

# A Decentralized and Non-reversible Traceability System for Storing Commodity Data

**Xiaojian He<sup>1</sup>, Ximeng Chen<sup>1</sup> and Kangzi Li<sup>1\*</sup>**

<sup>1</sup>School of Computer Science & Engineering, South China University of Technology  
Guangzhou 510006, Guangdong, P. R. China

[e-mail: hexj@scut.edu.cn, cxmzl@foxmail.com, likangzi@foxmail.com]

\*Corresponding author: Kangzi Li

*Received November 24, 2017; revised May 10, 2018; revised July 2, 2018; revised August 6, 2018;  
accepted September 1, 2018; published February 28, 2019*

---

## **Abstract**

In the field of traceability systems, researchers focus on applications in the agricultural food traceability and scanning commodities. The purposes of this paper, however, is to propose an efficient and reliable traceability system that can be applied to all kinds of commodities. Currently, most traceability systems store data in a central server, which is unreliable when the system is under attack or if the administrator tampers with the data for personal interests. Therefore, it is necessary to design a system that can eliminate these threats. In this paper, we propose a decentralized and non-reversible traceability system for storing commodity data. This system depends on blockchain technology, which organizes data in the form of chains without a central server. This chain-style storage mechanism can prevent malicious modifications. In addition, some strategies are adopted to reduce the storage pressure and response time when the system has stored all kinds of commodity data.

---

**Keywords:** Traceability system, decentralized, non-reversible, blockchain, data storage

## 1. Introduction

Traceability systems are applied widely to provide traceable data of food for quality and safety [1][2]. As a matter of fact, the application area of this system should be more extensive. Aside from the food industry, traceability systems could even play a role in storing various other kinds of commodity data. During an entire transaction process, commodity data could be used for a great many purposes. For example, customers could trace the source of a commodity to identify whether it is a counterfeit. In addition, commodity data could offer more dynamic and up-to-date evaluation information about transportation and usage, which could help consumers make better choices.

Furthermore, most researchers are concentrating on how to scan commodities and are utilizing various tools to improve this process, such as RFID and QR codes [3][4], and try to enhance the efficiency of scanning commodities to improve the user experience. In fact, the storage mechanism is another factor worth considering, because a more appropriate storage structure could bring about an obvious improvement in system performance.

In traditional centralized traceability systems, the characteristic of storing data in a central server is easily abused by the system administrator or can suffer directly malicious attacks. Most of the previous reports only consider tracing the specific data of transportation, which neglect of many the other types of commodity data. Furthermore, once the entire data is stored in a centralized system, storage space will be gradually filled with the growing amount of data, which inevitably causes trouble when indexing the target data and results in an increase in storage cost and query pressure.

In conclusion, to store commodity data more securely, a decentralized and non-reversible system is required. The development of blockchain technology provides an ideal technical method to not only realize the decentralization and non-reversibility mechanisms, but also to properly control the storage overhead. In this paper, we propose a traceability system with optimized storage strategies based on the mechanism of the blockchain. It records the detailed transaction processes of commodities.

In this system, traceability data is distributed across the relevant users' smart devices instead of being stored a central server during the trading process. Thus, without an obvious center, the data is evenly distributed and the storage pressure is scattered, which forms a decentralized system. Meanwhile, some of the nodes that have participated in transactions of the same commodity are connected to each other one by one as a transaction chain, which is non-reversible and prevents malicious modification.

The rest of this paper is organized as follows. Section 2 presents related work on the traceability system. Section 3 provides the architecture of this system and how the system manages nodes to maintain stability. Section 4 describes some strategies to optimize the storage structure in this system for better storage and query performance. Section 5 introduces an application scenario in this system. Section 6 carries out some experiments to analyzes the storage and query performance of this system. Finally, Section 7 concludes the paper and illustrates future work.

## 2. Related Work

As discussed above, the traceability system has been extensively studied for a long time. Kim et al. [5] emphasized that traceability could play an important role in quality control in 1995.

As time went by, the traceability system was gradually refined and became increasingly more mature in design and implementation. In this section, various traceability systems are introduced.

Folinas et al. [6] introduced a generic traceability data-management framework (architecture) which aims at automating and improving traceability using internet technologies. Bechini et al. [7] implemented a practical prototype of a traceability system that is able to trace and track product units and enable business collaborations, resulting in less time, cost, and errors when executing everyday transactions. They also created a common XML vocabulary for specific agribusiness documents to automate the definitions of business terms, conditions, and parameters. Because of the perishability of fresh agricultural products and food, more and more traceability systems were developed to deal with quality problems. Ruiz-Garcia et al. [8] implemented a prototype for tracking and tracing agricultural batch products. They used a client node to tackle tracing requests from users and subsequently sent them to a relevant service company to obtain the required information. The system could provide information about how the product is collected and processed for retailers and consumers. Pizzuti and Mirabelli [9] used the global track and trace information system (GTTIS) to store food data. They could manage the entire lifecycle to guarantee the origin and quality of food while ensuring compliance with regulations.

Furthermore, more researchers studied identifying and tracking tags, which are attached to objects and contain electronically stored information. Work in the field of electronics became one of the most crucial steps in traceability research, as it concerned the efficiency of digitizing products, which is realized by binding the tag with the product and then scanning the ID of the tag into the system. Bernardi et al. [10] used RFID technology to implement a rapid and effective traceability system. With a suitable RSA implementation, they brought about effective solutions to protect privacy in RFID, satisfying efficiency and privacy demands. Tarjan et al. [4] tested the readability of QR codes in various scenarios. They concluded that QR code readability is only directly influenced by the size of the code modules, code contents, and base material on which the QR code is printed.

