International Journal of Information Systems and Supply Chain Management
Volume 16 • Issue 1

A Quality Traceability System for Fruit and Vegetable Supply Chain Based on Multi-Chain Blockchain

Feng Xue, Zhengzhou University of Economics and Business, China*
Fangju Li, Zhengzhou University of Economics and Business, China

ABSTRACT

To enhance the efficiency and security of traceability in the fruit and vegetable industry, this paper proposes an optimized model based on multi-chain blockchain technology. Firstly, an analysis is conducted on the supply chain information of the fruit and vegetable industry, where traceability codes and product information from various stages of the supply chain are organized and extracted. Next, a trusted traceability optimization model is established based on blockchain technology. Finally, an information traceability system for the VFSC is implemented using Hyperledger fabric, and an improved Kafka load balancing algorithm is proposed to enhance message transmission efficiency. Simulation results demonstrate that the multi-chain traceability model proposed outperforms traditional single-chain blockchain models in terms of query efficiency when the number of data records exceeds 1000. After 10000 data records are deployed on the blockchain, the efficiency of the multi-chain model is improved by over 90% compared to the traditional single-chain model.

KEYWORDS

Blockchain, Consensus Mechanism, Fruit and Vegetable Supply Chain, Hyperledger Fabric, Smart Contracts, Traceability Efficiency

INTRODUCTION

In the 1980s, European experts proposed the concept of food traceability to investigate the causes of mad cow disease (Chen et al., 2015). Since then, Europe has embarked on building a food traceability system, with the issuance of the Food Safety White Paper in 2000 and the Food Basic Law in 2002, laying the legal foundation for the system. To reduce food safety incidents, China advocates the gradual establishment of a legal system for food traceability and has issued relevant regulations and guidelines. Product safety and food loss are essential in the food supply chain. However, it is challenging to achieve centralized supply chain management due to the involvement of multiple stakeholders and the complexity of the links in the food supply chain.

DOI: 10.4018/IJISSCM.330681 *Corresponding Author

Moreover, the information generated in the supply chain is complex, and traditional information management cannot cover all aspects, leading to information asymmetry and difficulty building trust among supply chain members (Jum'a et al., 2023). Therefore, despite the establishment of food traceability systems and sound legal regulations by governments worldwide, fraud, adulteration, and contamination still occur in the food supply chain (Rogerson & Parry, 2020). In the fruit and vegetable supply chain (FVSC), multiple entities such as agricultural planting companies, logistics companies, regulatory authorities, and consumers are often involved. The traceability process has the characteristics of multiple points, long lines, and comprehensive coverage, and the traceability data has the characteristics of multiple sources and heterogeneity. The high complexity and strong data coupling make it particularly challenging to establish a quality traceability system for fruits and vegetables (Francois et al., 2020).

In research on the traceability of agricultural product supply chains, some studies have used technologies such as radio frequency identification (RFID), QR codes, isotope technology, and mobile wireless monitoring to provide a full-chain traceability system for the supply chain (Hua et al., 2013), and established a food quality and safety traceability system for the supply chain (Zhang et al., 2017), meeting the requirements for traceability and food quality and safety of agricultural products. However, traditional traceability technologies still face the following problems: the information silo effect caused by different underlying platforms of various manufacturers in the supply chain and inadequate anti-tampering capabilities of the traceability system.

Blockchain technology is often used in traceability systems due to its tamper-proof and distributed characteristics. Its distributed nature can break down departmental barriers and enable information sharing (Wu et al., 2023). Aung et al. (2014) defined traceability and traceability objectives, identified the driving factors for traceability in the food supply chain, analyzed the traceability requirements related to food safety and quality, and constructed a conceptual framework for a food traceability system. Lin et al. (2019) designed a food safety traceability system based on blockchain and EPCIS information services and provided a prototype system framework. Pearson et al. (2019) discussed the importance of food safety traceability. They designed a food supply chain using distributed ledger technology (DLT) based on blockchain, demonstrating the system architecture of the data flow from producers to end-users. Feng et al. (2020) reviewed the technical characteristics of blockchain technology and designed an architecture for a food traceability system based on blockchain technology, along with an applicability flowchart.

Furthermore, some researchers have explored the application of blockchain technology in designing regulatory systems. Galvez et al. (2018) assisted governments in establishing cross-industry or cross-departmental professional systems under blockchain technology, providing application suggestions for regulating the food supply chain. Mao et al. (2018) constructed an evaluation system based on blockchain technology to strengthen food supply chain supervision and management effectiveness, assisting regulatory agencies in collecting more reliable and authentic information. Leng et al. (2018) established a public blockchain system for the agricultural supply chain based on a dual-chain architecture to improve the credibility of the public service platform and the overall efficiency of the system. Hao et al. (2020) constructed a food safety risk assessment model that combines blockchain technology and visualization, using seafood as an example, and developed risk area regulatory management strategies.

Currently, the problem with the quality traceability system for fruits and vegetables is the low trust level among the stakeholders. The different entities in the supply chain are in an interest-based game-theoretic relationship, and there is asymmetric information between them, which leads to low trust and high trust costs for traceability data (Truong et al., 2021). Different participating entities may have varying interests and objectives in a supply chain. They may prioritize maximizing their benefits and be reluctant to share complete, accurate traceability data. This lack of willingness to share information can hinder information flow, leading to information asymmetry. This information asymmetry leads to mistrust in traceability data and increased costs in building a trustworthy

traceability system. Therefore, a game-theoretic relationship exists among the participating entities in the supply chain, where each entity adopts a cautious and conservative approach to protect its interests. They may be unwilling to make excessive commitments regarding data sharing and collaboration. This game-theoretic relationship contributes to traceability data's low trustworthiness and higher trust costs. Traditional quality traceability systems use centralized databases, and the data in each link of the traceability process is managed independently by the enterprise, which may result in information loss and tampering. When disputes arise, providing evidence and assigning responsibility is difficult, a technical problem that must be solved urgently (Caro et al., 2018). Some scholars have researched enhancing the credibility of traceability data (Dabbene et al., 2014).

In recent years, research on enhancing the credibility of traceability data has mainly focused on reducing human intervention in the data collection process through IoT data acquisition devices, preventing tampering during data transmission through asymmetric encryption algorithms, and ensuring data authenticity through multi-node verification in distributed systems for data storage and processing. With the development of blockchain technology, more and more research is being conducted on using blockchain's natural ability to prevent data forgery and tampering to ensure the credibility of quality traceability system data (Köhler, & Pizzol., 2020; Brice et al., 2020). However, blockchain platforms' slow data reading speed often restricts their application in systems such as fruit and vegetable quality traceability that require a large amount of data reading. Therefore, research is needed to develop solutions that ensure data credibility and improve data reading efficiency based on application characteristics (Balaji & Arshinder, 2016). Some approaches utilize quantum key distribution technology to enhance data security and combine it with Hazard Analysis and Critical Control Points (HACCP) to ensure product safety (Guo & Yu, 2022). However, this will sacrifice some efficiency to guarantee data security. When blockchain and its storage methods are applied to traceability systems, multiple traversals are required to retrieve data, resulting in low traceability query efficiency (Zhang et al., 2022). As a result, blockchain application in FVSC traceability system is limited.

The proposed solution to the problem, as mentioned earlier, is a fruit and vegetable quality traceability system based on a dual-chain blockchain, and the main contributions of this paper are listed as follows.

The paper analyzes, organizes, and extracts the primary information of each business link in the fruit and vegetable supply chain to improve the query efficiency of the FVSC traceability model based on blockchain. Based on blockchain technology, an FVSC traceability storage optimization model is established, a multi-chain fast query mode is designed, and a supply chain information traceability smart contract is formulated to construct an FVSC information traceability system.

