provider will publish the data quotation p_0 to the data agent at the beginning of the auction. Upon receipt of the quotation, the data agent, taking into account the utility at the quotation, provides the corresponding data demands to the data provider. The data provider then raises the data quotation (namely $p_0 + C$) in an increasing constant ratio C and starts the next round of auction. The auction process continues to iterate until the data demands of the data agent are equal to the shared data resource set $\mathcal{D}$ of the data provider. On the other hand, a scalable ACC mechanism is adopted in the low-level sub-market to solve the optimal data allocation Problem (6). The data resources that each data agent can provide to data users within the ACZ where the data agent is located are accessed through auction in the top-level market by the data agent. As the utility of the data agent is unpredictable, the data agent is required to broadcast the auction information of the top-level market to the sharding ACZ where it is located. Finally, to ensure the synchronization of hierarchical auctions, the data provider needs to formulate rules on the allocation and pricing of data transactions.

- Allocation rule: The data resources that the data agent auctions and accesses in the top-level market have to be re-auctioned immediately in the sharding ACZ where the data agent is located.
- Pricing rule: The auction price of data resources in the sharding ACZ shall not be higher than its final auction price in the top-level market.

Next, this paper will give a mathematical description of the hierarchical data auction mechanism. The data provider's quotation set for its shared data resource set $\mathcal{D}$ is $\boldsymbol{p} = \{p_0, p_1, \cdots, p_l\}$, where p_0 denotes the initial auction price and p_l the final auction price. According to Theorem 1, the quotation sets of the data sharing set $\mathcal{D}$ in the top-level market and its low-level sub-market are the same, namely $\boldsymbol{p} = \{p_0, p_1, \cdots, p_l\}$.

Theorem 1. In a hierarchical data auction mechanism, the data auctions in the top-level market and the low-level sub-market are terminated simultaneously.

Proof: The data quotation set in the top-level market is $\boldsymbol{p}_t = \{p_0, p_1, \cdots, p_l\}$, and the auction termination quotation is p_l; the data quotation set in the low-level sub-market is $\boldsymbol{p}_s = \{p_0, p_1, \cdots, p_l'\}$, and the auction termination quotation is p_l', $p_l \neq p_l'$. At this point, $p_l < p_l'$ implies that the auction in the top-level market is terminated, but that in the low-level market continues. Then, when the termination quotation is p_l', there is a case where a winner data user accesses the data through the auction while the other users give up the auction. In other words, when the termination quotation is p_l', it is inevitable that the data user changes its demand for the shared data resources. However, in the hierarchical data auction mechanism, the data agent must first collect the demands of all data users for data sharing in its low-level sub-market

before determining its own utility and submitting the corresponding data demands to the data provider. In this context, there will be data agents that have changed their data demands in the top-level market when the quotation is p_l'. Then it proves that the data auction in the top-level market is not terminated when the quotation is p_l', which contradicts the assumption that the auction termination quotation in the top-level market is p_l. In the same vein, if $p_l > p_l'$, the data users in the low-level sub-market will not be able to change their data demand; accordingly, data agents will not change their data demands. Therefore, $p_l > p_l'$ does not hold. To sum up, data auctions in the top-level market and the low-level sub-market will be terminated simultaneously.

When the data quotation is $p_t \in \boldsymbol{p}$, the data demand of the data agent j is expressed as

$$\boldsymbol{r}_j^m(p_t) = \sum_{i \in \mathcal{N}_j} \boldsymbol{r}_j^e(p_t)[i] - C_{j,\mathcal{P}} \tag{7}$$

where $C_{j,\mathcal{P}}$ is the transmission cost of the data agent j accessing data resources of data providers; $\boldsymbol{r}_j^e(p_t)$ is the data demand vector of the data user $\mathcal{N}_j$ for data quoted at p_t in the sharding ACZ where the data agent j is located; and $\boldsymbol{r}_j^e(p_t)[i]$ is the demand of the data user $i \in \mathcal{N}_j$ for data quoted at p_t in the sharding ACZ.

When the data quotation is $p_t \in \boldsymbol{p}$, the data demand of the data user i is expressed as

$$\boldsymbol{r}_j^e(p_t)[i] = \sum_{k \in \boldsymbol{v}_i} I(k - C_{i,j}, p_t) \tag{8}$$

where $C_{i,j}$ is the transmission cost of the data user i accessing data via the data agent j; and $I(x, y)$ is the indicator function. When $x \leq y$, $I(x, y) = 0$; when $x > y$, $I(x, y) = 1$. As such, the data user i's demand for data quoted at p_t, namely $\boldsymbol{r}_j^e(p_t)[i]$, decreases as the quotation p_t increases. For any $k \in \boldsymbol{v}_i$, if $y' > y$ and $\{y, y'\} \in \boldsymbol{p}$, then $I(k, y') \leq I(k, y)$.

It is assumed that when the quotation is $p \in \boldsymbol{p}$, the data agent j resells the k-th data resource that it accesses in the top-level market through auction to the data user i. Then, the utility of the k-th data resource is expressed as $\boldsymbol{u}_j[k] = p - C_{i,j}(D_j[k])$, where $\boldsymbol{u}_j[k]$ is the k-th element in the utility vector of the data agent j; and $C_{i,j}$ is the transmission cost for accessing data resources. According to the allocation rule, when the data quotation is $p_t \in \boldsymbol{p}$, the sums of the data accessed through auction by the data agent j and the data user $i \in \mathcal{N}_j$ are respectively

$$\boldsymbol{q}_j^m(p_t) = \max\left\{0, \mathcal{D} - \sum_{k \in \mathcal{M} \setminus j} \boldsymbol{r}_k^m(p_t)\right\} \tag{9}$$

$$\boldsymbol{q}_i^e(p_t) = \max\left\{0, \boldsymbol{q}_j^m(p_t) - \sum_{k \in \mathcal{N}_j \setminus i} \boldsymbol{r}_j^e(p_t)[k]\right\} \tag{10}$$

When $p_t = p_l$, the auction is terminated, and then the solution vectors $\boldsymbol{q}^{m*}$ and $\boldsymbol{q}^{e*}$ of the optimal data allocation Problems (5) and (6) can be obtained. Furthermore, Formulas (9) and (10) mentioned above can be further expressed as

$$Q_j^m(p_t) = \begin{cases} \max\{0, \boldsymbol{q}_j^m(p_t)\}, t = 0 \\ \max\{0, \boldsymbol{q}_j^m(p_t) - \boldsymbol{q}_j^m(p_{t-1})\}, t \in \{1, 2, \cdots, l\} \end{cases} \tag{11}$$

$$Q_i^e(p_t) = \begin{cases} \max\{0, \boldsymbol{q}_i^e(p_t)\}, t = 0 \\ \max\{0, \boldsymbol{q}_i^e(p_t) - \boldsymbol{q}_i^e(p_{t-1})\}, t \in \{1, 2, \cdots, l\} \end{cases} \tag{12}$$