With the development of the blockchain [11], many unsolved technical problems could be easily solved. Abeyratne and Monfared [12] used blockchain principles in a manufacturing supply chain. They introduced some financial, social, and legal applications based on blockchain technology. They also focused on creating transparency and traceability in the supply chain using blockchain technology by considering authority control and data access.

Tian [13] combined the RFID and blockchain technology in a traceability system to better guarantee the food safety. In the solutions proposed in this article, data storage is based on the existing blockchain technology and there is no corresponding solution for storage optimization. Benshoof et al. [14] utilized the blockchain to build a distributed decentralized DNS, which makes his system more robust and extensible, but also decentralized. Zyskind et al. [15] described a decentralized personal data-management system that ensures that users own and control their data. As an access-control moderator, their platform combines the blockchain with an off-blockchain storage solution to keep users' privacy safe. Wu [16] developed a distributed and scalable data model for the sharing of information during the transport phase of the supply chain based on blockchain technology. This work focuses on the data structure, but there are no relevant solutions for data storage.

Our purpose is to build this system to increase reliability and safety based on the mechanism of the blockchain, and to propose an optimized storage strategy to reduce the storage pressure and response time.

### 3. System Architecture

In this traceability system, the concepts of the block and chain refer to the theory of blockchain and are used to realize decentralization and non-reversibility. However, their structures in this paper are totally different from those in a traditional blockchain network. In order to be suitable for the traceability system, the original data structures are adjusted greatly.

In this system, the query mechanism is realized by the transaction chain. To identify whether a modification operation is malicious, two states, active and inactive, are introduced. Moreover, the introduction of two states could also help maintain the correctness of synchronization and consistency of the transaction chains.

#### 3.1 Transaction Chain

In this system, every registered smart device is regarded as a node. Each node generates and stores transaction data. The specific contents of the data will be discussed in Section 4. In addition, commodity data is stored in the form of the transaction chain. Every single chain, which is composed of multiple transaction blocks, stores the data of a single commodity. The chain starts with the genesis block, which stores the attribute data, and it will not end until the commodity is no longer being traded. To be more precise, a block represents a single transaction record, whereas a chain represents the entire transaction process of a single commodity, including various transaction records. [Table 1](#) provides detailed descriptions of the terms above.

Every time a user carries out transactions, he should first be registered in the system using his smart device, and then his device would be regarded as a node. Meanwhile, the transaction data involved will be transmitted and stored in his device. To conclude, every user participating in this transaction owns a duplicate of the transaction chain, which is stored in his node.

Let us closely examine the whole commodity data. Suppose there is a transaction process from manufacturer M to dealer A and finally to customer B, as shown in [Fig. 1](#).

**Table 1.** The descriptions of terms

Smart device	An electronic device with an operating system and Internet access.
Node	Every smart device that has been registered in this system is considered a node. However, different registered smart devices from one user will be treated as the same node in the system.
Block	A block stores the data of a single transaction. As a transaction occurs, a block will be generated.
Chain	A chain is organized by sequential blocks. It represents the entire transaction process of a single commodity. Once a commodity begins to be traded, a chain is generated to store the transaction process.

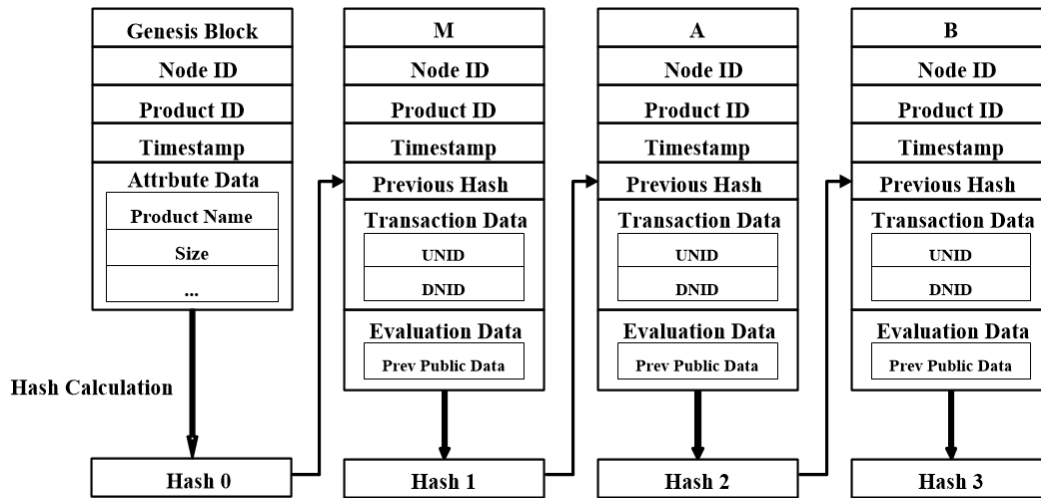


Fig. 1. The transaction chain

Fig. 1 is a transaction chain with hash pointers. The complete commodity data, which will be discussed in Section 4, is divided into three parts, namely the transaction data, attribute data, and evaluation data. It should be noted that each block generates a hash value, which will be a part of its next block. In this way, the last hash value of this chain could be considered a signal to identify whether this chain has changed, which could help prevent malicious modification. If some malicious users intend to tamper with a record in a node, the modification will be easily detected because the hash value will change accordingly. To keep the data consistent, the malicious user must modify the hash value from the beginning node to the very last one along the chain, which is almost impossible. Therefore, this tamper-evident chain could effectively prevent malicious addition and forgery of data.