2) To address the problem of decreased efficiency in consumer message processing due to traditional Kafka load balancing, an improved Kafka consensus algorithm optimization mechanism is proposed. By coordinating the deployment of a coordinator to assign work, the Kafka load-balancing process is optimized, the corresponding relationship between consumers and partitions is dynamically adjusted based on their respective quantities, and load balancing is initiated as needed to adjust the working state of the system.

The remaining parts of this paper are organized as follows: Section 2 introduces the relevant technical foundations of blockchain and smart contracts, laying a foundation for the subsequent system design. Section 3 explains the proposed FVSC traceability system based on a multi-chain blockchain and the improved Kafka algorithm. Section 4 compares the performance of the proposed multi-chain method with the traditional single-chain method through experiments and compares the improved consensus algorithm and the original algorithm. Finally, Section 5 summarizes the entire paper.

BACKGROUND

In 2008, a scholar named "Satoshi Nakamoto" first defined Bitcoin (Budish, 2022). The information exchange in the Bitcoin network can be completed without the involvement of any third party. Its essence is a distributed database system that includes all the shared transaction records among participating members (Ruan et al., 2021). Because most members jointly determine each transaction, false transactions cannot be collectively confirmed. It is difficult to tamper with once the blockchain creates and accepts a record (Li et al., 2021). This tamper-proofing is also a crucial reason blockchain is applied to grain and oil food traceability. According to different application scenarios, the blockchain development process can be divided into stages 1.0-3.0, as shown in Table 1.

A blockchain can be divided into public, private, and consortium chains. A public blockchain is generally considered "completely decentralized," meaning each node can join the network system at any time and place to read data and compete for accounting rights (Monrat et al., 2019). The level of decentralization of private blockchain is lower, and a particular institution or administrator controls its permissions. Based on different needs, selective read-write permissions are selectively opened to the outside, which is suitable for internal data management of specific institutions. The degree of decentralization of consortium blockchain is between the above two types of chains. Its data is only accessible to consortium members, achieving "partial decentralization" (Yue et al., 2021). Blockchain features benefits such as decentralization, traceability, tamper resistance, and anonymity. Decentralization means that a single node does not control the blockchain but relies on a consensus mechanism for verification and confirmation, making it almost impossible for nodes to tamper with data. The basic structure of a blockchain consists of chained blocks with timestamps, as shown in Fig. 1, which have robust traceability. Anonymity can protect user privacy to a certain extent, but blockchain cannot provide complete privacy protection. All nodes can view each transaction, and by analyzing the transactions that occur at each address, many relationships between users can be discovered.

Smart contracts are computer programs that run on a blockchain and automatically enforce contract terms when predetermined conditions are met, achieving the goal of "code is law" (Shao et al., 2020). In the Ethereum public chain, smart contracts can run on all nodes in the network, and no organization or individual can forcibly stop them. In the consortium chain, due to the limited number of nodes and the need to improve system efficiency, nodes do not compete for accounting rights, and transactions generated by smart contracts are recorded directly on the blockchain. Smart contracts expand the functions of blockchain and enrich the upper-layer applications of blockchain. After writing smart contract code according to business logic, it must be deployed to blockchain network nodes (Ratta et al., 2021).

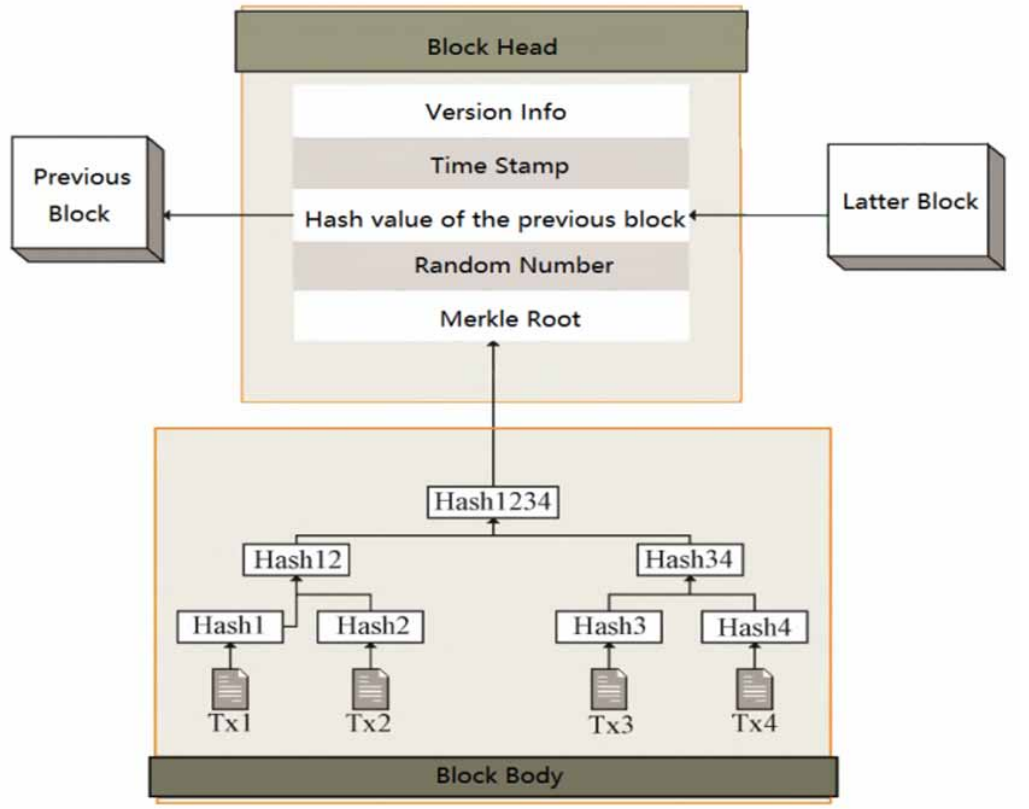
PROPOSED METHOD

The architecture of the FVSC information traceability system is presented in Figure 2. The system architecture consists of six layers: the Web application layer, the interface layer, the smart contract layer, the storage layer, the network layer, and the business layer.

Table 1. Blockchain development history

Stages	Period	Features
Blockchain 1.0	2008~2014	Cryptocurrency
Blockchain 2.0	2014~2016	Smart Contract + cryptocurrency
Blockchain 3.0	2016~	Smart Contract + Trust Mechanism

