applied sciences

MDPI

Article

Design of Multi-Chain Traceability Model for Pepper Products Based on Traceability Code

Wenxuan Jin, Mingjun Zheng and Pingzeng Liu *

College of Information Science and Engineering, Shandong Agricultural University, Taian 271018, China; jwx19970224@163.com (W.J.); zmj121493567@163.com (M.Z.)
* Correspondence: pzliu@sdau.edu.cn; Tel.: +86-13854853118

Abstract: In the specific application scenario of pepper product supply chain traceability, with the advancement of pepper product production, the expansion of links, and the increase of nodes, the quantity of data will become more and more enormous. The single-chain model is less efficient for querying if the data are all stored into the same blockchain. In order to improve the efficiency of blockchain data querying, this paper proposes a traceability model with one main chain and multiple side chain structures, which separate the uplinked data from each link and use multi-chain transactions to improve the efficiency of data queries. This model builds an indexing mechanism with a product traceability code, using one main chain and multiple side chains. The main and side chains form a one-to-many mapping relationship, storing the mapping relationship between the traceability code and the transaction address of the side chain traceability information in the main chain. This enables information to travel through the main chain traversal query based on the mapping relationship and then query the direct index out of the side chain, to achieve fast traceability query and improve the efficiency of querying.

Keywords: pepper products; tracing source code; mapping relationship; multi-chain traceability model; query efficiency

Citation: Jin, W.; Zheng, M.; Liu, P. Design of Multi-Chain Traceability Model for Pepper Products Based on Traceability Code. *Appl. Sci.* **2024**, *14*, 3809. https://doi.org/10.3390/app14093809

Academic Editors: Paweł Kiełbasa, Anna Miernik and Akinniyi Akinsunmade

Received: 7 March 2024
Revised: 17 April 2024
Accepted: 18 April 2024
Published: 29 April 2024

1. Introduction

Pepper is a member of the rue family in the genus of small perennial deciduous trees. The fruit skin has small prominent oil spots resembling flowers, hence the name "pepper" [1,2]. Pepper is an edible seasoning used as a spice and medicinal herb and therefore is a multi-purpose economic tree species in China with a long history of cultivation. Its market demand is very large! A variety of deep-finished spices produced using peppercorns also sell well in the marketplace. Pepper oil, pepper powder, pepper sprouts, and other pepper products are popular among consumers. In recent years, the flourishing of spicy hot pot, skewers, and other catering has also caused a surge in demand for fresh peppercorns supplied by large spicy seasoning processing enterprises [3]. Pepper quality and safety issues include pesticide residues, excessive heavy metals, water adulteration, and counterfeiting [4–7]. Therefore, we should monitor the safety of peppercorn products for hazards, establish a traceability system for the quality and safety of peppercorn products, identify problems as early as possible, and record and deal with them, so as to prevent the occurrence of food safety problems [8,9].

In the pepper product industry chain, there are many participants, including growers, planting enterprises, processing manufacturers, logistics enterprises, distributors and end users, which together constitute the key roles in the production and circulation process of pepper products. Blockchain is a distributed data storage technology designed to be decentralised, untamperable, and traceable. The rapid development of blockchain technology has boosted its applications in various fields, especially in the supply chain management of agricultural products, in which it plays a crucial role [10,11]. The application of blockchain technology can provide the pepper product supply chain with more powerful data management capabilities and data traceability and promote cross-organisational collaboration to enhance enterprise competitiveness.

However, with the dramatic increase in the quantity of data in the pepper product supply chain, this data-driven demand has led to an increasing reliance on querying agricultural traceability data by the participants and higher requirements for querying efficiency [12,13]. In a blockchain system, each node has to store a complete copy of the blockchain. When a node needs to query a transaction message, a traversal query will be performed in the local copy of the blockchain [14]. However, due to the increasing number of nodes and the total quantity of data, the data query in the blockchain will become more and more sluggish. In order to improve the query efficiency, a multi-chain traceability model indexed by pepper traceability code is proposed.

2. Current Status of Blockchain Traceability Storage Research

Blockchain technology effectively solves the problems of traditional traceability system, such as the ease of tampering with information, and provides an effective solution to solve the problem of commodity information traceability [15], but due to the characteristics of traceability data generated in the production process of agricultural products, such as the large quantity of data, if the traceability data are directly stored in the blockchain, the blockchain network data storage pressure will be very large. For the storage efficiency problem in the traceability system based on blockchain technology, many scholars also have their own views and solutions at the theoretical and technical levels.

Agarwal V et al. believe that the current blockchain system is not able to achieve the storage and scaling of internet of things (IoT) data with huge quantities of data while maintaining speed as well as time efficiency. For this reason, a combination of sidechain and offline data storage is used to alleviate the scalability problem of blockchain [16]. Chen J and other scholars of data traceability scenarios with a large quantity of data have proposed using data segmentation to store and reduce the memory occupation of data on the chain. This scheme categorizes the data according to the importance of the traceability data, the important data are directly stored in the chain, the non-important data are stored in the off-chain database, the off-chain data are executed with a hash operation, and the hash value obtained is stored in the chain together with the important data [17]. The on-chain and off-chain cooperative storage model has likewise been studied and applied by many scholars.