### 3.2 Two States for System Availability

As mentioned above, commodity data is stored on smart devices. Therefore, the availability of the system depends on these nodes. To reasonably manage nodes and keep this system available at all times, two states, namely the active and inactive states, will be introduced.

Normally, the system is set to the inactive state, meaning that any modification in any node will be rejected except for adding a new node in a legal way, the details of which will be discussed in the next section. In this state, if any instruction is executed to cause the modification of data, such as deletion of the data in a node, the hash value in the current transaction chain will change immediately. In order to detect the state of this system, a core detection operation is periodically executed that compares the hash values in different nodes of the same chain. Therefore, operations causing a change in the hash value when the node is in the inactive state will be considered illegal. Subsequently, the data in other nodes will be copied to damaged nodes to restore their data.

In contrast, the adding jobs could be allowed in the inactive state. Once there is a new node added to the system, the system will change into the active state. After finishing the addition task, the system will return to the inactive state.

To distinguish the adding operation from malicious operation, the addition should be executed strictly following the prescribed steps mentioned in the next section. The entire

transformation process between the active and inactive state is described in Fig. 2. Attribute data and evaluation data are removed to make it clearer that the changes in the transaction data are proper enough to express the transformation of state. In this figure, the red parts change at each step. Once the downstream node ID (DNID) of the last node has been appointed, the active state is transformed to inactive. Similarly, when this newly registered node has been added into the chain, the active state will terminate and the system becomes stable again.

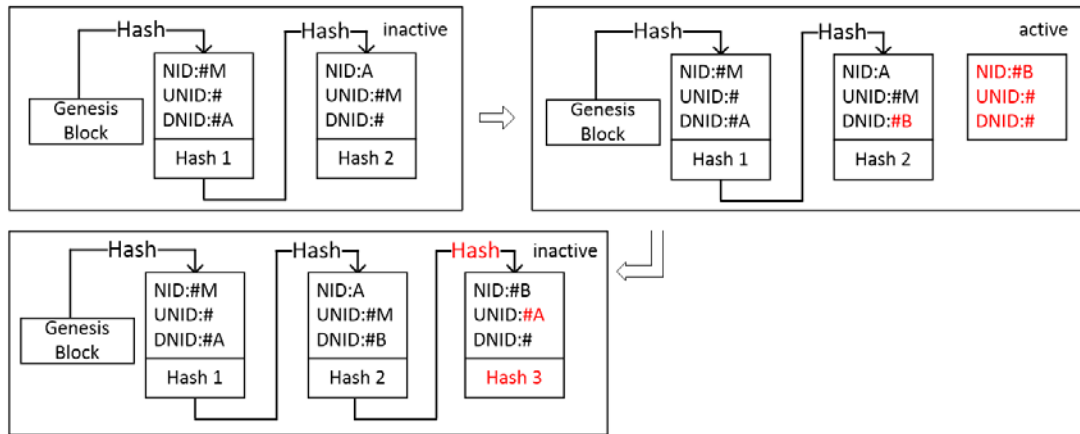


Fig. 2. State transformation process

### 3.3 Adding Procedure

After introducing all parts of this system, Table 2 shows the procedure of adding a new node and shows how it actually works. Fig. 3 indicates the main steps of the procedure.

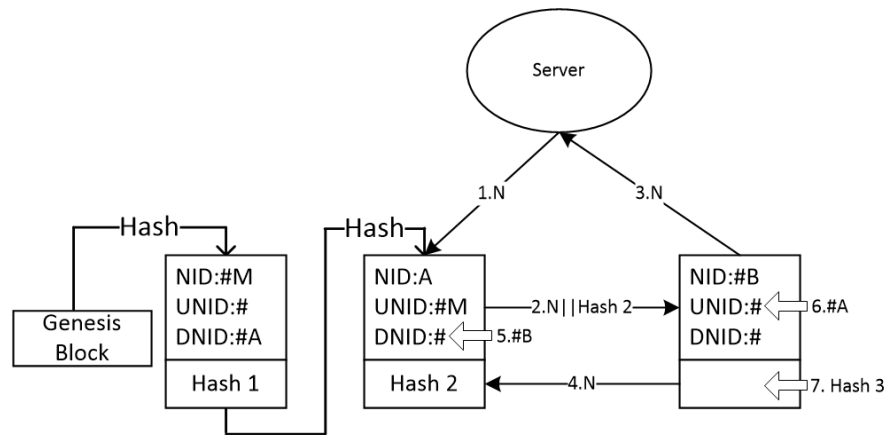
Table 2. The procedure of adding a new node

**Procedure** adding a new node in a chain

**function:** addNewNode()

**Inputs:** sellerId, buyerId, commodityId

- 1: token  $\leftarrow$  *applyToken*(sellerId)
- 2: prevHash  $\leftarrow$  *getHash*(commodityId, sellerId)  
&& *transfer*(sellerId, buyerId, token, prevHash)
- 3: buyerBlock  $\leftarrow$  *register*(buyerId, commodityId, token)
- 4: *transfer*(buyerId, sellerId, token)
- 5: sellerBlock.DNID  $\leftarrow$  buyerBlock.NID
- 6: buyerBlock.UNID  $\leftarrow$  sellerBlock.NID
- 7: buyerBlock.prevHash  $\leftarrow$  *hashGenerator*(buyerBlock)
- 8: *synchronizeChain*(commodityId)



**Fig. 3.** The steps of the adding job

According to [Table 2](#) and [Fig. 3](#), every time a commodity is sold, the previous certificated owner of the commodity should apply for a token to help the new owner register in the system. To prevent malicious usage, others are not allowed to obtain this token from the system. This token is a unique signal randomly generated based on the timestamp. It is kept secret from others so that it may be used to prove the authorization for the register in Step 3 and fill in the DNID field in Step 5. This mechanism could prevent a fake node from being added to this chain, because other nodes cannot obtain the token from the previous commodity owner.