Figure 1. Block structure

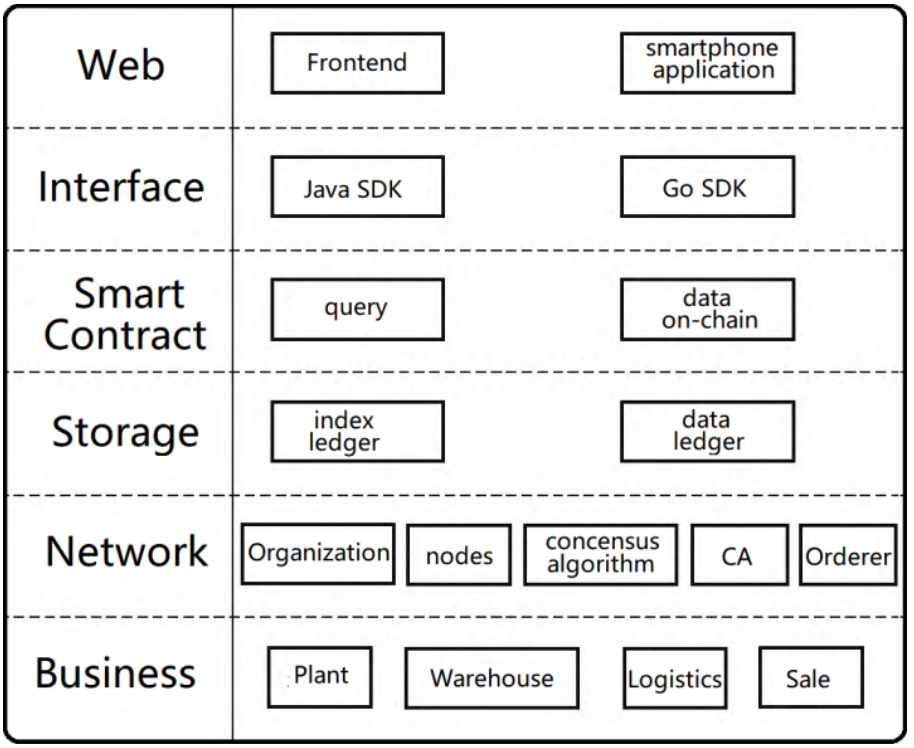


The application layer mainly presents the system to consumers through a visual interface, where consumers can use the front-end to input traceability codes for the traceability of fruits and vegetables. The interface layer connects the Web application layer and the smart contract layer, and the vendors in each link call the relevant functions of the smart contract through the program interface to implement data on-chain. The smart contract layer writes data query and data on-chain functions and is installed in the multi-chain model, enabling the smart contract to update the index ledger and data ledger. The storage layer stores the data structure of the multi-chain model, mainly including the index ledger and data ledger. The network layer includes relevant network nodes, organizations, consensus algorithms, Certification authority (CA), and Order nodes of the Hyperledger Fabric blockchain. The business layer connects the information systems of the original suppliers in each link, and vendors in each link put traceability information on the chain.

FVSC Traceability Optimization Model Architecture

When using a single-chain structure for blockchain traceability models, data from different links must be put separately on-chain. The on-chain data from each link will not continuously exist on the blockchain, resulting in a need to traverse the entire chain sequentially to find all traceability data during queries. As the amount of data increases, the speed of sequential traversal will increase linearly, resulting in a slower query speed (Kalodner et al., 2020). In this paper, based on the characteristics of FVSC and practical traceability needs, we construct an FVSC information traceability model based on blockchain and smart contract techniques, as shown in Figure 3. The traceability model manages the information of all supply chain production links and stores the mapping relationship's

Figure 2. System architecture

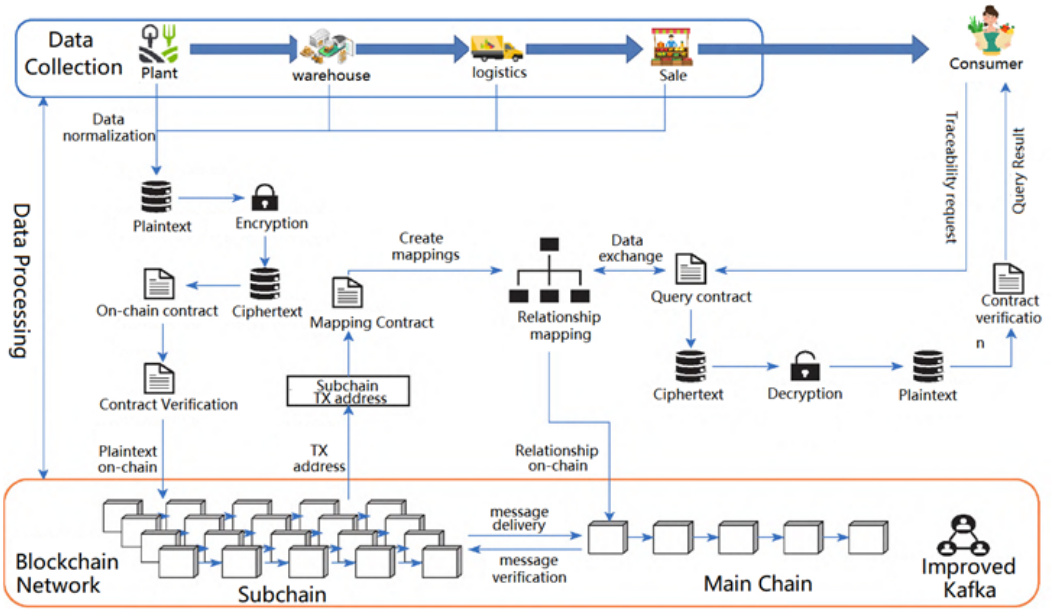


root node information in the main chain and the leaf node information in the sub-chain to achieve fast traceability queries. The model stores the data and associated information of all production link vendors in a decentralized environment, ensuring the tamper resistance, trustworthiness, and security of traceability data while ensuring the efficiency of traceability queries.

The model can be divided into three modules: supply chain data collection, data processing, and blockchain network. The supply chain data collection module collects data from various links through sensors, and then the data is standardized into the initial raw data. The data processing module consists of four parts: data on-chain, data mapping, data query, and data verification. The data on-chain process is carried out by nodes from various vendors at each link, utilizing the on-chain smart contract to synchronize and upload data onto the corresponding sub-chains. The design of on-chain smart contract will be explained in detail in Section 3.3. The hazard information must also call the verification contract to verify whether the significant hazard meets the requirements and whether the limit information exceeds the limit value. If all are qualified, the data is normally on-chain. Otherwise, the corresponding error message is returned. Consumers perform the data query through a client node to query the mapping relationship through the main chain and quickly locate the transaction data in each sub-chain through the transaction address in the mapping relationship. The data verification part verifies the data in the transaction, concatenates all the data into complete fruit and vegetable traceability data, and verifies the data integrity.

The blockchain network module consists of features such as sub- and main-chain block files and consensus algorithms to maintain data storage and queries. When data is on-chain, after collecting the traceability information, the vendor calls the data encryption module through the on-chain contract to convert the plaintext data into ciphertext data and then submits the on-chain request to transfer the ciphertext data to their respective sub-chains. Storing ciphertext data in sub-chains can prevent

Figure 3. System architecture



data leakage between upstream and downstream. After receiving the request, the sub-chain generates a transaction with the ciphertext data and stores the transaction in the current block. The sub-chain on-chain contract returns the transaction hash address and fruit and vegetable traceability code to the corresponding link node, and the node calls the mapping contract to establish a mapping relationship between the traceability code and each transaction. Finally, the fruit and vegetable mapping relationship is stored in the main chain. At this point, the traceability information on-chain phase is complete.

When tracing the data, consumers need to input the traceability code on the packaging of the fruit and vegetables into the client node. After verifying the format of the traceability code, the client node transfers it to the underlying network of the blockchain. By calling the mapping contract to query the main chain, the client node can obtain the mapping relationship and return it to the node. Through the mapping relationship, the client node can get the transaction hash address of each sub-chain. Then, the node queries each sub-chain separately through the transaction address index and can directly locate the transaction information of a specific block through the transaction address. After obtaining the ciphertext data of each sub-chain, the node calls the data decryption module to decrypt the ciphertext data into plaintext data and finally concatenates the complete fruit and vegetable traceability information and displays it on the client side. The proposed system is shown in Figure 2.

Design of Multi-Chain Storage and Fast Querying

The quality traceability of fruits and vegetables is divided from the supply chain perspective into the planting, storage, logistics, and sales phases. The planting phase includes operations such as sowing, fertilizing, weeding, watering, harvesting, and picking, which require functions such as recording origin information, field environmental variables, operation time, personnel involved, agricultural input information, completion progress, and operation photos. The storage phase must record the fruit and vegetable storage time, out-of-storage time, and cold storage temperature and humidity. The logistics phase needs to record the fruit and vegetable packaging and loading time, the GPS location of the logistics vehicle during the logistics process, and the cold chain temperature and humidity. The sales phase can record the city and distribution location of fruit and vegetable sales.