As long as the data agent or the data user can access the shared data resources quoted at $p_t \in \boldsymbol{p}$, transactions can be carried out. Moreover, the data agent or the data user needs to pay the price p_t for each unit of data resources. According to the payment rule, the prices that the data agent j and the data user $i \in \mathcal{N}_j$ have to pay are respectively

$$P_j^m(\boldsymbol{p}, \boldsymbol{Q}^m) = \sum_{t=0}^{t=l} \sum_{k=1}^{k=Q_j^m(p_t)} (p_t - C_{j,\mathcal{P}}(D_j[k])) \tag{13}$$

$$P_i^e(\boldsymbol{p}, \boldsymbol{Q}^e) = \sum_{t=0}^{t=l} \sum_{k=1}^{k=Q_i^e(p_t)} (p_t - C_{i,j}(D_i[k])) \tag{14}$$

where $C_{j,p}$ is the transmission cost to data agent j for accessing data resources shared by the data provider j; $C_{i,j}$ is the transmission cost to data user i for accessing data resources through the data agent j; $D_j[k]$ is the size of the k-th data resource of the data provider that the data agent j accesses through auction; and $D_i[k]$ is the size of the k-th data resource accessed by the data user i after the auction through the data agent j.

3.4 Three-layer data auction algorithm based on smart contracts

In the three-layer data auction mechanism, the data provider can gradually access the demand of the data user in the low-level market for the shared data resources during the auction process, and the data user can gradually access the shared data provided by the data agent in this process. Thus, upon the completion of the auction, the optimal solutions $\boldsymbol{q}^m$ and $\boldsymbol{q}^{e*}$ are obtained to maximize the social welfare. Algorithm 1 describes how the three-layer data auction based on smart contracts works. The concrete steps are as follows:

Step 1. The smart contract $\mathcal{H}$ for auction is compiled and deployed during the initial phase of the blockchain system. The data provider $\mathcal{P}$ calls the Register interface in the smart contract $\mathcal{H}$ for registration and initializes the top-level data market, providing relevant bidding data, such as the number D of the data sharing set $\mathcal{D}$, initial quotation p, and price increment C for each round of the auction iteration. Similarly, each data agent j with their account reserve funds is registered through the Register interface to join the top-level market and initializes its data demand $\boldsymbol{r}_j^m \leftarrow 0$.

Step 2. The data agent j calls the Create interface of the smart contract $\mathcal{H}$ to create the smart contract $\mathcal{H}_j$ of the low-level sub-market to which the data agent j is connected to. Then, the data users with their reserve funds in the ACZ can join the low-level sub-market by calling the Register interface of the smart contract $\mathcal{H}_j$ and initialize their data demand $\boldsymbol{r}_j^e \leftarrow [0, \cdots, 0]$.

Step 3. According to the initial quotation p, the data user in the low-level sub-market updates its data demand $\boldsymbol{r}_j^e[i]$ through the UpdateDemand interface of the smart contract $\mathcal{H}_j$. Taking into account the updated demands from the data users, the data agent updates its data demand $\boldsymbol{r}_j^m$ through the interface UpdateDemand of the smart contract $\mathcal{H}$.

Step 4. The data provider $\mathcal{P}$ shares a corresponding amount of data with the data agent in line with Formula (9), and the data agent allocates the corresponding data to the data user in line with Formula (10).

Step 5. According to Formulas (9) and (10), the data agent and the data user call the Pay interface of the smart contracts $\mathcal{H}$ and $\mathcal{H}_j$ respectively to complete the payment.

Step 6. In case of supply-demand imbalance in the top-level and low-level markets, it is necessary to perform the next round of auction. The data provider $\mathcal{P}$ calls the UpdatePrice interface of the smart contract $\mathcal{H}$ to update the quotation $p \leftarrow p + C$, and then starts the next round of iteration from Step 3 of Algorithm 1.

Step 7. After the whole data auction is completed, the data agent and the data user can call respectively the Withdraw interface of their corresponding smart contracts to withdraw the remaining funds in their own account.

Algorithm 1. Three-layer data auction algorithm based on the smart contract.

Input: $D, p \leftarrow p_0, C = 1, \boldsymbol{r}_j^m \leftarrow 0, \boldsymbol{r}_j^e \leftarrow [0, \cdots, 0]$;
Output: $\boldsymbol{q}^{m*}, \boldsymbol{q}^{e*}, \boldsymbol{p}^m, \boldsymbol{p}^e$.

1. In the top-level market, the data provider $\mathcal{P}$ registers data resource services through the **Register** interface of the smart contract $\mathcal{H}$;
2. Set the global variable of $\mathcal{H}$, including the amount of auction data resource set $\mathcal{D}$, initial data quotation $p \leftarrow p_0$, and price increment in each round $C = 1$
3. **for** $j \in \mathcal{M}$ **do**
4. The data agent j with account reserve funds participates in the auction in the top-level market through the **Register** interface of the smart contract $\mathcal{H}$;
5. The data agent j creates a smart sub-contract $\mathcal{H}_j$ through the **Create** interface of the smart contract $\mathcal{H}$ to manage its low-level sub-market transactions;
6. **for** $i \in \mathcal{N}_j$ **do**
7. The data user i with account reserve funds registers through the **Register** interface of the smart contract $\mathcal{H}_j$ and participates in the auction in the low-level market to which its data agent j is connected to;
8. **end for**
9. Set the global variable of $\mathcal{H}_j$, including the data demand $\boldsymbol{r}_j^m \leftarrow 0$ of the data agent j and the demand $\boldsymbol{r}_j^e \leftarrow [0, \cdots, 0]$ of the data user $i \in \mathcal{N}_j$ for shared data resources;
10. **end for**
11. **while** $D \neq \sum_{j \in \mathcal{M}} \boldsymbol{r}_j^m$ **do**
12. **for** $j \in \mathcal{M}$ **do**
13. **for** $i \in \mathcal{N}_j$ **do**
14. The data user i calls the **UpdateDemand** interface of the smart contract $\mathcal{H}_j$ to update its demand for data $\boldsymbol{r}_j^e[i]$;
15. **end for**
16. The data agent j calls the **UpdateDemand** interface of the smart contract $\mathcal{H}$ to update its data demand $\boldsymbol{r}_j^m$;
17. **end for**
18. The data provider $\mathcal{P}$ provides the corresponding amount of data in line with Formula (9) to the data agent, and the data agent j provides the corresponding amount of data in line with Formula (10) to the data users in the sharding ACZ where the data agent j is located;
19. According to Formulas (13) and (14), the data agent j and the data users $\mathcal{N}_j$ complete the payment for the data resources accessed through auctions via the **Pay** interface of smart contracts $\mathcal{H}$ and $\mathcal{H}_j$ respectively.
20. The data provider $\mathcal{P}$ calls the **UpdatePrice** interface of the smart contract $\mathcal{H}$ to update data quotation $p \leftarrow p + C$;
21. **end while**
22. After the auctions are completed, the data agent j and the data users $\mathcal{N}_j$ withdraw their remaining reserve funds in their own account by calling the **Withdraw** interface of smart contracts $\mathcal{H}$ and $\mathcal{H}_j$ respectively.