Furthermore, the hash value in Step 2 will be a part of the data in node B. If the previous data is modified, the hash value of node B will change accordingly. In this way, every modification will be easily detected and restored.

In particular, when the system is in the inactive state, every operation will be refused except for the previous owner appointing its downstream node, which is indicated in Step 5. After node A applies for a token, it is authorized to fill in its DNID, which will set the system to the active state. After the new node B fills in its UNID, the new hash value could be generated with the node data and previous hash value. Therefore, a new chain is created that will be synchronized to every node related to this transaction. After these steps, the whole system will stay consistent and become inactive again.

#### 4. Optimized Storage Strategies

As time goes by, the commodity data is constantly changing and growing. To establish a system that can adapt to data expansion, we divide commodity data into three categories: transaction data, attribute data, and evaluation data.

In this section, we propose several optimized storage strategies to store the corresponding type of data reasonably. For transaction data, we use a data-synchronization storage strategy to ensure data consistency and non-tampering. For attribute data, we adopt a single-point storage strategy in the genesis node to avoid redundant storage of the same commodity. For evaluation data, we use the optimized storage strategy to update the data so that the owner can store the most complete and latest data of the commodity.

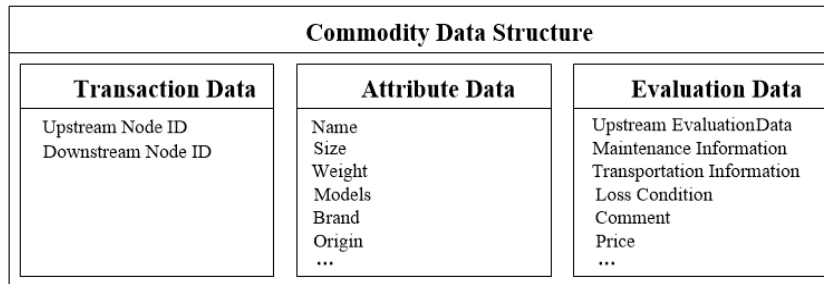
This system inherits the advantages of the blockchain technology and improves some parts of it for optimal storage performance. Compared with the blockchain, which stores all transaction records in a huge bill, this system stores parts of the records in users' smart

devices. To be clear, the entire data is divided into various parts and then stored in different devices. It should be noted that each user only stores the records of commodities that he has bought or sold.

Although the total volume of data in all devices is more than that stored in a centralized system, the storage pressure is evenly distributed across various devices such that the storage cost is allocated to every user.

Compared with the centralized system, this system apparently avoids limitless storage expansion and reduces the response time by scattering the storage pressure to every user. When a user queries the data of his purchased commodities, he could realize it locally rather than sending remote requests; this process greatly improves the efficiency. Meanwhile, mutual backup is beneficial for improving the disaster tolerance of this system.

In this system, the details of commodity data are organized in the structure shown in [Fig. 4](#). The total commodity data is composed of three parts: transaction data, attribute data, and evaluation data. The transaction data describes transaction relationships among various nodes, and supports the traceability of this system. The attribute data is a constant part that describes the initial information of commodities, such as weight and size. The evaluation data describes the dynamic information of commodities in the transaction processes, which helps consumers receive more information and make better choices.



[Fig. 4](#). Structure of commodity data

#### 4.1 Transaction Data

The transaction data, consisting of the upstream node ID and the downstream node ID, is the most important part of the data. These two fields point to each other between nearby blocks. Once one of these fields is modified, the hash value of the corresponding block will change accordingly. As mentioned above, if modification of the hash value is detected when the state is inactive, the restoration job will be activated to restore the modified nodes. On the other hand, after a transaction event has occurred, the system will be active, and the transaction data will be updated and copied as a backup to send to the downstream node.

#### 4.2 Attribute Data

The attribute data generally includes the commodity attributes. To reduce the storage burden, this system stores the attribute data in the genesis node of the transaction chain. This operation avoids repeated storage of the same commodity's attribute data in different nodes. When downstream nodes query the attribute data, they could directly send the request to the genesis node. The specific storage and query method are shown in [Fig. 5](#).



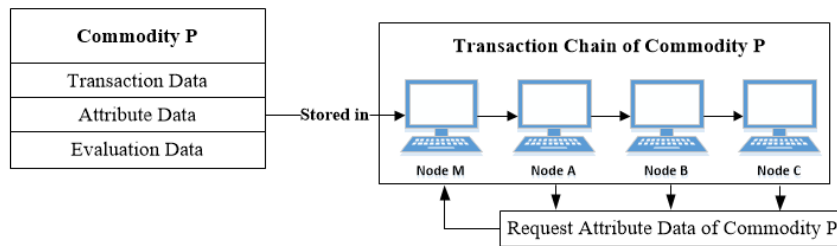


Fig. 5. Attribute data storage strategy

### 4.3 Evaluation Data

The evaluation data comprises data from the upstream nodes and the current node. It can provide commodity information during the entire transaction process, such as comments from each node and maintenance data. When a commodity is sold to the next node, the buyer will be authorized to visit the evaluation data of this commodity.