There are also many studies focusing on building more efficient data index structures, at home and abroad. Xing et al. optimised the query method based on the chain of account transaction trajectories and propose a subchain-based index structure of the account transaction chain, which changes the original block-by-block lookup mode into a subchain lookup mode to shorten the query path [18]. Guo et al. designed a double-layer index structure for optimising traceability query for food supply chain traceability scenarios, which achieves the fast querying of data by locating blocks with external indexes and transactions with internal indexes [19]. Zhang Hong et al. constructed a secondary indexing mechanism which enables users to quickly and accurately retrieve transaction information through account information by establishing a secondary index on the hash values of blocks and transactions. When the block data are persistently stored, the current master node invokes a contract to establish a secondary index and uses the smart contract to construct the access control of other nodes to the secondary index structure, which improves the query efficiency of the system [20]. Ruan et al. proposed LineageChain, which uses snapshots and creates a new index structure to improve the efficiency of queries on the chain [21].

3. Model Design Ideas—One Master Chain and Multiple Side Chain Structures

Existing blockchain technology requires that each node in the single-chain architecture must save all the information of the whole network, which results in bottlenecks in terms of capacity, privacy, and isolation, and therefore, constrained by the performance limit of a single node in the network, single-chain architecture cannot easily satisfy the performance and capacity requirements of the federated chain environment [22]. When the general blockchain traceability model is uploaded using a single-chain structure, the data of different links need to be uploaded separately, and the uploaded data of each link will not exist continuously on the block, which leads to the query needing to sequentially traverse the entire chain in order to find all of the traceability data, and with the increase in the quantity of data, the speed of sequential traversal will increase linearly, which makes the querying speed become slower [23].

In the specific scenario of pepper product supply chain traceability, if the data are all stored on the same blockchain, with the advancement of pepper product production and the expansion of the links, the data volume will become huger and huger. At the same time, the query efficiency of the single chain model is low. In order to improve the blockchain data query efficiency, this paper proposes a traceability model with one main chain and multiple side chain structures. The uplinked data of each link is separated, and multi-chain transactions are utilized to improve the efficiency of data querying. First, a main chain is constructed, which stores the product traceability code and the transaction address of each link, and then a side chain is constructed for each link of pepper products, which stores the traceability information of each link. When querying is performed, the product traceability code is used as an index to query the transaction address of the traceability information of each link in the main chain, and the information content stored in the side chain is directly queried through the transaction address, thus accelerating querying efficiency.

4. Construction of Multi-Chain Traceability Model for Pepper Products

4.1. Pepper Multi-Chain Traceability Model

Aimed at the low query efficiency of the single chain model, a multi-chain traceability model of pepper products is constructed to improve blockchain data query efficiency, as shown in Figure 1. First, a main chain is constructed, which stores the product traceability code and the transaction address of each link, and then a side chain is constructed for each link of pepper products, and the side chain stores the traceability information of each link. This model stores the traceability data of different links in the corresponding link side chain and builds a main chain to store the traceability code of pepper products and the transaction address of traceability information, which not only guarantees the tamper-proofness, credibility, and security of traceability data but also ensures the efficiency of traceability querying.

The blockchain is used to record the traceability information of each link, and the blockchain timestamp is used to record the chronological order so as to realize the uploading of all the records of peppercorn products from planting to consumers' hands, and the consumers can query all of the whole peppercorn supply chain information through its traceability code. In the process of peppercorn data uploading, the plaintext data are encrypted by encryption algorithms and then the ciphertext data are uploaded to protect data privacy, and the blockchain is stored in a multi-chain structure to optimize the efficiency of data query.

After information is collected, the traceability information is divided into public traceability data and private traceability data through the on-chain contract. The private traceability data are encrypted by RSA algorithm, and then the ciphertext data are uplinked to the corresponding side chain, while the public traceability data are directly uplinked after the quality and safety standard monitoring smart contract. Through the side chain uplinking contract, the hash address of the transaction and the link traceability code of the link are returned to the link node, and the node establishes the mapping relationship between the traceability code and each transaction by calling up the mapping contract and then saves the mapping relationship on the main chain.