3.5 Main theorem about the algorithm

The hierarchical data auction algorithm proposed in this paper has high efficiency and practicability. To prove these attributes, this paper compares it with the classic ACC-based double auction algorithm[41]. Double auction[31] is a practical auction mode extensively used in actual transactions. Both parties of the transactions submit the asking price and the bidding price to the agent of the auction. Finally, the agent of the auction matches the tender prices of both parties and sets the corresponding rules for resource allocation and pricing. However, standard auction algorithms are often inefficient when bidders have multi-unit demands. The

double auction algorithm based on ACC adopts the ACC mechanism in which the auction price increases by competition between similar products, which can cope with the problem of efficiency and produce an efficient and practical auction model. Therefore, it is selected as the benchmark algorithm to prove the effectiveness of the algorithm proposed in this paper. Next, the proof process of the theorem is given.

Theorem 2. When the limitations on communication and sharing services of ShareBC and the transmission cost for accessing the shared data are not considered, the hierarchical data auction algorithm proposed in this paper is equivalent to the ACC-based double auction algorithm.

Proof: It is assumed that the data provider will conduct data sharing transactions of D units of data resources. First, the double auction mechanism based on ACC is adopted, namely that IoT data users directly conduct data transactions with data providers. $\boldsymbol{Q}_i$ is the data allocation vector of the data user $i \in \mathcal{N}$; data auction quotation is $\boldsymbol{p} = \{p_0, p_1, \cdots, p_f\}$; and $\boldsymbol{r}(p_d)[\boldsymbol{i}]$ is the demand of the data user i for shared data resources when the auction quotation is $p_d \in \boldsymbol{p}$. It is set that $p_d \in \boldsymbol{p}$ is the price of the first shared data resource of the data user i. Afterward, the auction quotation becomes $p_d \leftarrow p_d + C$ in each round, satisfying

$$\begin{cases} 1 = Q_i(p_d) = D - \sum_{k \in \mathcal{N} \setminus i} \boldsymbol{r}(p_d)[k] \\ \sum_{i \in \mathcal{N}} \sum_{k=p_0}^{p_d - C} Q_i(k) = 0,\ p_d > 0 \end{cases} \tag{15}$$

According to Formula (15),

$$D - \sum_{k \in \mathcal{N}} \boldsymbol{r}(p_d)[k] + Q_i(p_d) = 1 \tag{16}$$

Next, the hierarchical data auction mechanism proposed in this paper is adopted for analysis. There are two data agents in the three-layer data sharing transaction market, i.e., M_1 and M_2, and i data users are in the low-level sub-market to which M_1 is connected. The data allocation vectors of M_1 and M_2 are $\boldsymbol{Q}_1^m$ and $\boldsymbol{Q}_2^m$ respectively, and the data auction quotation is $\boldsymbol{p} = \{p_0, p_1, \cdots, p_f\}$. When the auction quotation is $p_t \in \boldsymbol{p}$, the amount of data that M_1 and M_2 can access is respectively

$$\begin{cases} Q_1^m(p_t) = \max\{0, D - \boldsymbol{r}_2^m(p_t)\} \\ Q_2^m(p_t) = \max\{0, D - \boldsymbol{r}_1^m(p_t)\} \end{cases} \tag{17}$$

where $\boldsymbol{r}_j^m$ is the demand of shared data resources of the data agent $j \in \{1, 2\}$; $\boldsymbol{Q}_i^e$ is the data allocation vector of the data users $i \in M1$; and $p_h \in \boldsymbol{p}$ is the price of the data accessed by the data user i in the first auction in the low-level sub-market. Afterward, the quotation becomes $p_h \leftarrow p_h + C$ in each round of the auction. Then

$$\begin{cases} 1 = Q_i^e(p_h) = Q_1^m(p_h) - \sum_{k \in \mathcal{N}_1 \setminus i} \boldsymbol{r}_1^e(p_h)[k] \\ \sum_{i \in \mathcal{N}_1} \sum_{k=p_0}^{p_h - C} Q_i^e(k) = 0,\ p_h > 0 \end{cases} \tag{18}$$

where $\boldsymbol{r}_1^{p_h}[i]$ is the demand of the data user i for shared data resources priced at $p_h \in \boldsymbol{p}$. According to Formula (18), we can get

$$Q_1^m(p_h) - \sum_{k \in \mathcal{N}_1} \boldsymbol{r}_1^e(p_h)[i] + Q_i^e(p_h) = 1 \tag{19}$$

According to Formulas (17) and (19), the calculations are as follows:

$$\begin{aligned} 1 &= D - \boldsymbol{r}_2^m(p_h) - \sum_{i\in\mathcal{N}_1} \boldsymbol{r}_1^e(p_h)[i] + Q_i^e(p_h) \\ &= D - \boldsymbol{r}_2^m(p_h) - \boldsymbol{r}_1^m(p_h) + Q_i^e(p_h) \\ &= D - \sum_{i\in\mathcal{N}} \boldsymbol{r}^e(p_h)[i] + Q_i^e(p_h) \end{aligned} \tag{20}$$

The comparison between Formulas (16) and (20) reveals $p_d = p_h$. Therefore, when ShareBC communication and service limitations and the transmission cost for accessing shared data resources are not considered, the hierarchical data auction proposed in this paper is equivalent to the ACC-based double auction.

4 Performance Evaluation

4.1 Comparison between sharding protocols and analysis

This section evaluates the ShareBC sharding protocol in terms of sharding formation, Inter-shard consensus, security, and scalability. Table 1 shows a comprehensive comparison with the current classic blockchain sharding protocols. Section 2.2 of this paper has expounded on the key steps for the sharding protocol in data sharing implemented by ShareBC, such as sharding formation and consensus process. Next, sharding settings and performance comparisons are described.

Table 1 Sharding settings and performance comparisons

Protocol	Node joining	Transaction model	Protocol consistency	Node allocation	Inter-shard consensus			Security (fault tolerant capability)	Scalability
					Node configuration	Leader	Communication complexity		
RSCoin[42]	License based	UTXO	✓	✗	Static state	Internal election	O(n)	1/3	2,000 tx/s
Chainspace[43]	Flexible	account/ balance	✓	✗	Flexibility	Internal election	O(n^2)	1/3	350 tx/s
Elastico[44]	PoW	UTXO	✓	Dynamic randomness	Periodic transformation (Full exchange)	Internal election	O(n^2)	1/3	16 blocks/110 s
OmniLedger[45]	PoW/PoS	UTXO	✓	Dynamic randomness	Periodic transformation (Replace subsets)	Internal election	O(n)	1/4	6,000 tx/s
RapidChain[46]	Off-line PoW	UTXO	✓	Dynamic randomness	Periodic transformation	Internal election	O(n)	1/3	7,300 tx/s
Monoxide[47]	PoW	Account/ balance	✓	Static state	Static state	✗	✗	1/2	11,694 tx/s
ShareBC	License based	Account/ balance	✓	Static state	Periodic transformation	Internal election	O(n)	1/3	✓

✓ denotes containing the property; ✗ indicates not containing the property.