The evaluation data is gradually growing in the transaction process. To reasonably protect the privacy, the evaluation data is divided into two parts: the public and private data. The public part will be transferred to the downstream node, whereas the private data will be kept secret. Then, the downstream node can add its evaluation data to the chain. The main storage strategy is described in Fig. 6.

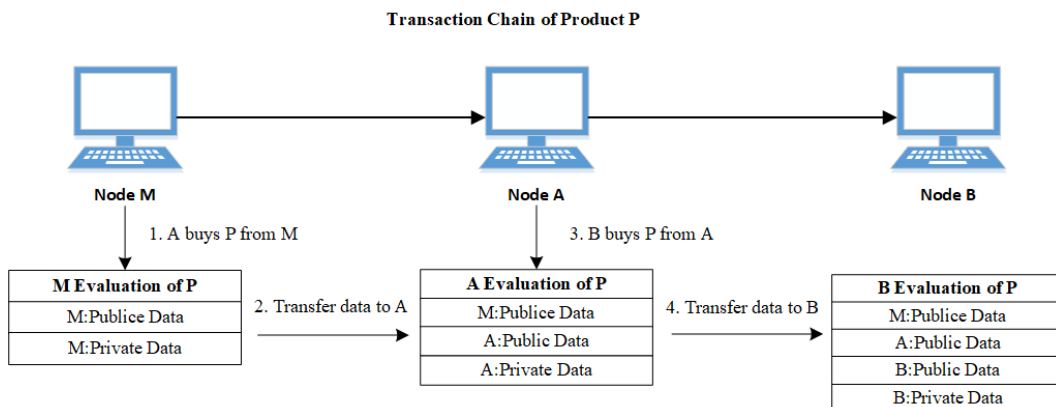


Fig. 6. Evaluation data storage strategies

Under this storage mechanism, each node simply stores the necessary piece of data instead of the full data. When the information is being passed to the next node, its extent of exposure is controlled by the node where it has been stored. Each user can decide what kind of data he would like to share with the public. In addition, when the data in a node is lost, it can be recovered from the backup, which is stored in the downstream node. This is a good way to reduce the risk of data loss.

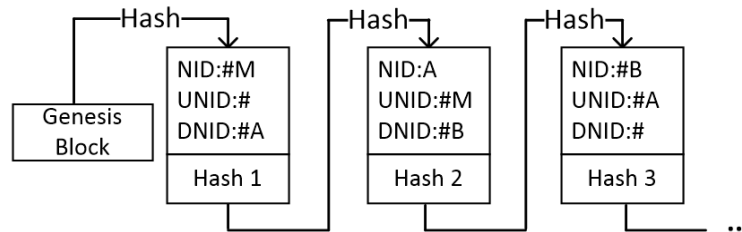
## 5. Application Scenario: Identifying Counterfeits

Counterfeits have a pernicious effect on the economy. The Organization for Economic Cooperation and Development (OECD) announced in 2005 that the counterfeits in international trade caused up to about US \$200 billion of loss [17], not to mention the counterpart data at present. With the help of traceability data, the trail of transaction

processes can be easily obtained, which can be applied in second-hand markets. For example, this blockchain method is an ideal way to identify whether a commodity is a counterfeit by analyzing the traceability data. It can easily detect the trick in which manufacturers attach famous brand logos to their commodities in order to benefit from the brand's influence while selling unauthorized replicas. Joseph et al. [18] have proposed a detection and reputation system on fake commodities for e-commerce, which is a centralized management system. His research presents a system that could effectively prevent consumers from being cheated when they trade in second-hand markets. Furthermore, it helps store the data in a more appropriate way.

Counterfeit manufacturers always attach their counterfeits to some legitimate manufacturers by modifying the commodity information. In this way, they successfully mix the fake product with the genuine. In fact, most counterfeits are produced by local small workshops and are illegal copies of famous brand commodities, and sell well in Thailand [19]. No matter where the sales venue is—a mall kiosk, a street vendor's stall, a flea market, or the Internet—it is not hard to find manufacturers who produce counterfeit luxury goods [20].

The situation can be optimized using this system. A consumer can trace the transaction chain back to the original block, which provides the original information of the commodity. In this way, he could check the manufacturer information to identify whether his purchased commodity is a counterfeit.

