EasyChair Preprint

№ 1525

Food Supply Chain Traceability Scheme based on Blockchain and EPC Technology

Haihui Huang, Xiuxiu Zhou and Jun Liu

September 15, 2019

Food Supply Chain Traceability Scheme based on Blockchain and EPC Technology

Haihui Huang[1], Xiuxiu Zhou[1(✉)], Jun Liu[2]

[1] School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China
huanghh@cqupt.edu.cn
s170131122@stu.cqupt.edu.cn
[2] School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China

Abstract. In order to effectively detect and prevent food safety issues and track responsibilities, it is indispensable to establish a reliable traceable system. Accurate recording, sharing and tracking of specific data in food production, processing, warehousing, transportation and retailing is particularly important throughout the food supply chain. This paper proposes a safe food traceable scheme based on blockchain and EPC technology. Encoding food by EPC technology, using the Ethernet block chain and smart contract to effectively execute transactions, manage all transactions among participants involved in the supply chain ecosystem, to track and trace food in the entire agricultural supply chain. In addition, a data management system structure combining on-chain and off-chain is proposed, which uses IPFS to store data under the chain, to alleviate the data explosion in the block chain of the Internet of Things. Which provide safe, efficient and transparent food traceable scheme.

Keywords: Food Traceability, Blockchain, Smart Contract, EPC, IPFS.

1 Introduction

[1]Product traceability is an indispensable means of modern supply management and a key technology to solve food safety problem [1]. At present, the mainstream product

[1] This work is supported by the National Natural Science Foundation (Grant No. 61772099, 61772098); the Program for Innovation Team Building at Institutions of Higher Education in Chongqing (Grant No.CXTDG201602010); Chongqing Science and Technology Innovation Leadership Support Program (Grant No. CSTCCXLJRC201917); the University Outstanding Achievements Transformation Funding Project of Chongqing (Grant No. KJZH17116); the Artificial Intelligence Technology Innovation Important Subject Projects of Chongqing (cstc2017rgzn-zdyf0140); The Innovation and Entrepreneurship Demonstration Team Cultivation Plan of Chongqing (cstc2017kjrc-cxcytd0063); the Chongqing Research Program of Basic Research and Frontier Technology (Grant No. cstc2017jcyjAX0270, Grant No. cstc2018jcyjA0672, Grant No. cstc2017jcyjAX0071); the

traceability system is mainly controlled by government departments or a core enterprise [2], traceability records are processed by a department or company. This traditional traceability system has the following hidden dangers: information tampering, label copy, accountability difficult, spamming Products.

To solve the above hidden dangers, many companies, governments [3−4] and related researchers and technology companies have proposed solutions based on blockchain technology. Xiwei Xu[5] et al. designed a blockchain-based traceable system originChain, which reconstructs the current system by replacing the central database with blockchain, and provides high-availability transparent tamper-proof traceability data for originChain. Abeyratne [6] prospected the application of block chain technology in supply chain management, and analyses the possible problems in supply chain management. Niya et al [7] employed SC on the Ethereum blockchain (BC), their Decentralized Application provided a hardware-and platform-independent approach that flexibly enables multiple object combinations and transformations to be tracked with a use case-agnostic design and utilization. Kentaroh [8] et al. used the Ethereum architecture to create a blockchain smart contract model for supply chain management, and designed a project-level smart contract to manage event information for products in the supply chain. Feng Tian [9] constructed a traceability system based on RFID technology and blockchain to prevent the tracing system information from being tampered with. To protect data privacy, Zyskind [10] et al. have established a blockchain-based personal data management system to ensure users protect data privacy in a distributed situation. Zhu LieHuang [11] et al proposed a controllable blockchain data management (CBDM) model that can be deployed in a cloud environment. Gai Keke [12] et al presented a consortium blockchain-oriented approach to solve the problem of privacy leakage without restricting trading functions. Gai Keke, Wu Yulu [13] et al proposed a model Permissioned Blockchain Edge Model for Smart Grid Network (PBEM-SGN) to address the two significant issues in smart grid, privacy protections and energy security, by means of combining blockchain and edge computing techniques.

From the above research, the main problems of block chain traceability schemes at present are small scale, small number of nodes on the chain, low traceability efficiency, imperfect design of intelligent contracts, and data explosion caused by data overload. To solve the above problems, this paper proposes a collaborative food safety traceability scheme based on block chain and EPC coding.

The main contributions of this paper are as follows:(1) We proposed a safe food traceability solution based on block chain and EPC technology to prevent data tampering, improve traceability accuracy.(2) Applying IPFS to store data under the chain, we can alleviate the data explosion on the block chain by dynamically managing the data on and off the chain.(3) Manage transactions on the chain by deploying smart contracts, through authentication to prevent disclosure of sensitive information, improve the efficiency of traceability and security.