Regarding the protocol setting, node joining implies that nodes are allowed to join the rules and standards on which the current epoch is based. For example, ID accessing based on the PoW or PoS mechanism is an important way for unpermissioned blockchain systems to prevent Sybil attacks. However, ShareBC is proposed based on permission blockchains and allows the blockchain system to operate in a relatively trusted environment, in which successfully registered IoT devices are allowed to participate in nodes. In addition, ShareBC adopts an account/balance transaction model. This is a user-friendly model applicable to the smart contract, and the transactions of any amount can be executed by one sending account and one receiving account

rather than multiple bilateral UTXOs. Such an equilibrium can extend to more complex states, thus supporting the logic of programmable applications. Finally, the classic Byzantine fault tolerance protocol adopted in the ShareBC sharding scheme has high consistency in negotiated consensus.

Node allocation refers to how participating nodes are allocated to corresponding shardings in the blockchain system. Most of the existing research is based on the random number generated by epoch, i.e., a Verifiable Random Function (VRF). In a few studies, Monoxide[47] protocol nodes are not randomly allocated but based on addresses. In the ShareBC sharding protocol, for each successfully registered IoT device, the system will allocate the device, in line with one key property value (such as its geographical coordinates), to the corresponding ACZ. In terms of consensus within the sharding, in general, its node configuration can be (permanently) static or dynamically and periodically changed, as happens with alternate replacement, full exchange, or replacement of subset nodes. Considering the mobility of IoT devices and the security of the ACZ, ShareBC will set the device nodes in the ACZ to change periodically. In addition, each ACZ will conduct a leader election within each epoch. The leader comes from the IoT device nodes within the shard. The communication complexity denotes the time complexity of the communication between the internal nodes of ACZ. Assuming that it denotes the number of nodes within an ACZ, the communication complexity within each ACZ in ShareBC is expressed as $\mathrm{O}(n)$.

Concerning security and scalability, the ShareBC competitor model is set on the basis of BFT. The number of malicious or erroneous nodes, which can be tolerated by its consensus protocol, is at most 1/3. The throughput values shown in Table 1 are correlated to the settings of experimental parameters[45]. Specifically, the experimental parameters are listed as follows: RSCoin includes 3 nodes/shard and 10 shards; Chain Space includes 4 nodes/shard and 15 shards; Elastico includes 100 nodes/shard and 16 shards; OmniLedger includes 72 nodes/shard (12.5% of the competitors) and 25 shards; RapidChain includes 250 nodes/shard and 4,000 nodes; Monoxide includes 2,048 shards and 48,000 nodes. Throughput values show that these sharding systems have scalability. The ShareBC sharding protocol requires two rounds of verification. The IoT device node (follower) first verifies the internal consistency of ACZ sharding through PBFT consensus and then submits it to the consortium blockchain committee to achieve a global consistency and adds the verified block to the chain. The proposed scheme reduces the transaction latency and improves the transaction throughput because the chained block does not need to wait for the confirmation time of six blocks as the Bitcoin network does. In addition, the consensus mechanism in this paper is inspired by RSCoin to some extent. The Committee in the ShareBC system relies on the distributed sharding ACZ while maintaining full control over data sharing transactions, ensuring highly open auditable secure transactions.

4.2 Prototype implementation

The smart contract enforces key events in the incentive mechanism in a non-repudiation and automated manner, improving the security and efficiency of data sharing. This experiment develops a prototype system for the performance test of the smart contract $\mathcal{H}$ and its sub-contract $\mathcal{H}_j$ and deploys it to the Ethereum test network to compute the Gas cost of the smart contract and each interface. In the Ethereum blockchain, the Gas price represents the Ether consumed to perform a task. The unit of measurement is Wei, and 1 Wei $= 10^{-18}$ Ether. Table 2 shows the average Gas cost for each interface in the smart contracts $\mathcal{H}$ and $\mathcal{H}_j$ (after 20 rounds of tests). In the table, the execution cost represents the Gas consumption of the smart contract executing the instruction; other costs indicate the amount of Gas consumed by the transaction that calls the interface. According to the test results in Table 2, the Create interface of the entire smart contract consumes the most Gas, but it is only called once when a data agent creates the

low-level sub-market to which it is connected to.

Table 2 Gas cost of interfaces in the smart contracts $\mathcal{H}$ and $\mathcal{H}_j$

Interface	Execution cost (Gas)	Other costs (Gas)	Transaction cost (Gas)	Transaction cost ($)
Create	1,120,572	382,327	1,502,899	0.71537992
Register ($\mathcal{H}$)	652,320	21,191	673,511	0.32059124
Register ($\mathcal{H}_j$)	60,978	21,191	82,169	0.03911244
UpdateDemand	20,671	21,442	42,113	0.02004579
UpdatePrice	5,565	21,442	27,007	0.01285533
Pay	27,260	21,191	48,451	0.02306268
Withdraw	12,884	5,799	18,683	0.00889311

It is assumed that there is one data provider, two data agents, and ten data users in a hierarchical data auction system. According to the current exchange rate of 1 Ether $\approx$ 238$ and 1 Gas $= 2 \times 10^{-11}$ Ether, the data provider in this system only needs 0.32059124$ to publish a data sharing transaction in the blockchain network. The total cost of data demanders (including one data agent and five data users) in each low-level sub-market after one round of data auction is 1.490184$. The average cost of each data demander is 0.248364$. It should be noted that one round of data auction computed here specifically includes one call to the Register ($\mathcal{H}$) interface, one call to the Create interface, five calls to the Register ($\mathcal{H}_j$) interface, six calls to the UpdateDemand interface, and six calls to the Pay interface. After repeated tests, the results show that the cost for executing smart contracts $\mathcal{H}$ and $\mathcal{H}_j$ is low, which proves that the hierarchical auction mechanism implemented by the smart contract, if applied to the IoT data sharing incentive framework, is economically feasible.

4.3 Simulation results and analysis

The performance of the hierarchical data auction algorithm proposed in this paper is tested by simulation. In the experiment, the scale of the data-sharing participants is $(x, y, (z_1, z_2, \cdots, z_y))$, where x is the number of data providers, y the number of data agents, and z_i the number of data users in the relevant sharding ACZ. The simulation parameters are set as follows: The scale of the participants in the three groups is #1:(1, 3, (5, 5, 5)), #2:(1, 3, (10, 10, 10)), and #3:(1, 3, (15, 15, 15)), respectively.