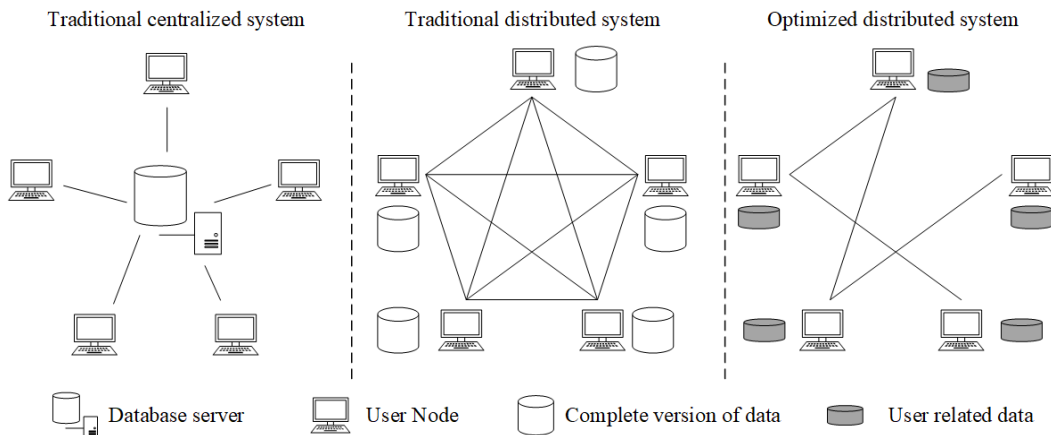


**Fig. 7.** The abstraction of transaction data in transaction chain

**Fig. 7** extracts the transaction data in a complete chain and forms a transaction chain. If someone has produced a counterfeit and intends to pass it off as a certificated commodity from manufacturer M to customer A, he must add his information in different nodes of the entire transaction chains of this commodity simultaneously. However, if the data is modified when the system is inactive, the modification will be reverted immediately. In this chain, both the downstream node of manufacturer M and the upstream node of consumer B are considered to be node A. Therefore, a counterfeiter has no way to modify its upstream node and downstream node as well as all related nodes that point to the target node at the same time.

## 6. Performance Analysis

To analyze the performance when adopting storage strategies, we compared this system with two other systems in terms of the storage space and response time. One of the control systems is a traditional centralized system [3] and the other is a distributed system based on the blockchain without an optimized storage strategy [13]. The architectural differences among the three systems are shown in **Fig. 8**.



**Fig. 8.** Architectures of different systems

The traditional centralized system is generally composed of a small number of central database servers, which depend on the configuration of the administrator. With the continuous increase in the scale of the network, the service requirements are also increasing. These central servers are required to back up and manage large amounts of data. In order to maintain the service quality, they should keep responding quickly, even if there exists a large number of query requests.

The traditional distributed system based on the blockchain stores a copy of the entire transaction record in each user node. With the increase in the amount of transaction data, every node in the network is supposed to update data, causing a large amount of network overhead.

The optimized distributed system proposed in this paper is based on the transaction chain. In this system, each node only stores the relevant transaction data. With the increase in the number of transactions, the increment of data in each node could be controlled to a small scale.

In order to simulate the actual trading relationships, we use real transaction data to form a trading network. In the initial phase of the experiment, 400 blocks of the blockchain were obtained from a publicly available blockchain data website (<https://blockchain.info/>), containing 1,321,946 transaction records from different accounts. Through the analysis of trading relationships, we obtained 583,495 trading relationship data points. **Table 3** shows the number of different kinds of transaction chains in the network.

**Table 3.** Four kinds of chains in a network

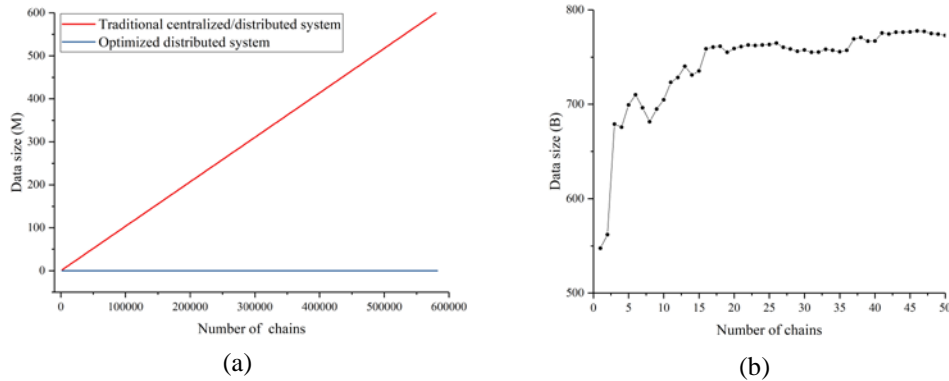
Trading chain type	Number of nodes in chain	Number of chains	Percentage
A>B	2	145,725	24.97%
C>D>E	3	392,108	67.20%
F>G>H>I	4	41,647	7.14%
J>K>L>M>N	5	4,015	0.69%

In **Table 3**, there are four kinds of chains in this network, and each chain represents the transaction relationship of one commodity among several nodes. In this network, transaction records involve 1,654,437 users and 583,495 commodities. Take the chain of three nodes in the second line of the **Table 3** as an example: a commodity is made by manufacturer C and

then sold to retailer D. Finally, it is a possession of customer E. Under this situation with our optimized storage strategies, the attribute data is only stored in node C, because the attribute data is stored by manufacturers. On the other hand, the transaction data is synchronized between users C, D, and E, meaning that their data is consistent. Moreover, the amount of evaluation data increases gradually in the process of the transaction.

## 6.1 Storage Space

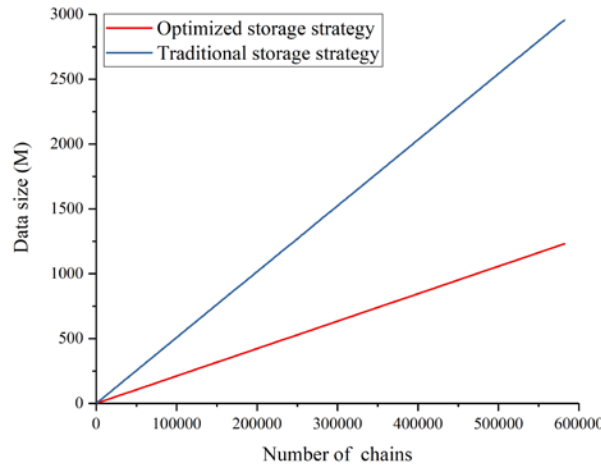
In the traditional centralized system without storage optimization, the central server must store all kinds of commodity data, which puts too much pressure on this node. In our system, the main data is scattered to every user. Therefore, each node faces less pressure on average. This experiment measures the trend of storage space when transactions increase from one chain to hundreds of thousands of chains. A chain represents a commodity, which means that this experiment regards the increase in commodities as an independent variable. Fig. 9 tells us how much the storage pressure will be reduced in a single node.