Industry Important Subject Projects of Chongqing(Grant No. CSTC2018JSZX-CYZTZX0178, Grant No. CSTC2018JSZX-CYZTZX0185)

The rest of this article is organized as follows. Part 2 introduces the related techniques of the blockchain traceability scheme; part 3 describes the system architecture and data flow on the chain; part 4 describes the algorithmic design of smart contracts in the supply chain, part 5 analyses the performance of smart contracts. Finally, part 6 analyses the experimental results.

2 Related Technology

2.1 EPC Technology

The EPC [14] system is based on computer Internet and radio frequency technology RFID. It applies EPC coding technology to uniquely encode each entity object, and construct an "Internet of Things" that realizes real-time sharing of global item information. The core idea of the EPC system is to scan the electronic tag by radio frequency identification technology, read the unique identifier EPC code of the entity object in the tag, complete the data collection, and obtain the EPC code by the RFID and then transmit the code to the server connected to the Internet to Store and query subsequent data.

2.2 IPFS

IPFS is a point-to-point distributed hypermedia distribution protocol. IPFS is based on content addressing, saving information to IPFS nodes, and the IPFS system will return a unique hash value calculated based on this information. The hash value corresponds to the content of the message, and even if the information is slightly modified, a completely different hash value will be obtained. When IPFS is requested for a file hash, it uses a distributed hash table to find the node where the file is located, gets the file and verifies the file data [15].

2.3 Ethereum

Ethereum [16] is a blockchain development platform that supports smart contracts and lowers the threshold for users to build blockchain applications. Just like Bitcoin, Ethereum is an open source blockchain [17] underlying system. In addition, one of the biggest features of Ethereum is the combination of smart contracts, trustless, no tampering. It provides a credible execution environment for the operation of smart contracts.

3 Design of Traceability scheme

3.1 Traceability Architecture Design

The traceability scheme is mainly composed of producer users, distributors, regulators and consumers. Manufacturer users combine EPC technology to capture and manage key traceability information for their products. The entire system architecture is shown in Figure 1. The manufacturer node server consists of five modules whose functions are described in detail as follows:

- EPC Traceability Information Collection Module: This module is designed to collect key traceability information generated during food production, storage, and distribution. Relevant data can be automatically collected by RFID or staffed to identify.
- Event Information Database: This database is mainly used to save and manage all food information in the information collection module.
- Effective Information Extraction Module: This module is mainly used to extract information that needs to be uploaded the blockchain from the traceability information database, and prepare to upload data.
- Blockchain Module: The blockchain module has two functions. One is data interaction, including key traceability information for uploading blockchains, requests for information on the chain, and verification of event information. The other is to provide the user with the option to become a complete blockchain node or a lightweight blockchain node, which is to decide whether to participate in the maintenance of the blockchain.
- Interaction Rights Management Module: When there is any event information interaction, the module is responsible for the verification of the user identity, that is, whether the requestor that initiated the event information request is in the supply chain.

Traceability client consists of two modules:

- Blockchain Module: This module is designed for the link between the client and the system. It can request the information of the blockchain and verify the legitimacy of the information. Select a light node for the module to reduce user maintenance costs.
- Information Cache Database: This cache database is used to cache the corresponding food traceability data tracked by the user.

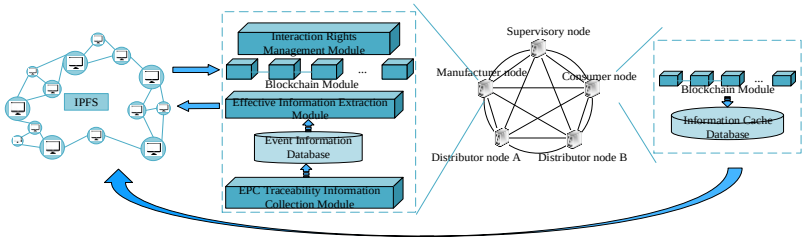

Fig. 1. Food traceability scheme architecture based on blockchain and EPCIS

3.2 Data flow on the chain

The system's data flow includes the process of uploading data to the blockchain, the interaction of offline data and consumer queries. The design of these two processes is as follows:

1. Process of Data Uploading to Blockchain

(1) Food producer A produces food and assigns a unique identifier, The collection of event data for the food is done by the data collection module of A's

enterprise user server, and then A will store the collected information by traceability information database.

(2) A extracts key traceability information through the information extraction module, and stores in local database.

(3) A will put relevant certification documents and the manufacturer's detailed information in the local database, encrypt the information and upload it to IPFS.