The data in the quality traceability system is generated in real-time. The data generated by the sensing devices installed in the field meteorological stations, cold storage, and logistics vehicles are based on time series. These data need to be stored in real-time. Storing complete data directly in the blockchain system can cause system blockage. Therefore, the "data storage + hash value on-chain" method can be used. When saving data, the original data is stored in the database, and the hash value of the original data is added to the blockchain. The data is first read from the database, and the hash value is calculated. Then, the hash value of the data is queried from the blockchain, and the calculated hash value is compared with the hash value on the blockchain to verify whether the data has been tampered with.

The "data storage + hash value on-chain" method can meet the requirements when the data volume is small and the response time is not high. However, the fruit and vegetable quality traceability system needs to trace all data throughout the entire lifecycle of fruits and vegetables. When consumers perform one traceability query operation, it involves multiple links, such as farmland environment, planting operations, storage, and logistics, and requires querying multiple records. Obtaining hash values from the blockchain for comparison and verification consumes system resources and time. Especially in the consumer traceability link, the system's response time is extremely high, and system delay will seriously affect user experience. Therefore, a mapping relationship data structure is designed.

Data mapping is the key to improving the efficiency of traceability queries in this model. The mapping relationship is a data structure similar to a tree, reflecting the one-to-many relationship between the product traceability code and the transaction address. Indexing is performed through the mapping relationship when tracing the source, and the query speed is relatively fast. The root node records the product traceability code, each stage's batch number or order number, and the transaction address. The leaf node refers to the transaction ID in the sub-chain, which has uniqueness to prevent data from being tampered with. The leaf node classifies and stores manufacturer information, product information, certificate of conformity, and other information. This process realizes the need for market supervision and quality safety, and the anti-tampering function is achieved through the blockchain timestamp in the process of putting fruit and vegetable data on the chain.

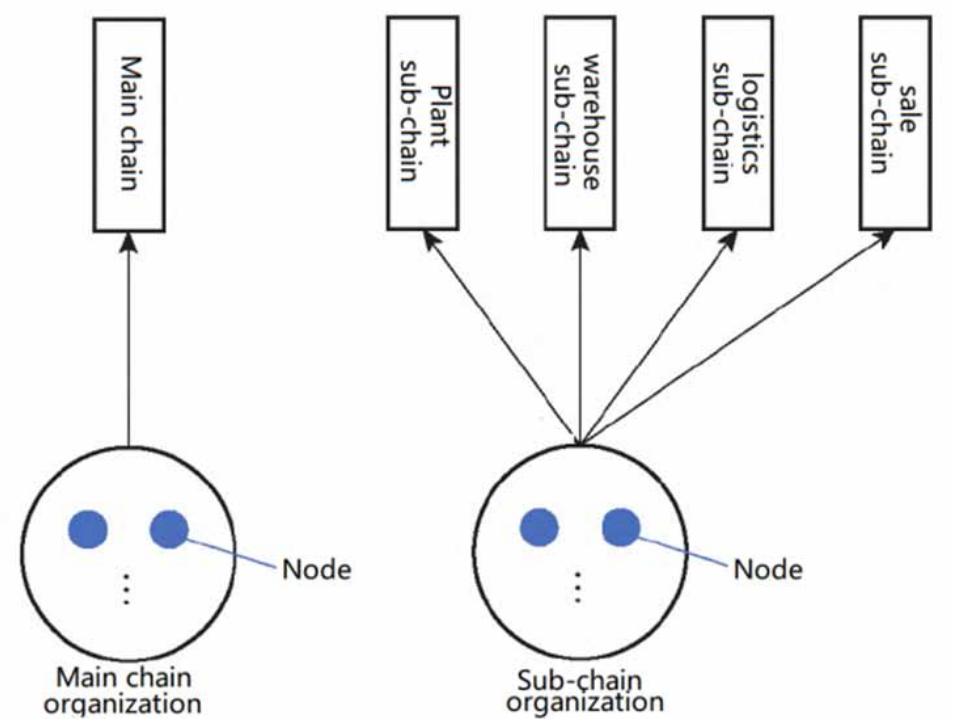
The node architecture is shown in Fig. 4. In a single-chain query, all nodes are part of the same organization and only maintain the same chain. In a multi-chain query, nodes are divided into main- and sub-chain organizations, each containing main- and sub-chain nodes. The main chain organization maintains the main chain, and the sub-chain organization maintains four sub-chains simultaneously.

Design of the Smart Contract

Smart contract design is critical in the fruit and vegetable traceability system implementation. The on-chain contract implements the on-chain functionality, specifically the fruit and vegetable safety judgment function. This function strictly controls the consumption safety of fruit and vegetable products by determining whether the significant harmful data of the fruit and vegetable products is within the limit value. For instance, at the FVSC storage stage, various manufacturers upload traceability information in JavaScript Object Notation (JSON) format to the blockchain ledger by calling the corresponding on-chain contract. The contract then returns the transaction hash address. The specific algorithm for the storage of information on-chain is Algorithm 1.

The mapping relationship is established and written into the main chain through the mapping contract. A new mapping relationship is created when the mapping relationship for a particular fruit and vegetable product has not been established. When the mapping relationship already exists, it is updated. The specific algorithm is as follows: Each link node uploads data to the corresponding sub-chain by calling the smart contract. The smart contract first uploads all the data of this link to the corresponding sub-chain. Then, it establishes a mapping relationship between the traceability code and the sub-chain transaction hash address. If there is already data with the same traceability code in the mapping relationship, the corresponding transaction hash address is updated. The specific mapping contract algorithm is described in Algorithm 2.

Figure 4. Node structure



Algorithm 1. Storage information on-chain

Input: traceability code *BatchCode*, storage temperature *T*, quality measure *Q*
Output: hash address of successful transactions *TxID*, or error cause if failed.
1. Message= Stub. GetState(*BatchCode*) // check if the BatchCode is existed in the Sub-chain
2. **If** *Q* > safety threshold
3. **Return** on-chain failed, and the error cause
4. **If** *T* > storage temperature threshold
5. **Return** on chain failed, and the error cause
6. Message. Encryption()
7. Sub. PutState(message) // encrypted data on-chain
8. **Return** *TxID*

When a user queries the traceability of a fruit and vegetable product, the query contract is used to input the traceability code and retrieve the information on the sub-chain. The specific algorithm for the query contract is described in Algorithm 3.