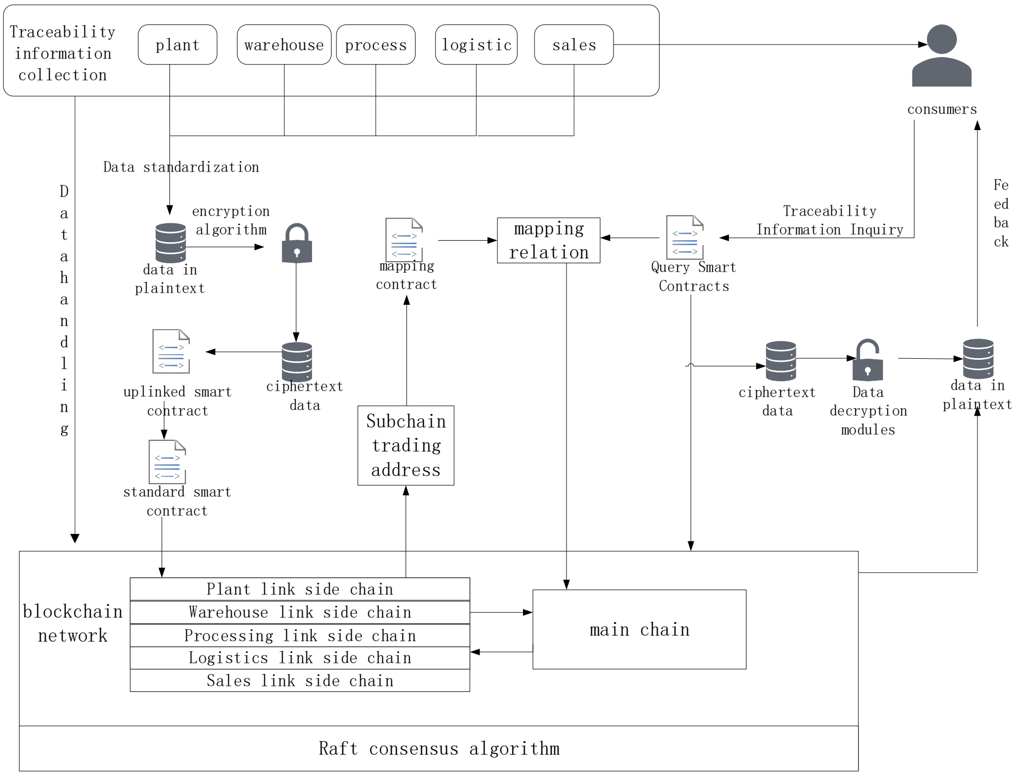

Figure 1. Multi-chain traceability model for pepper products.

In the process of data traceability, consumers query the traceability information through the traceability code in the peppercorn traceability system; the client node verifies the traceability code, transmits it to the blockchain network, and calls up the mapping contract to query the main chain to obtain the mapping relationship. Then the mapping relationship is used to obtain the transaction hash address on each side chain while the nodes use the transaction address to index the query on each side chain and obtain the traceability data of each side chain through the transaction address, and finally, the traceability information is integrated into each side chain to create the complete traceability information of the whole industry chain, which is presented in front of the consumers.

This model uses one main chain and multiple side chains, the main chain and side chains form a one-to-many mapping relationship, and the mapping relationship between the traceability code and the transaction address of the side chain traceability information is stored in the main chain. The mapping relationship is queried by traversing the main chain, and then the side chain traceability information is queried by direct indexing to realize fast traceability, which improves the efficiency of traceability querying.

4.2. Storage Schema Design for Multi-Chain Traceability Model

The mapping relationship reflects the one-to-many relationship between the product traceability code and the transaction address. The traceability is only indexed according to the mapping relationship, so the query speed is much faster. The product traceability code and transaction address are recorded by the root node, and the leaf nodes are addressed by the transaction address, which is unique on the side chain to avoid tampering with the data. The mapping relationship and key are stored on the main chain; when a transaction is passed on the side chain, the main chain first checks whether there is already a map for this traceability code. If there is no record on the main chain, the mapping relationship of the traceability code is created, which establishes a mapping relationship between the traceability code in the main chain in the form of an object and the address of the transaction in the side chain, and if the mapping relationship already exists in the main chain, the mapping relationship is updated. The main chain and side chains correspond to one-to-many relationship, in which one main chain corresponds to several side chains. The transaction addresses in the side chains are unique, which prevents the data from being tampered with during the storage process. The schematic diagram of the mapping relationship is shown in Figure 2.