In the top-level market, the amount of shared data of the data provider is $D = 50$; in the low-level market, the transmission power of the data user $i \in \mathcal{N}_j$ is $P_i = 2$ W, and the communication bandwidth is $B = 10$ MHz; and the distance between the data user (or the data agent) $i \in \mathcal{N} \cup \mathcal{M}$ and the data agent (or the data provider) $j \in \mathcal{M} \cup \mathcal{P}$ is randomly set within a range of (0, 20] m. The value range of the cost factor f^E is [0, 1], and $f^T = 1 - f^E$. The size of the data resource accessed by the data user is 1, and the size of the element corresponding to the utility vector $\boldsymbol{v}_i$ of the data user $i \in \mathcal{N}_j$ is within [0, 100]. According to Hong *et al.*[9], the noise power and the channel power per unit distance $d_{i,j} = 1$ m are $\delta^2 = -120$ dBm and $\beta_0 = -50$ dB respectively. Then, the channel capacity between the data user $i \in \mathcal{N}_j$ and the data agent j is $H_{i,j} = B \log_2(1 + \lambda_i^0 / d_{i,j}^2)$, in which λ_i^0 denotes the receiving Signal-to-Noise Ratio (SNR) of the data user i when $d_{i,j} = 1$ m, and $\lambda_i^0 = \beta_0 P_i / \delta^2$. Finally, for accurate experimental results, data of each experiment are the means of 100 independent simulation results.

To test the effectiveness of the algorithm, experiments are carried out on the social welfare when the scale of the participants is #1, #2 and #3 respectively. Figure 4 shows the convergence of the social welfare function of the hierarchical data auction algorithm at different scales. From the figure, this algorithm can quickly obtain the maximum social welfare at different data sharing scales. As the scale of the participants $(x, y, (z_1, z_2, \cdots, z_y))$ expands, the value of the converged social welfare becomes increasingly large. This is because the resource

competitiveness of the three-layer data transaction market increases with the larger number of data sharing users. It also means that the data user has to pay higher prices to become the auction winner to access data.

Figure 5 describes the relationship between the total demand on data demand side (including data agents and data users) and the supply of data providers. The overall trend shows that the data supply by data providers grows with the higher data auction price, while the total demand of data agents and data users will decrease with the higher data auction price. Eventually, the two curves converge to the same value, namely the number of data sharing resources owned by the data provider $D = 50$. From Figures 4 and 5, social welfare is maximized when the total demand of data agents and data users for data sharing is equal to the supply by data providers. In addition, the changing trend of the social welfare curve shown in Figure 4 is the same as that of the supply curve of data providers shown in Figure 5, because social welfare will not change unless the number of data resources that data providers are willing to share varies. Experimental results demonstrate that the hierarchical data auction mechanism is effective and can maximize social welfare.

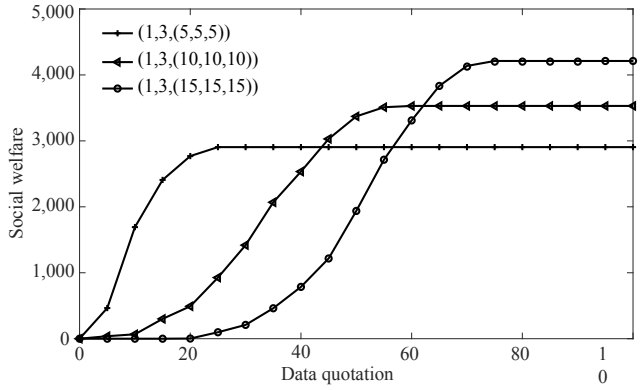


Figure 4 Convergence of social welfare functions

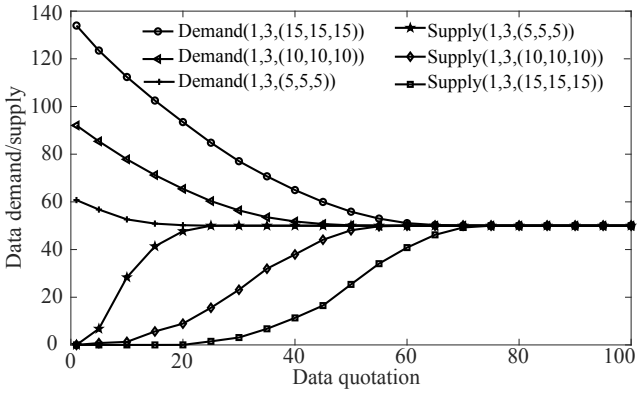


Figure 5 Relationship between demands and supplies

Figure 6 shows the impact of the transmission cost on social welfare in the algorithm when the scale of the participants is #2. A smaller cost factor f^E can lead to greater convergence of the social welfare function, namely that a higher transmission cost results in lower maximum

social welfare. Before the convergence, a larger cost factor f_E leads to a faster change in social welfare curves. Accordingly, the data users in the auction system will access data resources more quickly. This is because, in the hierarchical data auction mechanism, the transmission cost of the low-level sub-market is the cost to the data users within the sharding ACZ. On the precondition that the utility of data sharing is the same, a higher transmission cost indicates a lower price of an auction in which data users participate. Considering the same market competitiveness, a lower auction quotation leads to a lower transaction price, and the winner will access the data resources sooner.

Figure 7 illustrates the total utility of data providers, data agents, and data users when the scale of the participants is #1, #2, and #3 respectively after social welfare is maximized. The first column bar is the net utility of the data provider; the difference between the center and left bar is the net utility of the data agent; the difference between the right and the center bar is the net utility of the data user. The net utilities of data providers, data agents, and data users are all positive, indicating that the proposed hierarchical data auction mechanism satisfies the weak budget balance.

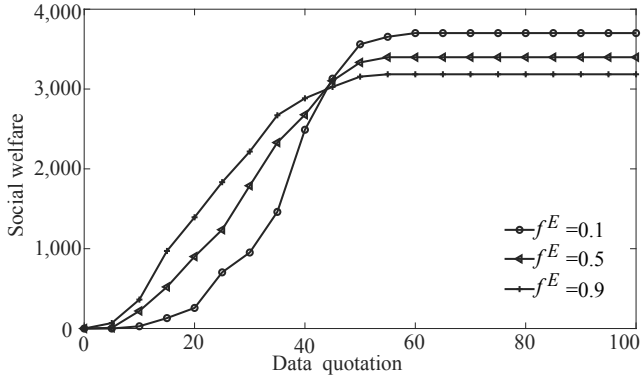


Figure 6 Social welfare under different cost factors

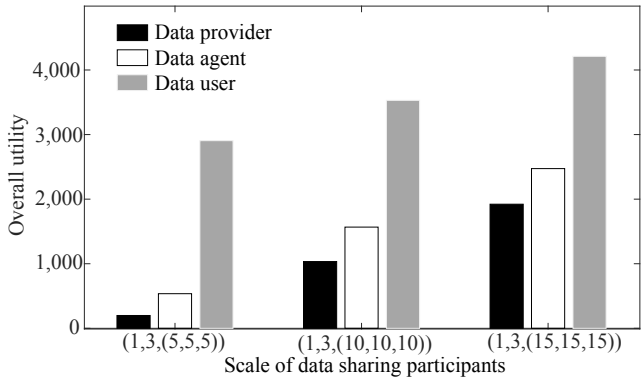


Figure 7 Overall utility of data sharing participants

In the hierarchical data auction mechanism, the data agent is required to immediately resell the data sharing resources to the data user in the low-level sub-market, once accessing data sharing resources through the auction in the top-level market. The experiments are carried out on the real-time performance of the hierarchical data auction mechanism, as shown in Figures 8 and 9.