Improved Kafka Consensus Mechanism

Kafka is a high-throughput, distributed, publish-subscribe messaging system that includes roles such as producers, consumers, and brokers (Tsenos, & Kalogeraki., 2021). It can process hundreds of thousands of messages per second and has excellent fault tolerance (Le Noac et al., 2017). A coordinator role

Algorithm 2. Mapping contract

Input: traceability code *BatchCode*, transaction hash address *TxID*
Output: mapping relationship *Index*
*1. BatchCode*AsBytes: sub. GetState(*BatchCode*) // check if the mapping relation is existed in the main chain
2. Index: = &Index{} // create new mapping relation
3. Func UpdateIndex(args[]string) //update existing mapping relatioin
4. If updateItem = = "plant"{ //update the TxID of each stage
5. Index. plant*TxID* = *TxID*
6. }else if updateItem = = "warehouse"{
7. Index. warehouse*TxID* = *TxID*
8. }else if updateItem = = "logistics"{
9. Index. logistics*TxID* = *TxID*
10. }else if updateItem = = "sale"{
11. Index. sale*TxID* = *TxID*.
12. }
13. stub. PutState (*BatchCode*, *Index*) // mapping relation on-chain of the main chain
14. Return Index

Algorithm 3. Query contract

Input: BatchCode
Output: sub-chain Traceability information
1. Stub. GetState(*BatchCode*)
2. (plant*TxID*, warehouse*TxID*, logistics*TxID*, sale*TxID*) = GetMapping (*BatchCode*)
3. Ciphertext. plant = plantChain. GetState(plant *TxID*)
4. Ciphertext. warehouse = warehouse. GetState(warehouse *TxID*)
5. Ciphertext. logistics = logistics. GetState(logistics *TxID*)
6. Ciphertext. sale = sale. GetState(sale *TxID*)
7. Ciphertext. Decrypt(); // Decryption of the ciphertext
8. **Return** Traceability information

was added to the algorithm framework to address the low consumer message processing efficiency caused by load-balancing operations in Kafka. Each consumer in the cluster subscribes to messages from the server, and each consumer follows the coordinator's instructions to distribute and deploy information transmission. Based on the original information transmission process of Kafka, the producer cluster publishes messages, which are then transmitted to the broker server and stored on the broker under different topics according to specific storage rules.

In Kafka, there are many partitions in a topic, and consumer groups subscribe to different partitions to receive the required information. This article defines the consumer in the Kafka cluster with the longest standard message reception time in all consumer groups as the coordinator. It is responsible for load-balancing work for one or more consumer groups. Each consumer submits its subscription

information to the coordinator, who then assigns initial partitions uniformly. If the coordinator does not receive heartbeat signals from consumers, it identifies the consumers experiencing abnormal situations and triggers load balancing accordingly. In this context, a heartbeat signal refers to a periodic signal sent by consumers to a coordinator to indicate their liveliness and regular activity. The coordinator can monitor their active status by receiving the heartbeat signals from consumers. In other words, the heartbeat signal can be seen as a health check mechanism to ensure that consumers remain active while processing messages.

First, the coordinator detects the consumer group where the abnormal consumer appears and checks the load of the consumers in the group and the corresponding topic. Suppose there are partitions in the topic that have no consumers receiving messages. In this case, messages will accumulate in those partitions due to the inability to receive them, leading to an excessively high partition load. The coordinator obtains the load status of each partition in the topic and records it as $\delta(t)$. If the number of partitions is N, the average load of the partitions is $\Delta^N(t)$, which is the values of $\{\delta_1(t),\delta_2(t),\delta_3(t),...,\delta_i(t),...,\delta_N(t)\}$. If there are partitions with high loads, all partitions in the corresponding topic are sorted according to their load, from high to low, and the sorting result is stored in a data table. The set of partitions is denoted by P, the n-th partition in the data table is denoted by n_n, and all consumers in the consumer group corresponding to the topic are sorted according to their normal message reception time, from long to short. The set of consumers is denoted by C, and the n-th consumer in the consumer group is denoted by m_n. If $n = m$, the partitions and consumers are corresponded one by one:

$$n_n = m_n \tag{1}$$

If $n > m$, then there is a consumer corresponding to y partitions:

$$y = n / m \tag{2}$$

If y is a positive integer, each consumer corresponds to y partitions. If y is not a positive integer, each consumer corresponds to $y+1$ partitions. Regardless of whether y is a positive integer or not, when assigning partitions for the last time, the remaining partitions are assigned to the last consumer.

If there is no partition with the maximum load, i.e., $\delta_i(t) = \Delta^N(t) / n$, it means that all partitions have consumers responsible for receiving messages. However, since the number of consumers in the consumer group does not match the number of partitions, load balancing is triggered, and the number of consumers still needs to be balanced. At this time, z consumers should be added or removed to satisfy the following conditions:

$$k(m \pm z) = n \tag{3}$$

Where k is a positive integer, and then the partitions are reassigned to consumers in a round-robin manner. This approach ensures that the Kafka system will not have any imbalanced message distribution, and the load balancing work will be completed.

EXPERIMENT AND ANALYSIS

The experiment conducted on a PC with 32 GB of RAM, 1 TB of disk space, and a bandwidth of 200 Mb/s. The experimental environment was based on Ubuntu 16.04 and Docker 19.03, and the blockchain architecture used was Hyperledger Fabric 1.4.2. The data for each experiment was the average of 10 tests. The block size was set to 10 MB, the transaction size was 512 KB, and every ten transactions were packaged to generate a block, with a block generation time of 2 seconds. The consensus algorithm was an improved Kafka, with a batch processing information threshold of 16 KB, a maximum message size of 1 MB, and a maximum waiting time for request responses of 30 seconds. The experimental data was obtained from the traceability records of Sunshine Rose Grapes products from an agricultural company located in Jiangsu Province, China.

Data Credibility Verification

After data is stored and uploaded to the blockchain, WebyogSQLyog database client tool was used to log in to the database management system to randomly modify one data record in 50 traceability chains. After the data was modified, the traceability query client was used to query the results of the 50 traceability chains, which all show that the data has been tampered with. The results indicate that the proposed traceability system based on blockchain technology can accurately identify situations where data has been tampered with, thus enhancing data security.

Comparison Results

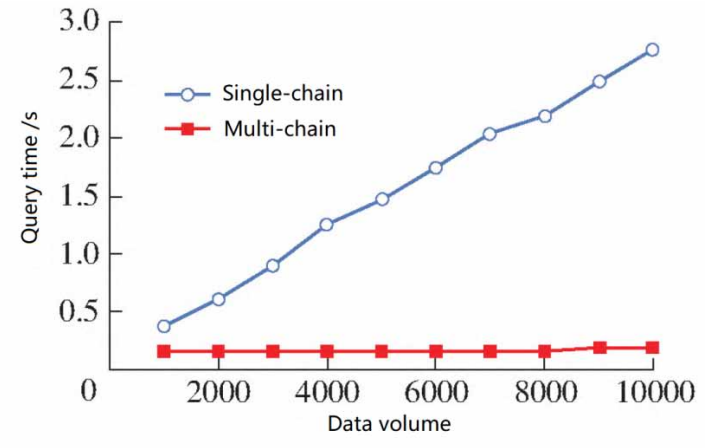
When querying the traceability code of a specific fruit or vegetable in a single-chain query, for the data uploaded by five traceability enterprises, the entire chain needs to be traversed five times according to the traceability code to search for the content. The entire process is divided into two stages in a multi-chain query: the main and sub-chain query. In the main chain query stage, the mapping relationship is traversed and searched on the main chain based on the traceability code. Then, the transaction hash address of the sub-chain is obtained from the mapping relationship. In the sub-chain query stage, the transaction records of the sub-chain are quickly located based on the obtained transaction hash address. Multi-chain queries are faster because the main chain query stage uses traversal queries, but the main chain only stores mapping relationships and has a shorter block length. In the single-chain model, all traceability information is stored on one chain, with a longer block length, so the main chain traversal query is much faster than the single-chain traversal query. The sub-chain query stage is queried based on the transaction hash address in the mapping relationship, which is significantly faster than content traversal queries in a single chain. As shown in Table 2, the query method provided in this paper consists of two parts: the main chain query time and the sub-chain query time. When the data volume increases to 1,000 records, the total query time is 0.157 s, the main chain query time is 0.07 s, and the sub-chain query time is 0.087 s. When the data volume increases to 10,000 records, the query time gradually increases to 0.205 s, with the main chain query time being 0.086 s, and the sub-chain query time being 0.119 s.