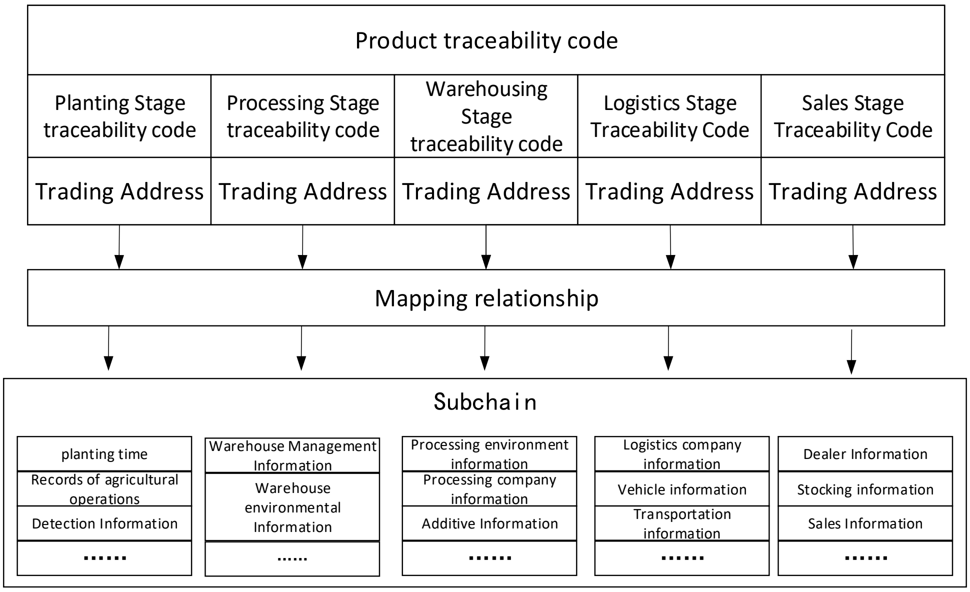

Figure 2. Mapping relationship diagram.

The final traceability code of the product is written into the blockchain to form a mapping relationship with the transaction address of the side chain storing the traceability information, and the traceability code is used as an index and unique identification. A mapping relationship is established between the transaction address of each link when it is uplinked into the side chain and stored in the main chain, and the block body uses key-value pairs to store the relevant information.

A side chain is created for each of these nodes: planting, processing, storage, logistics and sales. The traceability information of each link is stored in the side chain of each node of the supply chain and returns the transaction address of its exchange in the transaction address of each link traceability code. The transaction address in the format of key-value pairs is stored in the main chain, and if it is necessary to update the information of peppercorn products, it is necessary to write the transaction address of the traceability information into the corresponding structure to update its transaction address. The consumer can enter the traceability code of the liquid product, exchange codes by entering the traceability code in order to obtain the traceability code of each link, obtain the transaction address, and query the traceability information on the corresponding blockchain with the transaction address.

When uploading, the vendors of each link can upload the data into the side chain respectively and store the transaction address of the side chain and the link traceability code of each link through the main chain. When querying, the transaction address of the pepper product in each side chain is queried in the main chain through the link traceability code, and then the traceability data of each link is queried in the side chain through the transaction address, and the querying of each side chain can be carried out synchronously.

First of all, the traceability data should be uplinked, and the data of different traceability links should be uplinked by generating smart contracts for the corresponding channels. After the uplinking of each link, the main chain is updated through the exchange node, and when the data of other traceability links are uplinked, the key-value pairs of the main chain object data are updated by finding the corresponding traceability data objects through the batch number. Taking the uplinking process of planting side chain as an example, each stage is described in detail as follows:

1. Traceability information is stored on the planting sidechain block:

After collecting the information, the manufacturer submits the traceability information regarding planting, processing, storage, and transportation to the respective side chain through the uplink function in the smart contract. After receiving the request, the planting side chain will generate a transaction and store the transaction content in the current block and generate the transaction address.

2. Sidechain update status data:

The side chain updates the status data according to the incoming transaction content, and the status data of the planting side chain will add a piece of traceability information regarding the planting link of the pepper product, so that the status data can be quickly queried through the transaction address without traversing all the data.

3. Coded exchange smart contract:

The uplink function of the side chain will return the transaction address, the traceability code of each link, and the peppercorn product traceability code to the public node of the link and then store them in the main chain. Thus, the whole peppercorn product traceability process is controlled through the main chain.

4. Main chain query to see if a record already exists:

Only the traceability code and the transaction address of each link are stored in the main chain, because a traceability code will correspond to the transaction address of four side chains, so after the transaction information is passed from the side chains, the main chain is first queried to see if the information record of the traceability code already exists.

5. New/updated records:

If the record already exists in the main chain, then the record is updated according to the update record; the update record only updates the corresponding transaction address of the link, and the link does not have the authority to update the transaction address of other links. If the mapping relationship does not exist in the main chain, a new mapping relationship is created between the traceability code and the transaction address of the traceability information of the side chain, taking the traceability code as the primary key and setting up the transaction address of the link. This completes the traceability information uplinking phase.

6. Creating mapping relationships;

A mapping relationship is established between the traceability code in the form of an object in the main chain and the transaction address in the side chain, and the main chain and the side chain correspond to each other in a one-to-many relationship, with one main chain corresponding to multiple side chains.

4.3. Fast Query Process for Multi-Chain Traceability Model

When querying on the blockchain, the query time increases linearly with the quantity of data. In this paper, we propose a multi-chain query process that can not only guarantee data security, but also improve the query efficiency of traceability, as the main chain only stores the traceability code and mapping relationship data while the main pepper industry chain traceability information is stored in the side chain, and the query efficiency is improved by establishing the mapping relationship with the side-chain transactions and utilizing index querying, and this query mode will shorten the query time of the traceability data when the data volume is large.

The traceability data query process of the multi-chain model is shown in Figure 3. When the system receives the query request for traceability information, it first traverses the main chain to query the mapping relationship, conducts a quick query on the information of each stage through the mapping relationship, and then splices these together to form the traceability information of the whole supply chain and transmits it to the main chain, which transmits the query result to the client through the smart contract, and the front-end page reads the data to the consumer. The front-end page reads the data and displays the traceability information to consumers.