Figure 8 shows the average reduced time versus the number of demanders (data agents and data users) in the three-layer data auction algorithm. It concludes that the reduction in latency will decrease as the number of data sharing users rises.

Figure 9 explains this trend. It shows the whole auction process of data agents and data users when the scale of participants is #1, #2, and #3 respectively. x_i ($i \in \{1, 2, 3\}$) denotes the process from data users winning the first shared data resource set through the auction to the completion of the auction while $x_1 > x_2 > x_3$ indicates that a smaller scale of participants relates to fewer data users. Accordingly, as the auction is completed sooner, data resources will be accessed more quickly. Hence, the scale of participants in data sharing transactions determines market competitiveness. When the market competitiveness is poor, it will take data users a shorter time to access shared data resources. Therefore, when the number of devices participating in data sharing in the IoT is small, the real-time performance of the algorithm will be better.

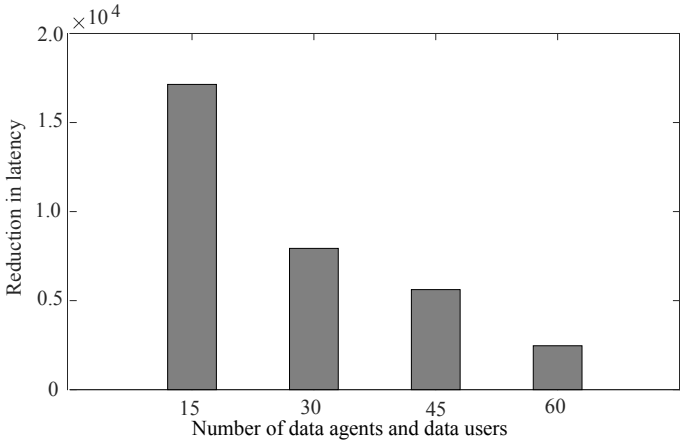


Figure 8 Reduction in latency with the number of demanders

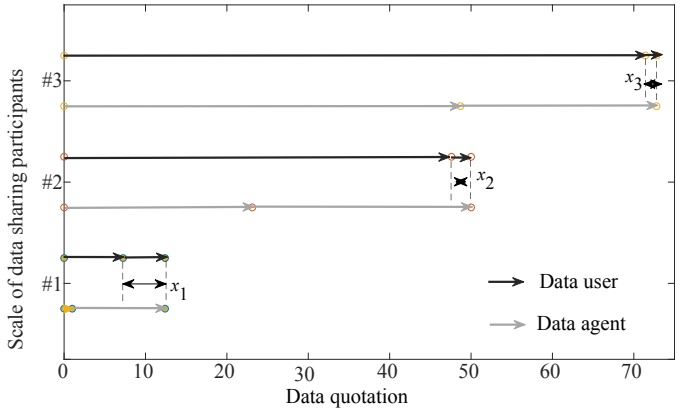


Figure 9 Auction process of players in different cases

Finally, given different numbers of data users, experiments are conducted to compare algorithms in terms of maximal social welfare. The test algorithms include the double auction algorithm based on ACC and the three-layer data auction algorithm based on the smart contract.

Experimental results show that the maximum social welfare implemented by the two auction algorithms is basically the same.

Figure 10 shows a small gap between the two; this may be because the values of the two algorithms in the experiments are all means. The experimental results verify the correctness of Theorem 2. When ShareBC communication and service limitations and the transmission cost for accessing shared data resources are not considered, the hierarchical auction proposed in this paper is equivalent to the ACC-based double auction.

Figure 11 illustrates the testing effect after the algorithm expands the node scale. With more ACZ groups and data users in each sharding ACZ, the number of rounds of iteration required for the algorithm execution increases gradually. From the plot, the algorithm has a linear growth trend, with good scalability.

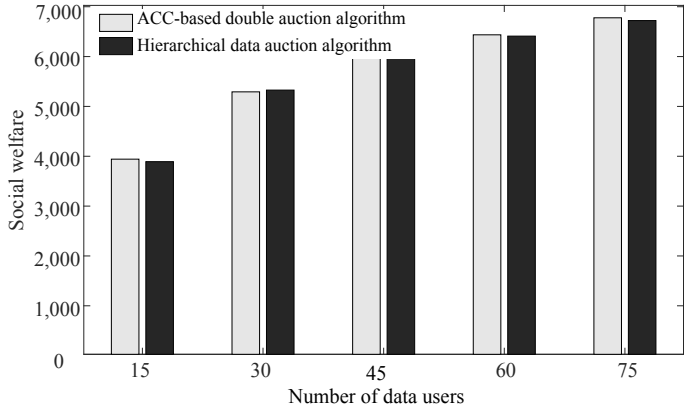


Figure 10 Average maximum social welfare in hierarchical data auction and double auction algorithms

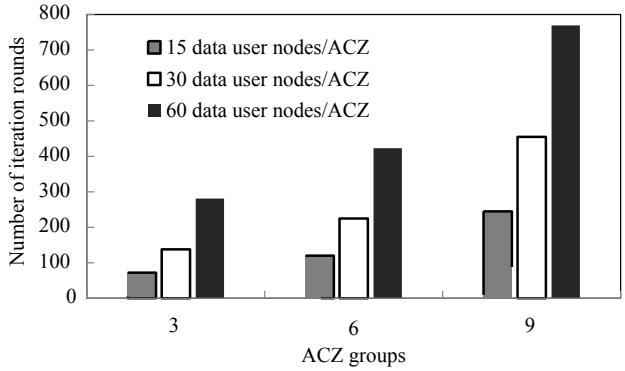


Figure 11 Average number of rounds with varying ACZ groups and numbers of data users in each ACZ

5 Summary and Prospect

In this paper, we studied an efficient IoT data sharing incentive scheme based on blockchain. On the one hand, the scheme proposes an efficient blockchain-empowered IoT data sharing incentive framework, called ShareBC. ShareBC relies on sharding technology to divide the IoT devices in the network into several ACZs, thereby enhancing the transaction processing capability of the system. It processes the transaction verification, which needs to be performed

by the whole network nodes, in parallel in each sharding ACZ. On this basis, ShareBC provides an efficient consensus mechanism, which is highly transparent and can be audited, with advantages in computation cost and scalability. On the other hand, the scheme proposes an incentive mechanism based on a hierarchical data auction model, which solves the problem of shared data resource allocation between data providers and data demanders. The data users that cannot access the shared resources themselves can access the resources via data agents. It aims to encourage more IoT users to join data sharing. In the hierarchical data auction mechanism, a three-layer data auction model is built, and relevant data allocation and pricing rules are designed. The impact of the data transmission cost on social welfare has also been investigated. Finally, to ensure the non-repudiation and execution efficiency of the auction mechanism, we deployed a smart contract to make the auction mechanism work automatically. Theoretical results and experimental evaluation show that the data sharing incentive scheme proposed in this paper has individual rationality, incentive compatibility, weak budget balance, real-time performance, and scalability, with low computing cost and good practicability.