As shown in Fig.5, when the data volume increases to 1,000 records, the multi-chain query time is 0.157 s, and the single-chain query time is 0.381 s; when the data volume increases to 10,000 records, the multi-chain query time gradually increases to 0.205 s, and the single-chain query time is 2.785 s. When uploaded data reaches 10,000 records, the query efficiency can be improved by more than 90%.

Table 2. Query time cost for main chain and sub-chain (/s)

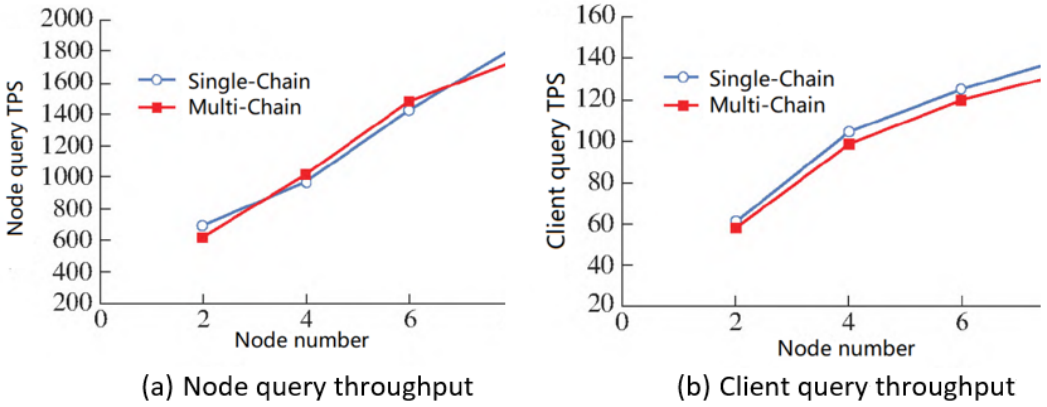
Data volume	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
Main chain	0.070	0.072	0.074	0.075	0.077	0.078	0.080	0.082	0.084	0.086
Sub chain	0.087	0.090	0.094	0.099	0.102	0.106	0.107	0.110	0.112	0.119
Total	0.157	0.162	0.168	0.174	0.179	0.184	0.187	0.192	0.196	0.205

Figure 5. Query time comparison



Network load balancing is used for multi-node testing to select different blockchain nodes for synchronous querying. Node query throughput refers to the throughput of directly initiating transaction queries from the blockchain node, executing smart contracts, and receiving node query results. This process only occurs on the node, so the throughput is relatively large. Client query throughput is the throughput of querying remote blockchain nodes from the client's browser. Transactions must be submitted to the node, and the entire process requires network transmission of information, so the throughput is relatively small. As shown in Fig. 6a, the node query throughput will gradually increase with the number of nodes. When the number of nodes is 2, 4, 6, and 8, the single-chain node query throughputs are 705, 969, 1,398, and 1,799 TPS (Transactions per Second), respectively. The multi-chain node query throughputs are 615, 1,023, 1,477, and 1,709 TPS, respectively, and the difference between the two is not significant. As shown in Fig. 6b, the client query throughput will also gradually increase with the number of nodes. When the number of nodes is 2, 4, 6, and 8, the single-chain client query throughputs are 59, 102, 119, and 138 TPS, respectively. The multi-chain client query throughputs are 55, 101, 119, and 129 TPS, respectively. In terms of client query throughput, single-chain queries are slightly higher than multi-chain queries.

Figure 6. Throughput comparison



As shown in Figure 7, for different situations with 2, 4, 6, and 8 nodes, 1,000 and 3,000 records were stored on the blockchain, respectively. Then, ten queries were simultaneously initiated, and the average query time was taken for each query. The experiment showed that as the number of nodes increased, the query time for both methods decreased, and the multi-chain query time was smaller than the single-chain query time. The test results show that compared with the traditional single-chain storage model, the multi-chain storage model can greatly improve query efficiency when the data volume is large and has a relatively small impact on throughput.

The original Kafka algorithm was compared with the proposed algorithm to verify the performance of the consensus mechanism. When using the original Kafka algorithm to run data on-chain business, it was found that the chain could not be maintained when three machines were shut down out of the six machines, indicating that the original algorithm could not cope well with adverse working environments. The consensus speed of different algorithms was tested in the same blockchain environment, and the test results are shown in Fig.8. When using the system's built-in Kafka consensus algorithm, which is the original algorithm, the performance is consistent with that of the proposed algorithm when no consumers are down. However, after one or two consumer failures, the consensus speed of the original algorithm is slower than that of the proposed algorithm. In an environment with five consumers, the consensus speed of the proposed algorithm is 7% faster than that of the original algorithm. In an environment with four consumers, the proposed algorithm is 8% faster than the original algorithm. This speed improvement is because the improved algorithm can quickly deploy and adjust by the coordinator when consumers are down, and the corresponding relationship between consumers and partitions is more reasonable without other consumers making too many attempts at load balancing. Currently, the consensus speed is about 4.5 seconds. In an environment with three consumers, the original algorithm cannot reach a consensus, but the proposed algorithm can reach a consensus speed of about 4.7 seconds. In environments with two and one consumer, neither the improved algorithm nor the original algorithm can reach consensus, and the consensus speed is 0. We can observe that the stability of the original algorithm is not as good as the proposed consensus algorithm in this paper. When using the proposed consensus algorithm in this paper, the consensus speed of the system has been improved, and the security has been stabilized, which can effectively cope with the adverse working conditions of multiple consumer failures.

Regarding algorithm accuracy, after careful comparison and inspection, the information is consistent before and after the consensus, and no errors occur. Therefore, the algorithm accuracy of the two algorithms is consistent. As the proposed algorithm can avoid message accumulation, it can still reach consensus in an environment with three consumers. Hence, we assess that the proposed algorithm is superior to the original algorithm in terms of real-time performance.

Figure 7. Average query time comparison

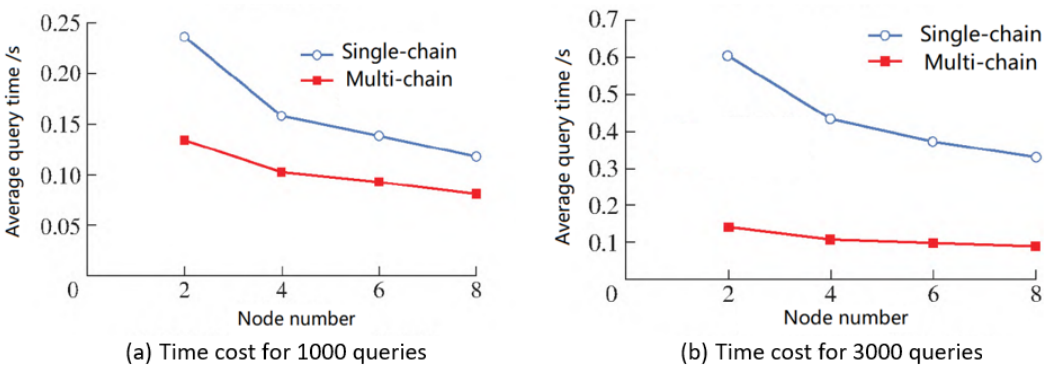
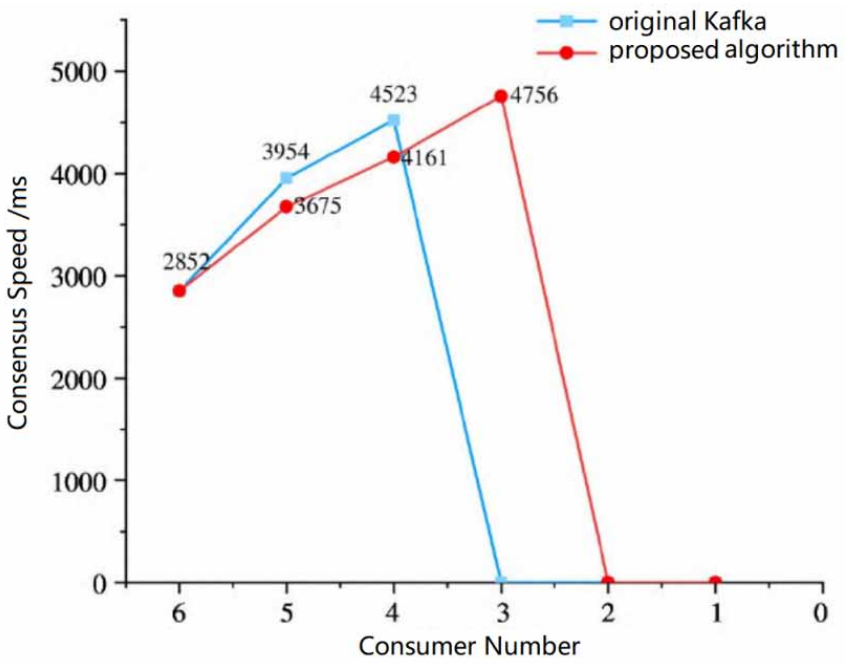