**Fig. 9.** (a) Comparison of data sizes in a single node on average under different systems. (b) The data size in a single node on average of a small network under the optimized distributed system.

Each node in the traditional centralized and distributed system stores the entire transaction data. When the number of chains increases, the storage pressure increases linearly. As time goes by, the rapid expansion and growth in data will greatly increase the burden and risk on the overall system. Each node in the optimized distributed system just stores their own transaction chains. It can be seen from Fig. 9 (a) that the storage space in this system performs well and the cost will be stable and less than 800 bytes after the network contains 20 chains, from Fig. 9 (b).

To represent the impact of our optimized strategy, which has been introduced from Section 4.1 to 4.3, we conduct another experiment to compare this strategy with the traditional strategy in our system. Fig. 10 shows the trend of total data sizes in all nodes under different storage strategies.



**Fig. 10.** Comparison of total data sizes under different storage strategies in our system

In the optimized storage strategy, the attribute data is only stored by manufacturers’ nodes. The transaction data will constantly expand and grow. In this system, each node only stores its related transaction data, including the commodities bought and sold. Meanwhile, each node stores the public evaluation data from its upstream nodes, which spares the storage space for private data. Therefore, these three kinds of data all benefit from the optimized storage strategies. Compared with the system without storage optimization, the amount of data is greatly reduced by about 50% to 60% in a single chain.

In order to make a clearer comparison of the storage performance of the three architectures, we use **Table 4** to describe them.

**Table 4.** Attributes of systems

Attribute	Traditional Centralized System	Traditional Distributed System	Optimized Distributed System
Number of Nodes	Depend on the administrator	Number of users	Number of users
Data Growth Rate	Linear Growth	Exponential Growth	Linear Growth
Storage Pressure of the Whole Network	Middle	High	Low
Storage Pressure of a Single Node	High	High	Low

**6.2 Response Time**

Compared with the traditional centralized system, this system could bring an obvious improvement in response time. When querying the transaction data, the network cost and query cost are the main factors affecting the performance. In a central server, every query action will bring both a network and query cost. Furthermore, its query should be executed among all kinds of data. However, in this system, transaction data could be realized locally in a small scale, saving transmission time on the Internet. Compared with the traditional distributed system based on the blockchain, network cost will be needed when users need attribute data in this system. However, queries in the smaller dataset will make up for the performance loss.

In order to represent the advantages in terms of query time, considering the difficulty of objective query response measurement in the complex network environment, we use the symbols  $Q$  and  $N$  to represent the time cost of one query and one network transmission in the centralized system, respectively. Considering that the local query time is much less than the query time of the central server, it is more believable that  $Q_1 > Q_2 \approx Q_3$ .

**Table 5.** Time cost of systems

Time Cost	Attribute Data			Other Data		
	Centralized System	Distributed System	Optimized System	Centralized System	Distributed System	Optimized System
Query	$Q_1$	$Q_2$	$Q_3$	$Q_1$	$Q_2$	$Q_3$
Network	$2N$	0	$2N$	$2N$	0	0
Total	$Q_1+2N$	$Q_2$	$Q_3+2N$	$Q_1+2N$	$Q_2$	$Q_3$

**Table 5** shows the comparison of the response time in different systems, and it can be concluded that the optimized system has better performance than the traditional centralized system in various queries. Compared with the distributed system, they have a similar performance. From the above analysis,  $Q_1$  is much greater than  $Q_3$  because  $Q_1$  is the result of a query in a larger amount of data. In terms of the attribute data, the result is more obvious.

## 7. Conclusion and Future Work

There exist several problems in traditional centralized traceability systems, such as threats from malicious data attackers and administrators. Furthermore, they are not used to store the entire commodity data. Even when storing the entire data in a central server, when the storage space is gradually filled, the query efficiency and storage cost will be impacted greatly. To solve these problems, this paper introduces a decentralized and non-reversible traceability system for commodity transaction. By referring to the mechanism of the blockchain, we store the commodity data by distribution in users' devices to realize the decentralization of this system. With the optimized storage strategies, storing data in the form of chain, this system could be non-reversible. In addition, the architecture could effectively reduce the storage pressure and response time.

However, there exist some unsolved questions. The system has not been fully realized. Although we have tested and analyzed its storage space and response time, its transaction time, time delay, and throughput still need to be verified. As a traceability system, the privacy of commodity data is another essential task. The open characteristic of the blockchain would cause problems with the protection of privacy. To solve this, some privacy strategies should be researched and utilized.