Future studies are expected to explore blockchain-based data sharing in IoT. To remove the inherent performance bottleneck of blockchains, further research should be conducted on the system scalability such as sharding and off-chain payment channels. In this paper, we offer suggestions on the setting of ShareBC sharding protocols and compare the performances of different algorithms. In the future, we will continue to study the specific implementation of ShareBC sharding protocols and investigate how to form sharding in a dynamic IoT environment and how to dynamically adjust sharding while balancing decentralization, security, and scalability. In addition, multi-chain-driven heterogeneous IoT sharing application platforms can also improve the performance of blockchain systems. In the model proposed in this paper, ShareBC is based on permissioned blockchains, and data sharing transactions are performed in an environment assumed to be relatively trusted. In this environment, all participating nodes, such as data providers, possess the member qualification strategically authorized by the consortium organization. For follow-up research, we consider credit rating and reward-punishment practice in the incentive mechanism to improve the quality and transaction security of data sharing.

References

[1] Tan HB, Zhou T, Zhao H, *et al*. Archival data protection and sharing method based on blockchain. Ruan Jian Xue Bao/Journal of Software, 2019, 30(9): 2620–2635. http://www.jos.org.cn/1000-9825/5770.htm. [doi: 10.13328/j.cnki.jos.005770]

[2] IDC. IoT growth demands rethink of long-term storage strategies, says IDC. https://www.idc.com/getdoc.jsp?containerId=prAP46737220. [2020-7-28].

[3] Liu AD, Du XH, Wang N, *et al*. Blockchain-based access control mechanism for big data. Ruan Jian Xue Bao/Journal of Software, 2019, 30(9): 2636–2654. http://www.jos.org.cn/1000-9825/5771.htm. [doi: 10.13328/j.cnki.jos.005771]

[4] Aitzhan NZ, Svetinovic D. Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. IEEE Trans. on Dependable and Secure Computing, 2018, 15(5): 840–852. [doi: 10.1109/TDSC.2016.2616861]

[5] Li L, Liu JQ, Cheng LC, *et al*. CreditCoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles. IEEE Trans. on Intelligent Transportation Systems, 2018, 19(7): 2204–2220. [doi: 10.1109/TITS.2017.2777990]

[6] He YH, Li MR, Li H, *et al*. A blockchain based incentive mechanism for crowdsensing applications. Ji Suan Ji Yan Jiu Yu Fa Zhan/Journal of Computer Research and Development, 2019, 56(3): 544–554. [doi: 10.7544/issn1000-1239.2019.20170670]

[7] Yin H, Zhang X, Zhao S. Tradeoffs between cost and performance for CDN provisioning based on coordinate transformation. IEEE Trans. on Multimedia, 2017, 19(11): 2583–2596. [doi: 10.1109/TMM.2017.2696309]

[8] Liang L, Wu YF, Feng G. Resource allocation algorithm of network slicing based on online auction. Journal of Electronics and Information Technology, 2019, 41(5): 1187–1193. [doi: 10.11999/JEIT180636]

[9] Hong ZC, Chen WH, Huang HW, *et al*. Multi-hop cooperative computation offloading for industrial IoT-edge-cloud computing environments. IEEE Trans. on Parallel and Distributed Systems, 2019, 30(12): 2759–2774. [doi: 10.1109/TPDS.2019.2926979]

[10] Gao GJ, Xiao MJ, Wu J, *et al*. Truthful incentive mechanism for nondeterministic crowdsensing with vehicles. IEEE Trans. on Mobile Computing, 2018, 17(12): 2982–2997. [doi: 10.1109/TMC.2018.2829506]

[11] Pu LJ, Chen X, Mao GQ, *et al*. Chimera: An energy-efficient and deadline-aware hybrid edge computing framework for vehicular crowdsensing applications. IEEE Internet of Things Journal, 2019, 6(1): 84–99. [doi: 10.1109/JIOT.2018.2872436]

[12] Petrov V, Samuylov A, Begishev V, *et al*. Vehicle-based relay assistance for opportunistic crowdsensing over NarrowBand IoT (NB-IoT). IEEE Internet of Things Journal, 2018, 5(5): 3710–3723. [doi: 10.1109/JIOT.2017.2670363]

[13] Wu SK, Chen YJ, Wang Q, *et al*. CReam: A smart contract enabled collusion-resistant e-auction. IEEE Trans. on Information Forensics and Security, 2019, 14(7): 1687–1701. [doi: 10.1109/TIFS.2018.2883275]

[14] Zhang MM, Chen C, Wo TY, *et al*. SafeDrive: Online driving anomaly detection from large-scale vehicle data. IEEE Trans. on Industrial Informatics, 2017, 13(4): 2087–2096. [doi: 10.1109/TII.2017.2674661]

[15] Liang W, Li KC, Long J, *et al*. An industrial network intrusion detection algorithm based on multi-characteristic data clustering optimization model. IEEE Trans. on industrial Informatics, 2019, 2063–2071. [doi: 10.1109/TII.2019.2946791]

[16] Cai T, Chen WH, Yang ZT, *et al*. A blockchain-assisted trust access authentication system for solid. IEEE ACCESS, 2020. [doi: 10.1109/ACCESS.2020.2987608]

[17] Kang JW, Xiong ZH, Niyato D, *et al*. Towards secure blockchain-enabled Internet of vehicles: Optimizing consensus management using reputation and contract theory. arXiv: Cryptography and Security, 2018. [doi: arXiv:1809.08387]

[18] Yu Y, Ding YJ, Zhao YQ, *et al*. LRCoin: Leakage-resilient cryptocurrency based on bitcoin for data trading in IoT. IEEE Internet of Things Journal, 2019, 6(3): 4702–4710. [doi: 10.1109/JIOT.2018.2878406]

[19] Yang Z, Yang K, Lei L, *et al*. Blockchain-based decentralized trust management in vehicular Networks. IEEE Internet of Things Journal, 2019, 6(2): 1495–1505. [doi: 10.1109/JIOT.2018.2836144]

[20] Jia DY, Xin JC, Wang ZQ, *et al*. ElasticQM: A query model for storage capacity scalable blockchain system.Ruan Jian Xue Bao/Journal of Software, 2019, 30(9): 2655–2670. http://www.jos.org.cn/1000-9825/5774.htm. [doi: 10.13328/j.cnki.jos.005774]

[21] Cai T, Chen WH, Yu Y. BCsolid: A Blockchain-based Decentralized Data Storage and Authentication Scheme for Solid. Springer- Verlag, 2019. 676–689. [doi: 10.1007/978-981-15-2777-7_55]

[22] Hu ZY, Tang YJ, Yang ZG, *et al*. Improved scheme based on S-BAC cross-shard consensus protocol. Ji Suan Ji Application Research of Computers, 2019, 8(1): 1–6. [doi: 10.19734/j.issn.1001-3695.2019.10.0585]