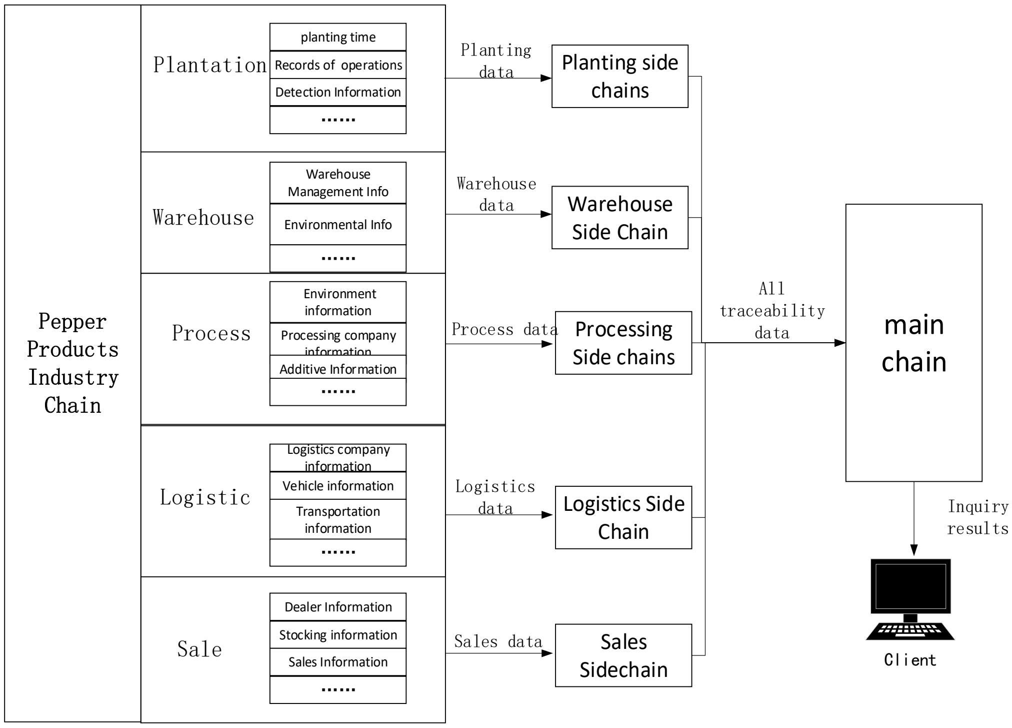

Figure 3. Tracing data query process for multi-chain models.

The traceability code is input into the query box in the traceability system, and the query module conducts sequential querying on the main chain by calling up the smart contract and returns the acquired traceability code and transaction address of each link of pepper products to the public node. According to the transaction address of each link obtained, the transaction address query is performed on the side chain of each link, the traceability data of the corresponding link is obtained and returned to the client, and the client finally splices the data to generate complete traceability information.

4.4. Smart Contract Design under Multi-Chain Traceability Modeling

Smart contracts in the multi-chain traceability model mainly include data uplinking smart contracts; quality and safety standard monitoring smart contracts; mapping smart contracts; and query smart contracts. The smart contract specifies the format of the data, the process and rules for executing transactions, etc., and these rules together form the business model for transactions conducted by both parties.

When each node in each link carries out data uplinking, the uplinking smart contract is invoked to first uplink all the data of the link to the side chain of the link, and then uplink the link traceability code and its corresponding transaction address of each link to the main chain.

4.4.1. Uplinked Smart Contract

Data uplinking is realized through the uplinking contract, and the uplinking smart contract mainly has the following functions: 1. Judging the heavy metal content in the traceability information of pepper products to be uplinked by using the smart contract, and if it does not meet the national standard, the traceability information will fail to be uplinked, and the system will return the reasons for the failure of the uplinking, limiting value data from "GB/T 30391-2013 Peppercorn Standard" [24]. 2. Judging whether the traceability data are private traceability data; if they are privately traceable, then the data are entered into an encryption module, and the public traceability data will be directly uploaded to the chain after they meet the conditions of the smart contract execution. 3. Judging data encryption by encrypting private traceability data through the RSA algorithm to ensure the data security of the private data. Taking the processing of traceability data as an example, the pseudo-code of the uplinking contract is shown in Algorithm 1.