## References

- [1] Monique H. Jansen-Vullers, Christian A. van Dorp, and Adrie J.M. Beulens, "Managing traceability information in manufacture," *International journal of information management*, vol. 23, no. 5, pp. 395-413, October, 2003. [Article \(CrossRef Link\)](#)

- [2] Fabrizio Dabbene, and Paolo Gay, "Food traceability systems: Performance evaluation and optimization," *Computers and Electronics in Agriculture*, vol. 75, no. 1, pp. 139-146, January, 2011. [Article \(CrossRef Link\)](#)
- [3] Ruey-Shun Chen, C. C. Chen, K. C. Yeh, Y. C. Chen, and C. W. Kuo, "Using RFID technology in food produce traceability," *WSEAS Transactions on information science and applications*, vol. 5, no. 11, pp. 1551-1560, November, 2008. [Article \(CrossRef Link\)](#)
- [4] Laslo Tarjan, Ivana Šenk, Srdjan Tegeltija, Stevan Stankovski, and Gordana Ostojic, G., "A readability analysis for QR code application in a traceability system," *Computers and Electronics in Agriculture*, vol. 109, pp. 1-11, November, 2014. [Article \(CrossRef Link\)](#)
- [5] Henry M. Kim, Mark S. Fox, and Michael Gruninger, "An ontology of quality for enterprise modelling," in *Proc. of Enabling Technologies: Infrastructure for Collaborative Enterprises, 1995, Proceedings of the Fourth Workshop on. IEEE*, pp. 105-116, January, 1995. [Article \(CrossRef Link\)](#)
- [6] Dimitris Folinias, Ioannis Manikas, and Basil Manos, "Traceability data management for food chains," *British Food Journal*, vol. 108, no. 8, pp. 622-633, November, 2006. [Article \(CrossRef Link\)](#)
- [7] Alessio Bechini, Mario G.C.A. Cimino, Francesco Marcelloni, and Andrea Tomasi, "Patterns and technologies for enabling supply chain traceability through collaborative e-business," *Information & Software Technology*, vol. 50, no. 4, pp. 342-359, March, 2008. [Article \(CrossRef Link\)](#)
- [8] Luis Ruiz-Garciaa, G. Steinbergerb, and M. Rothmund, "A model and prototype implementation for tracking and tracing agricultural batch products along the food chain," *Food Control*, vol. 21, no. 2, pp. 112-121, February, 2010. [Article \(CrossRef Link\)](#)
- [9] Teresa Pizzuti, and Giovanni Mirabelli , "The Global Track&Trace System for food: General framework and functioning principles," *Journal of Food Engineering*, vol. 159, pp. 16-35, August, 2015. [Article \(CrossRef Link\)](#)
- [10] P. Bernardi, C. Demartini, F. Gandino, B. Montrucchio, M. Rebaudengo, and E.R. Sanchez, "Agri-Food Traceability Management using a RFID System with Privacy Protection," in *Proc. of Advanced Information Networking and Applications, 2007. AINA'07. 21st International Conference on. IEEE*, pp. 68-75, May, 2007. [Article \(CrossRef Link\)](#)
- [11] Satoshi Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," available: <https://bitcoin.org/bitcoin.pdf>, 2009. [Article \(CrossRef Link\)](#)
- [12] Saveen A. Abeyratne, and Radmehr P. Monfared, "Blockchain ready manufacturing supply chain using distributed ledger," *International Jour-nal of Research in Engineering and Technology*, vol. 5, no. 9, pp. 1-10, September, 2016. [Article \(CrossRef Link\)](#)
- [13] Feng Tian, "An agri-food supply chain traceability system for China based on RFID & blockchain technology," *Service Systems and Service Management (ICSSSM)*, in *Proc. of 2016 13th International Conference on. IEEE*, pp. 1-6, June, 2016. [Article \(CrossRef Link\)](#)
- [14] Brendan Benshoof, Andrew Rosen, and Anu G. Bourgeois, "Distributed Decentralized Domain Name Service," in *Proc. of 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). IEEE*, pp. 1279-1287, May, 2016. [Article \(CrossRef Link\)](#)
- [15] Guy Zyskind, Oz Nathan, and Alex 'Sandy' Pentland, "Decentralizing Privacy: Using Blockchain to Protect Personal Data," in *Proc. of Security and Privacy Workshops (SPW), 2015 IEEE. IEEE*, pp. 180-184, May, 2015. [Article \(CrossRef Link\)](#)
- [16] Haoyan Wu, "A Distributed Blockchain Ledger for Supply Chain," *Doctoral dissertation, Purdue University*, August, 2017. [Article \(CrossRef Link\)](#)
- [17] Danny Scorpecci, "The economic impact of counterfeiting and piracy," in *Proc. of Presentation at the conference 'Transatlantic IP Collaboration' on April*, vol. 27, 2009. [Article \(CrossRef Link\)](#)
- [18] Ayana Joseph, Divya V R, and Linda Sara Mathew, "Fake Product Detection and Reputation System for E-commerce," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 2, no. 3, pp. 300-304, May, 2016. [Article \(CrossRef Link\)](#)

- [19] Jon O. Neher, “Knock-off goods,” *Evidence-based practice*, vol. 20, no. 1, pp. 2, January, 2017. [Article \(CrossRef Link\)](#)
- [20] Connie D. Powell, “We all Know it's a Knock Off - Re-Evaluating the Need for the Post-Sale Confusion Doctrine in Trademark Law,” *North Carolina Journal of Law & Technology*, vol. 14, no. 1, pp. 1-42, October, 2012. [Article \(CrossRef Link\)](#)



**Xiaojian He** is an associate professor in the School of Computer Science & Engineering, South China University of Technology, China. His research interests include distributed computing, business intelligence, machine learning, and software development. Xiaojian He has a PhD in Computer Software and Theory from Shanghai Jiao Tong University, China.



**Ximeng Chen** received his B.S. degree from South China University of Technology in 2015. He received his M.S. degree in computer technology from South China University of Technology in 2018. His research interests include blockchain and software development.



**Kangzi Li** received the B.S. degree from South China University of Technology in 2016. She is currently pursuing the M.S. degree in computer technology from South China University of Technology, Guangzhou, China. Her research interests are in the areas of blockchain application and distributed key management.