Figure 8. Test results of different consensus algorithms



DISCUSSIONS

As proposed in this study, implementing a multi-chain blockchain-based traceability system for the FVSC demonstrates significant improvements in the efficiency and security of fruit and vegetable quality traceability. Within the multi-chain blockchain framework, data is distributed and encrypted, making it more resistant to tampering or destruction, thereby enhancing the security and reliability of the data. In traditional single-chain models, query efficiency may significantly decrease as the volume of data increases. However, the proposed multi-chain blockchain model exhibits notably superior query efficiency compared to the traditional model, particularly when the data records exceed 10,000. This substantial enhancement greatly improves the efficiency of traceability efforts. The model facilitates comprehensive on-chain data storage, further enhancing the security of traceability data. Additionally, through an improved Kafka load-balancing algorithm, the system achieves consensus. It completes on-chain transactions even if half of the nodes are offline, enhancing system stability.

The proposed approach establishes closer connections between various links in the FVSC and the traceability model, reducing the isolated effects of each link in the supply chain. Each link can easily upload data to the blockchain, ensuring consumers can access complete, authentic, and comprehensive traceability information for fruits and vegetables. Moreover, through a simulation implementation based on the Hyperledger Fabric framework, the reliability and applicability of the proposed model have been verified. This secure traceability model assures consumers of the authenticity of fruit and vegetable traceability information and offers regulators a convenient and effective monitoring framework.

However, the specific implementation of fruit and vegetable blockchain systems also faces several challenges. These include:

Complexity and resource intensiveness of system integration: Integrating a multi-chain blockchain system into existing supply chain systems can be highly complex and resource-intensive. The fruit and vegetable industry involves multiple stages, including cultivation, harvesting, processing,

transportation, and more, each generates significant amounts of data. Developing and implementing data collection, transformation, and integration systems require substantial resources. Integrating blockchain-based FVSC traceability systems with existing information systems and technologies, such as farm management systems and supply chain management systems, may require technical adjustments and interface development to ensure seamless data transmission and interoperability. Additionally, blockchain traceability systems require ongoing maintenance and updates to ensure their functionality and security, necessitating dedicated technical teams to monitor system operations, address issues, and promptly fix vulnerabilities.

Time and effort required for widespread adoption: Promoting the widespread adoption of proposed FVSC traceability systems requires the support and cooperation of various industry stakeholders, including farmers, producers, different stages of the supply chain, and retailers. However, each stakeholder may have different interests and needs, necessitating communication, coordination, and fostering an understanding of the system's benefits to encourage adoption. This process requires investing time and resources to build trust, foster cooperation, and address potential conflicts of interest. Additionally, promoting the widespread adoption of FVSC traceability systems requires striking a balance between costs and benefits. While the system may offer numerous potential benefits, such as improved supply chain transparency, reduced food safety risks, and improved consumer trust, ensuring the cost-effectiveness of implementation and maintenance is essential. Demonstrating the economic benefits and ensuring the reasonable cost of the system are critical factors in driving widespread adoption.

Regulatory and legal challenges: Utilizing blockchain technology for FVSC traceability may encounter regulatory and legal challenges. FVSC traceability systems collect and store significant amounts of data, including sensitive information such as personal identities and transactional details. Compliance with relevant data privacy and protection regulations, such as the European General Data Protection Regulation (GDPR), is essential when utilizing this data. Furthermore, when dealing with intellectual property rights such as branding, trademarks, and patents related to agricultural products, it is crucial to ensure that using such intellectual property rights within the system does not infringe upon the rights of others. If the FVSC traceability system involves multiple stakeholders, particularly those involved in different stages such as production, processing, and transportation, compliance with antitrust regulations is necessary to prevent market monopolies or unfair competition. In the event of disputes during the implementation, appropriate dispute resolution mechanisms must be established, potentially involving arbitration, mediation, or litigation, with corresponding solutions tailored to local legal regulations.

CONCLUSION

This paper proposed a fruit and vegetable quality traceability system based on a multi-chain blockchain. The use of blockchain-based data hashing and verification methods for secondary chaining has improved the trustworthiness of the data storage in the fruit and vegetable quality traceability system, ensuring the authenticity of the data among multiple participating entities. Experimental results demonstrate that the proposed system solves the problem of long query times for consumers when storing fruit and vegetable quality traceability data on the blockchain, and the query time remains largely constant with increasing data volume, thereby improving the system's scalability. Additionally, the improved Kafka algorithm's consensus speed is faster than the original algorithm, and it can adapt to more adverse node failure situations, improving the system's stability.

FUNDING STATEMENT

This work was supported by Research on key technologies of authentication mechanism based on enhanced blockchain framework (No. 232102210197). Research on consistency checking algorithm of spatial direction relation in spatial database (No.232102210085).

REFERENCES

Aung, M. M., & Chang, Y. S. (2014). Traceability in a food supply chain: Safety and quality perspectives. *Food Control*, *39*, 172–184. doi:10.1016/j.foodcont.2013.11.007

Balaji, M., & Arshinder, K. (2016). Modeling the causes of food wastage in Indian perishable food supply chain. *Resources, Conservation and Recycling*, *114*, 153–167. doi:10.1016/j.resconrec.2016.07.016

Brice, J., Donaldson, A., & Midgley, J. (2020). Strategic ignorance and crises of trust: Un-anticipating futures and governing food supply chains in the shadow of Horsegate. *Economy and Society*, *49*(4), 619–641. doi:10.1080/03085147.2020.1781387

Budish, E. B. (2022). *The economic limits of Bitcoin and anonymous, decentralized trust on the blockchain* (Working Paper, 83). University of Chicago, Becker Friedman Institute for Economics. 10.2139/ssrn.4148014

Caro, M. P., Ali, M. S., Vecchio, M, & Giaffreda, R. (2018). Blockchain-based traceability in agri-food supply chain management: A practical implementation. *2018 IoT Vertical and Topical Summit on Agriculture-Tuscany (IOT Tuscany)*, (pp. 1-4). IEEE. 10.1109/IOT-TUSCANY.2018.8373021

Chen, K., Wang, X., & Song, H. (2015). Food safety regulatory systems in Europe and China: A study of how co-regulation can improve regulatory effectiveness. *Journal of Integrative Agriculture*, *14*(11), 2203–2217. doi:10.1016/S2095-3119(15)61113-3