Algorithm 1. Up chain Smart contract pseudo code.

```
Input: Pepper for total arsenic X, lead Y, cadmium Z, total mercury G
Output: successful transaction hash address TxID, failure to return error reason
message = Stub. GetState(TraceabilityCode) //Checking for TraceabilityCode records in the sidechain
when (ID = 1)//At this time the product is fresh or frozen peppercorns
if (X > 0.07 or Y > 0.42 or Z > 0.11 or G > 0.01)
return “Excessive heavy metals”//fail to upload
when (ID = 2)//At this time the product is dried peppercorns or powdered peppercorns
if (X > 0.30 or Y > 1.86 or Z > 0.50 or G > 0.01)
return “Excessive heavy metals”//fail to upload
message.encryption();//Calling the encryption module to encrypt
Stub.PutState (message) //uplink ciphertext data
return TxID//Upload successfully, return transaction hash address
```

4.4.2. Mapping Smart Contract

The mapping relationship is established through the mapping contract and written into the main chain. If the peppercorn product has not yet established a mapping relationship, a new mapping relationship is created, and the mapping relationship is updated in the case of an existing mapping relationship. The specific process is that each link node is uploaded by calling up the smart contract, and the smart contract first uploads all the data of the link to the corresponding side chain, and then establishes the mapping relationship between the traceability code and the side chain transaction hash address, and if there is already the data of the traceability code in the mapping relationship, the corresponding transaction hash address is updated. The mapping contract pseudo-code is shown in Algorithm 2.

Algorithm 2. Mapping Smart contract pseudo code.

```
Input: TraceabilityCode, TxID, Update Various Segments Breed Procees Pransport Storge Sales
Output: return Mapping
message = Stub.GetState(TraceabilityCode) //Checking for TraceabilityCode records in the sidechain
Index := &Index{} //If you don’t have one, create a new mapping relationship
func UpdateIndex (args []string)//If yes, update the mapping relationship
if updateItem == “Plant” {//Update transaction hash addresses for each link
Mapping.PlantTxId = TXID
}else if updateItem == “Process”{
Mapping.ProcessTxId = TXID
}else if updateItem == “Transport”{
Mapping.TransportTxId = TXID
}else if updateItem == “Storage”{
Mapping.StorageTxId = TXID
} else if updateItem == “ Sales “{
Mapping. SalesTxId = TXID
}
stub.PutState(traceabilityCode, Mapping)//Write the mapping relationship to the main chain
return Mapping
```

4.4.3. Query Smart Contract

When consumers query the traceability information of pepper products, by entering the traceability code query, the system calls up the query contract to query the information on the side chain, and the specific query contract pseudo-code is shown in Algorithm 3.

Algorithm 3. Query Smart contract pseudo code.

Input: TraceabilityCode
Output: Return Ciphertext
1: Stub.GetState(TraceabilityCode)//Query TraceabilityCode records
2: (PlantTxId,ProcessTxId,StorageTxId,TransportTxId, SalesTxId) = GetMapping (TraceabilityCode)//Get the address of the transaction hash in the mapping relationship
3: Ciphertext.Plant = PlantChain.GetState(PlantTxId)//Request planting sidechain traceability data based on transaction hash address
4: Ciphertext. Storage = StorageChain.GetState(StorageTxId)//Request Warehouse Sidechain Traceability Data
5: Ciphertext. Process = ProcessChain.GetState(ProcessTxId)//Request for processing side-chain traceability data
6: Ciphertext. Transport = TransportChain.GetState(TransportTxId)//Request logistics side-chain traceability data
7: Ciphertext.Sales = SalesChain.Getstate(TransportTxId)//Request sales sidechain traceability data
8: return Ciphertext//Return to Traceability Information

4.5. Multi-Chain Model Performance Tests

4.5.1. Enquiry Efficiency Performance Test

This experiment is a simulation test in a virtual machine with an environment based on Centos 7.9, Docker 18.09, and Fabric 1.4. The system runs on 4 GB of RAM, 20 GB of ROM, and 10 Mb/s of bandwidth. The Fabric federation network consists of 4 peer nodes and 3 orderer nodes.

In order to compare the query efficiency of the multi-chain model with that of the single-chain model, a validation data query time-consuming experiment is designed. Sets consisting of 1000, 4000, 6000, 8000, 10,000 and 12,000 datapoints were prepared to test the time required to query 1, 50, 150 and 200 pieces of traceability data, respectively. The results of the experimental data comparison are shown in Figure 4.

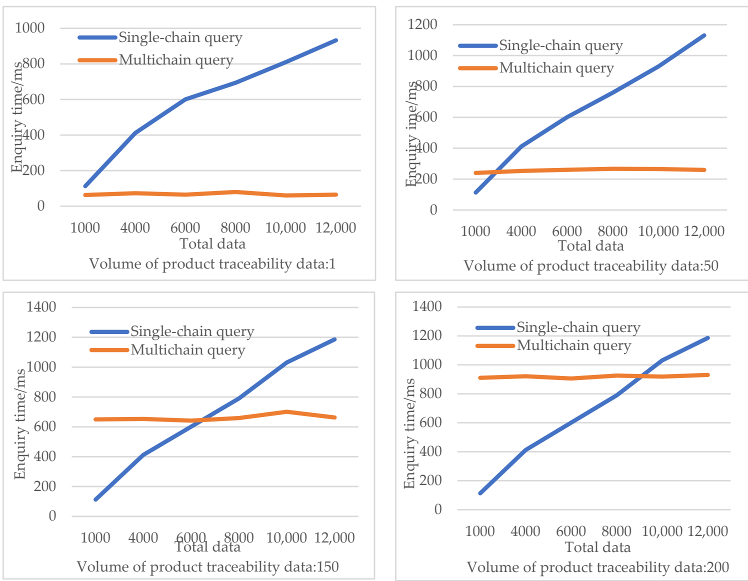

Figure 4. Query time comparison chart.