[23] Pan JF, Huang DC. Blockchain dynamic sharding model based on jump hash and asynchronous consensus group. Ji Suan Ji Ke Xue/Computer Science, 2020, 47(3): 273–280. [doi: 10.11896/jsjkx.190100238]

[24] Qiu XY, Liu LB, Chen WH, *et al*. Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing. IEEE Trans. on Vehicular Technology, 2019, 68(8): 8050–8062. [doi: 10.1109/TVT.2019.2924015]

[25] Chen WH, Zhang Z, Hong ZC, *et al*. Cooperative and distributed computation offloading for blockchain-empowered industrial internet of things. IEEE Internet of Things Journal, 2019, 6(5): 8433–8446. [doi: 10.1109/JIOT.2019.2918296]

[26] Wang XM, Wu WW, Qi DY. Mobility-aware participant recruitment for vehicle-based mobile crowdsensing. IEEE Trans. on Vehicular Technology, 2018, 67(5): 4415–4426. [doi: 10.1109/TVT.2017.2787750]

[27] Ni JB, Zhang AQ, Lin XD, *et al*. Security, privacy, and fairness in fog-based vehicular crowd sensing. IEEE Communications Magazine, 2017, 55(6): 146–152. [doi: 10.1109/MCOM.2017.1600679]

[28] Xiao L, Chen TH, Xie CX, *et al*. Mobile crowdsensing games in vehicular networks. IEEE Trans. on Vehicular Technology, 2017, 67(2): 1535–1545. [doi: 10.1109/TVT.2016.2647624]

[29] Kiani A, Ansari N. Toward hierarchical mobile edge computing: An auction-based profit maximization approach. IEEE Internet of Things Journal, 2017, 4(6): 2082–2091. [doi: 10.1109/JIOT.2017.2750030]

[30] Wang LJ, Liu M, Meng MQH. A hierarchical auction-based mechanism for real-time resource allocation in cloud robotic systems. IEEE Trans. on Cybernetics, 2016, 47(2): 473–484. [doi: 10.1109/TCYB.2016.2519525]

[31] Jin AL, Song W, Wang P, *et al*. Auction mechanisms toward efficient resource sharing for cloudlets in mobile cloud computing. IEEE Trans. on Services Computing, 2016, 9(6): 895–909. [doi: 10.1109/TSC.2015.2430315]

[32] Wen YT, Shi JY, Zhang Q, *et al*. Quality-driven auction-based incentive mechanism for mobile crowd sensing. IEEE Trans. on Vehicular Technology, 2015, 4203–4214. [doi: 10.1109/TVT.2014.2363842]

[33] Dong XQ, Guo B, Shen Y, *et al*. An efficient and secure decentralizing data sharing model. Ji Suan Ji Xue Bao/Chinese Journal of Computers, 2018, 41(5): 1021–1036. [doi: 10.11897/SP.J.1016.2018.01021]

[34] Wang RH, Zhang LF, Xu QQ, *et al*. Byzantine fault tolerance algorithm for consortium blockchain. Ji Suan Ji Ying Yong Yan Jiu/Application Research of Computer, 2019, 37(11): 1–6. [doi: 10.19734/j.issn.1001-3695.2019.07.0268]

[35] Xu CH, Wang K, Li P, *et al*. Making big data open in edges: A resource-efficient blockchain-based approach. IEEE Trans. on Parallel and Distributed Systems, 2019, 30(4): 870–882. [doi: 10.1109/TPDS.2018.2871449]

[36] Li M, Weng J, Yang AJ, *et al*. CrowdBC: A blockchain-based decentralized framework for crowdsourcing. IEEE Trans. on Parallel and Distributed Systems, 2019, 30(6): 1251–1266. [doi: 10.1109/TPDS.2018.2881735]

[37] He YH, Li H, Cheng XZ, *et al*. A blockchain based truthful incentive mechanism for distributed P2P applications. IEEE ACCESS, 2018, 27324–27335. [doi: 10.1109/ACCESS.2018.2821705]

[38] Kang JW, Yu R, Huang XM, *et al*. Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains. IEEE Trans. on Industrial Informatics, 2017, 13(6): 3154–3164. [doi: 10.1109/TII.2017.2709784]

[39] Vijayakumar P, Obaidat MS, Azees MS, *et al*. Efficient and secure anonymous authentication with location privacy for IoT-based WBANs. IEEE Trans. on Industrial Informatics, 2020, 16(4): 2603–2611. [doi: 10.1109/tii.2019.2925071]

[40] Erdem E, Sandikkaya MT. OTPaaS—One time password as a service. IEEE Trans. on Information Forensics and Security, 2019, 14(3): 743–756. [doi: 10.1109/TIFS.2018.2866025]

[41] Ausubel LM. An efficient ascending-bid auction for multiple objects. The American Economic Review, 2004, 94(5): 1452–1475. [doi: 10.1257/0002828043052330]

[42] Danezis G, Meiklejohn S. Centrally banked cryptocurrencies. Proc. of the Conf. on Network and Distributed System Security Symposium (NDSS). 2016. 1–14. [doi: 10.14722/ndss.2016.23187]

[43] Al-Bassam M, Sonnino A, Bano S, *et al*. Chainspace: A sharded smart contracts platform. Proc. of the Conf. on Network and Distributed System Security Symp. (NDSS). 2018. 1–16. [doi: 10.14722/ndss.2018.23244]

[44] Luu L, Narayanan V, Zheng CD, *et al*. A secure sharding protocol for open blockchains. Proc. of the ACM SIGSAC Conf. on Computer and Communications Security. 2016. 17–30. [doi: 10.1145/2976749.2978389]

[45] Kokoris-Kogias E, Jovanovic P, Gasser L, *et al*. Omniledger: A secure, scale-out, decentralized

ledger via sharding. Proc. of the Symp.on Security and Privacy (SP). IEEE, 2018. 583–598. [doi: 10.1109/SP.2018.000-5]

[46] Zamani M, Movahedi M, Raykova M. Rapidchain: Scaling blockchain via full sharding. Proc. of the ACM SIGSAC Conf. on Computer and Communications Security. 2018. 931–948. [doi: 10.1145/3243734.3243853]

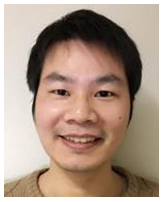
[47] Wang JP; Wang H. Monoxide: Scale out blockchains with asynchronous consensus zones. Proc. of the Symp.on Networked Systems Design and Implementation (NSDI). 2019. 95–112. [doi: 10.13140/RG.2.2.32017.48489]

Ting Cai Ph.D., associate professor. Her research interests include blockchain, IoT security, access control, and edge/cloud computing.

Zibin Zheng Ph.D., professor, doctoral supervisor, senior member of CCF. His research interests include blockchain, service computing, and software engineering.

Hui Lin master. Her research interests include the game theory, resource allocation of edge computing, and blockchain.

Yang Yu Ph.D., professor, doctoral supervisor, senior member of CCF. His research interests include workflow, service computing, cloud computing, and software engineering.



Wuhui Chen Ph.D., associate professor, CCF professional member. His research interests include edge/cloud computing, cloud robot, and blockchain.