Dabbene, F., Gay, P., & Tortia, C. (2014). Traceability issues in food supply chain management: A review. *Biosystems Engineering*, *120*, 65–80. doi:10.1016/j.biosystemseng.2013.09.006

Feng, H., Wang, X., Duan, Y., Zhang, J., & Zhang, X. (2020). Applying blockchain technology to improve agri-food traceability: A review of development methods, benefits and challenges. *Journal of Cleaner Production*, *260*, 121031. doi:10.1016/j.jclepro.2020.121031

Francois, G., Fabrice, V., & Didier, M. (2020). Traceability of fruits and vegetables. *Phytochemistry*, *173*, 112291. doi:10.1016/j.phytochem.2020.112291 PMID:32106013

Galvez, J. F., Mejuto, J. C., & Simal-Gandara, J. (2018). Future challenges on the use of blockchain for food traceability analysis. *Trends in Analytical Chemistry*, *107*, 222–232. doi:10.1016/j.trac.2018.08.011

Guo, H., & Yu, X. (2022). A survey on blockchain technology and its security. *Blockchain: Research and Applications*, *3*(2), 100067. doi:10.1016/j.bcra.2022.100067

Hao, Z., Mao, D., Zhang, B., Zuo, M., & Zhao, Z. (2020). A novel visual analysis method of food safety risk traceability based on blockchain. *International Journal of Environmental Research and Public Health*, *17*(7), 2300. doi:10.3390/ijerph17072300 PMID:32235378

Hua, X. F., Wang, W. Q., Tian, Y. C., & Xing, K. Z. (2013). Design of aquatic product traceability system based on anti-collision RFID techniques. *Advanced Materials Research*, *765*, 2017–2020. . doi:10.4028/www.scientific.net/AMR.765-767.2017

Jum'a, L., Zimon, D., & Madzik, P. (2023). Impact of big data technological and personal capabilities on sustainable performance on Jordanian manufacturing companies: The mediating role of innovation. *Journal of Enterprise Information Management*. doi:10.1108/JEIM-09-2022-0323

Kalodner, H., Möser, M., Lee, K., Goldfeder, S., Plattner, M., Chator, A., & Narayanan, A. (2020). Blocksci: Design and applications of a blockchain analysis platform. *29th USENIX Security Symposium*, (pp. 2721-2738). ACM. https://dl.acm.org/doi/10.5555/3489212.3489365

Köhler, S., & Pizzol, M. (2020). Technology assessment of blockchain-based technologies in the food supply chain. *Journal of Cleaner Production*, *269*, 122193. doi:10.1016/j.jclepro.2020.122193

Le Noac, H. P., Costan, A., & Bougé, L. (2017). A performance evaluation of Apache Kafka in support of big data. *IEEE International Conference on Big Data (Big Data),* (pp. 4803-4806). IEEE. doi:10.1109/BigData.2017.8258548

Leng, K., Bi, Y., Jing, L., Fu, H. C., & Van Nieuwenhuyse, I. (2018). Research on agricultural supply chain system with double chain architecture based on blockchain technology. *Future Generation Computer Systems*, *86*, 641–649. doi:10.1016/j.future.2018.04.061

Li, X., Wu, L., Zhao, R., Lu, W., & Xue, F. (2021). Two-layer adaptive blockchain-based supervision model for off-site modular housing production. *Computers in Industry*, *128*, 103437. doi:10.1016/j.compind.2021.103437

Lin, Q., Wang, H., Pei, X., & Wang, J. (2019). Food safety traceability system based on blockchain and EPCIS. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 20698–20707. doi:10.1109/ACCESS.2019.2897792

Mao, D., Wang, F., Hao, Z., & Li, H. (2018). Credit evaluation system based on blockchain for multiple stakeholders in the food supply chain. *International Journal of Environmental Research and Public Health*, *15*(8), 1627. doi:10.3390/ijerph15081627 PMID:30071695

Monrat, A. A., Schelén, O., & Andersson, K. (2019). A survey of blockchain from the perspectives of applications, challenges, and opportunities. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 117134–117151. doi:10.1109/ACCESS.2019.2936094

Pearson, S., May, D., Leontidis, G., Swainson, M., Brewer, S., Bidaut, L., Frey, J. G., Parr, G., Maull, R., & Zisman, A. (2019). Are distributed ledger technologies the panacea for food traceability? *Global Food Security*, *20*, 145–149. doi:10.1016/j.gfs.2019.02.002

Ratta, P., Kaur, A., Sharma, S., Shabaz, M., & Dhiman, G. (2021). Application of blockchain and internet of things in healthcare and medical sector: Applications, challenges, and future perspectives. *Journal of Food Quality*, *2021*, 1–20. doi:10.1155/2021/7608296

Rogerson, M., & Parry, G. C. (2020). Blockchain: Case studies in food supply chain visibility. *Supply Chain Management*, *25*(5), 601–614. doi:10.1108/SCM-08-2019-0300

Ruan, P., Dinh, T. T. A., Loghin, D., Zhang, M., Chen, G., Lin, Q., & Ooi, B. C. (2021). Blockchains vs. distributed databases: Dichotomy and fusion. *Proceedings of the 2021 International Conference on Management of Data*, (pp. 1504-1517). IEEE. doi:10.1145/3448016.3452789

Shao, W., Jia, C., Xu, Y., Qiu, K., Gao, Y., & He, Y. (2020). Attrichain: Decentralized traceable anonymous identities in privacy-preserving permissioned blockchain. *Computers & Security*, *99*, 102069. doi:10.1016/j.cose.2020.102069

Truong, N., Lee, G. M., Sun, K., Guitton, F., & Guo, Y. (2021). A blockchain-based trust system for decentralised applications: When trustless needs trust. *Future Generation Computer Systems*, *124*, 68–79. doi:10.1016/j.future.2021.05.025

Tsenos, M., & Kalogeraki, V. (2021). Dynamic rate control for topic-based pub/sub systems. *22nd IEEE International Conference on Mobile Data Management (MDM)*, (pp. 272-273). IEEE. doi:10.1109/MDM52706.2021.00057

Wu, H., Zhong, W., Zhong, B., Li, H., Guo, J., & Mehmood, I. (2023). Barrier identification, analysis and solutions of blockchain adoption in construction: A fuzzy DEMATEL and TOE integrated method. *Engineering, Construction, and Architectural Management*. doi:10.1108/ECAM-02-2023-0168

Yue, K., Zhang, Y., Chen, Y., Li, Y., Zhao, L., Rong, C., & Chen, L. (2021). A survey of decentralizing applications via blockchain: The 5g and beyond perspective. *IEEE Communications Surveys and Tutorials*, *23*(4), 2191–2217. doi:10.1109/COMST.2021.3115797

Zhang, X., Sun, Y., & Sun, Y. (2022). Research on cold chain logistics traceability system of fresh agricultural products based on blockchain. *Computational Intelligence and Neuroscience*, *2022*, 1–13. doi:10.1155/2022/1957957 PMID:35154298

Zhang, Y., Fu, Z., Xiao, X., Zhang, X., & Li, D. (2017). MW-MTM: A mobile wireless monitoring and traceability management system for water-free live transport of aquatic products. *Journal of Food Process Engineering*, *40*(3), e12495. doi:10.1111/jfpe.12495

Feng Xue, Associate Professor. Master's degree. Graduated from Liaoning Petrochemical University in 2010. Worked in Zhengzhou University of Economics and Business. His research interests include Algorithms, Computer applications.

Fangju Li, Professor, Master's degree, Graduated from Information Engineering University in 2000, Worked in Zhengzhou University of Economics and Business. Her research interests include Computer applications.