As shown in the figure, when querying the traceability data, the single-chain query method shows a linear increase in query time as the total quantity of data increases. The query time of the multichain model follows the increase of the number of data entries of the query's traceability records being less affected by changes in total data volume. In the actual pepper industry chain application, the pepper product batch record is about 200, and the total quantity of data in the blockchain system increases with the increase of nodes and time, so the multi-chain model is more suitable for the actual application scenario.

4.5.2. Throughput Performance Test

TPS refers to the number of transactions that the system can process per second, which is one of the important indicators of blockchain system performance. The multi-chain traceability model is tested in comparison with the traditional single-chain blockchain structure to test the throughput of the two storage structures when sending 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500 requests, respectively, under the same test environment, where the requests sent are data query requests. The test results are shown in Figure 5.

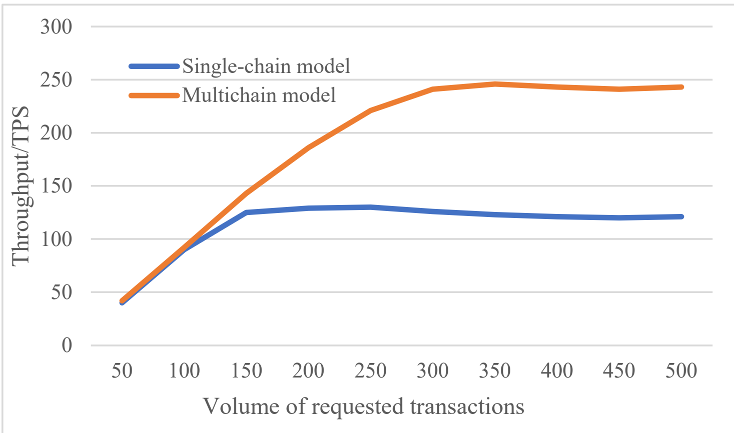

Figure 5. Throughput comparison chart.

The throughput of single-chain model data queries increases linearly when the number of blockchain transactions is between 50 and 150, and it levels off after reaching 125 TPS. The throughput of multi-chain model data queries increases linearly when the number of blockchain transactions is between 50 and 300, and it levels off after reaching 240 TPS. Single-chain query throughput is up to 130 TPS, and multi-chain query throughput is up to 246 TPS. The multi-chain model also works more efficiently than the single-chain model based on the number of transactions completed per second.

5. Conclusions

5.1. Summaries

This paper first introduces the ways to improve the efficiency of blockchain traceability and then chooses to use multi-chain storage to improve the query efficiency and constructs a multi-chain traceability model for pepper products based on the design idea of "main chain + side chain" on the basis of the traditional blockchain traceability model. According to the multi-chain model, the design of the corresponding smart contract is completed, and the function of the model is improved. Eventually, through the model comparison test analysis, the multi-chain model becomes more applicable to the actual traceability scenario of the pepper industry chain and further improves the efficiency of traceability querying.

(1) The proposed multi-chain traceability model compared to the previous blockchain single-chain traceability model will have higher query efficiency than the traditional single-chain model when the total quantity of data is greater than 1000 items. When the number of batch traceability data entries is high and the total quantity of data is large, the query efficiency of the multi-chain model is higher than that of the

single-chain model. The batch traceability data volume of pepper products is about 150 items, and the query efficiency of the multi-chain model is higher than that of the single-chain model when the total data volume is greater than 7000 items. The multi-chain model is more suitable for real traceability scenarios of pepper products.

(2) The model enables full on-chain data storage, making traceability data secure. The nodes in each link of the pepper supply chain can be easily uplinked with data to ensure that consumers can query the complete, accurate, and comprehensive pepper traceability information.

5.2. Prospects

(1) Most of the traceability data in this paper need to be entered manually through the traceability management system, which may generate errors in the entry process. Later, you can use the technology of the internet of things to automatically collected all kinds of information regarding pepper products. The use of IoT equipment makes the traceability process of pepper products precise and automated, reduces the errors and misoperations caused by human input, and improves the operational efficiency of the traceability system and the accuracy of the data.

(2) It may also be possible to enhance the integration of "traceability systems + big data". The traceability system is capable of collecting complete data from all links in the industry chain, but it is not capable of easily processing the data. Enhancing the level of integration with big data can efficiently and flexibly analyse and process data in the supply chain, analysing more valuable information for industry workers.

Author Contributions: Software, W.J. and M.Z.; Formal analysis, W.J.; Investigation, W.J.; Writing—original draft, W.J.; Writing—review & editing, P.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Shandong Province Key R&D Program (Rural Revitalization Science and Technology Innovation Enhancement Action Plan), grant number 2022TZXD0030.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zheng, Y.G.; Jing, Y.S. (Eds.) *Atlas of Medicinal Plants of Taihang Mountain Series*; Shanghai Scientific & Technical Publishers: Shanghai, China, 2021; *in press*.
2. Zhang, Z.G.; Dong, Y.S. *Chongqing Xiaomian Dictionary*; Chongqing Publishing Group: Chongqing, China, 2021; *in press*.
3. Hui, R.H. *Financial Empowerment of Pepper Industry to Do Better and Stronger to Promote Modern Agricultural Development—Research and Thinking Based on the Development of Pepper Industry in a Region*; Modern Enterprise: Seattle, WA, USA, 2022; pp. 152–153.
4. Zheng, Y.N.; Fan, M.T. Determination of toxic and essential metal elements in pepper in Shaanxi. *China Condiment* **2012**, *37*, 88–91.
5. Li, J.F. Analysis determination of elementary in Clovershrub pepper. *China Condiment* **2012**, *37*, 87–88.
6. Peng, Y.; Yao, J.; Yu, Y.H. Investigation and Evaluation on Heavy Metal Content and Pesticide Residues of Three Edible Forest Products in Sichuan Province. *J. West China For. Sci.* **2018**, *47*, 26–30.
7. Chen, X.F.; Tang, S.C.; Gou, J.J. Determination of iron, copper and manganese trace elements in peppercorns of different origins. *China Condiment* **2014**, *39*, 121–123.
8. Foong, S.Y.; Ma, N.L.; Lam, S.S.; Peng, W.; Low, F.; Lee, B.H.; Alstrup, A.K.; Sonne, C. A recent global review of hazardous chlorpyrifos pesticide in fruit and vegetables: Prevalence, remediation and actions needed. *J. Hazard. Mater.* **2020**, *400*, 123006. [CrossRef] [PubMed]
9. Bosona, T.; Gebresenbet, G. Food traceability as an integral part of logistics management in food and agricultural supply chain. *Food Control* **2013**, *33*, 32–48.
10. Sun, C.H.; Yu, H.J.; Luo, N. Fruit and Vegetable Blockchain Traceability Data Storage Method of Fruit and Vegetable Foods Supply Chain Based on Smart Contract. *Trans. Chin. Soc. Agric. Mach.* **2022**, *53*, 361–370.
11. Creydt, M.; Fischer, M. Blockchain and more-Algorithm driven food traceability. *Food Control* **2019**, *105*, 45–51. [CrossRef]

12. Zhou, E.; Hong, Z.; Zhao, D.; Pei, Q.; Guo, S.; Akerkar, R. MSTDB: A Hybrid Storage-Empowered Scalable Semantic Blockchain Database. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 8228–8244. [CrossRef]
13. Dong, S.H.; Xin, J.C.; Hao, K. Optimization Algorithm for Connected Queries in a Multi-Blockchain Environment. *J. Zhejiang Univ.* **2022**, *56*, 313–321.
14. Jia, D.Y.; Xin, J.C.; Wang, Z.Q.; Guo, W.; Wang, G.R. An efficient query model for storage capacity scalable blockchain systems. *J. Softw.* **2019**, *30*, 2655–2670.
15. Min, H. Blockchain technology for enhancing supply chain resilience. *Bus. Horiz.* **2019**, *62*, 35–45. [CrossRef]
16. Agarwal, V.; Pal, S. Blockchain meets IoT: A scalable architecture for security and maintenance. In Proceedings of the 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Delhi, India, 10 December 2020.
17. Chen, J.; Lv, Z.; Song, H. Design of personnel big data management system based on blockchain. *Future Gener. Comput. Syst.* **2019**, *2019*, 1122–1129. [CrossRef]
18. Xing, X.; Chen, Y.; Li, T.; Xin, Y.; Sun, H. A blockchain index structure based on subchain query. *J. Cloud Comput.* **2021**, *10*, 52. [CrossRef]
19. Guo, C.; Liu, Y.; Na, M.; Song, J. Dual-Layer Index for Efficient Traceability Query of Food Supply Chain Based on Blockchain. *Foods* **2023**, *12*, 2267. [CrossRef] [PubMed]
20. Zhang, H.; Wei, Z.Q. Query model of blockchain system based on secondary indexing mechanism. *J. Comput. Appl.* **2022**, *42*, 129–134.
21. Ruan, P.; Dinh, T.T.; Lin, Q.; Zhang, M.; Chen, G.; Ooi, B.C. LineageChain: A fine-grained, secure and efficient data provenance system for blockchains. *VLDB J.* **2021**, *1*, 3–24. [CrossRef]
22. Liang, H.; Liu, S.C.; Zhang, Y.N. Multi-chain Blockchain Application Technology for Agricultural Products Transaction. *Smart Agric.* **2019**, *1*, 72–82.
23. Zhu, Y.C. *Research on Query Processing for Blockchain Systems*; East China Normal University: Shanghai, China, 2020.
24. *GB/T 30391-2013*; Prickly Ash. National Standards of People's Republic of China: Beijing, China, 2013.