(4) A call the smart contract writes the encrypted file and traceability information into the block, and automatically generate A's smart contract transaction through the blockchain module. When the peer-to-peer (P2P) network accepts the transaction and successfully uploads it to the blockchain, the manufacturer will transfer the goods.

(5) When dealer B receives these products, it is necessary to verify the legality of the goods by initiating a discovery service request to the corresponding manufacturer A's smart contract. If B is verified, the blockchain module will return the encrypted information hash of A and the server IP address or URL of A.

(6) Distributor B uses the hash value returned by IPFS to query, obtains the encrypted detailed information, and submits its identifier to the manufacturer's server to initiate a request for event data to share the product from A. It compares the hash value with the encrypted hash of A in local database, so we can know whether the data has been tampered with and obtain the detection information of the food.

2. Consumer Inquiry

The detailed process of event data interaction for consumer queries can be divided into six steps:

(1) After receiving the product, the retailer submits the product identification code and the address of the smart contract to the blockchain to request the information discovery service.

(2) The smart contract judges the identity of the requester. Once confirmed, the smart contract will return the manufacturer's server address and encrypted information to the retailer.

(3) The retailer initiates a request for event data to the manufacturer's server and submits its identity (including the public key and the digital signature created by its private key).

(4) The interactive rights management module of the manufacturer server initiates a rights verification request to the smart contract.

(5) The smart contract judges the retailer's authority and returns the result.

(6) According to the judgment of the smart contract, the interactive rights management module determines whether to return the event data.

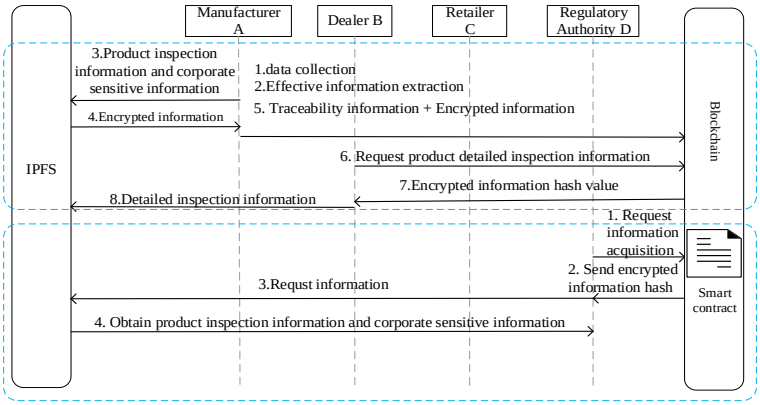

Fig. 2. Data writing and query process

4 Trading contract design

4.1 Entity relationship

To ensure that products are safely tracked using Ethereum Smart Contracts and all participants participate in the process, the manufacturer produces the product and maintains the detailed process in the product production process. Manufacturers record the details of product production in a decentralized file system through IPFS and store the IPFS hash of related files in smart contracts.

The distributor purchases the finished product from manufacturer to ship the product to buyers. After that, the dealer sells the product to the retailer. Figure 3 shows the entity relationship diagram, illustrating the smart contract attributes and functions and the relationship between participating entities and smart contracts.

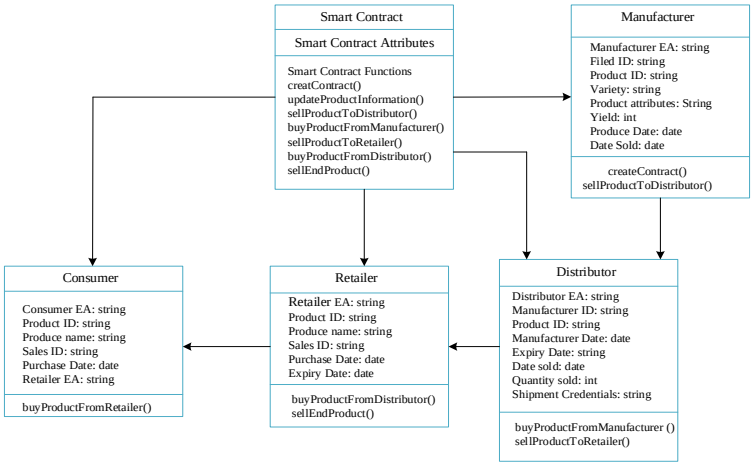

Fig. 3. Entity relationship diagram

4.2 Transaction algorithm design

In this section, we describe an algorithm that defines how our proposed blockchain-based approach works. As mentioned earlier, manufacturers have created smart contracts. Then it agrees to the purchase terms with a registered distribution company. Algorithm 1 describes the process by which a manufacturer sells a product to a distributor. After the initial state of the contract is established, the smart contract checks to confirm that the requesting distribution company has registered and paid the price of the product. If the plan is successful, the contract status will change to *SeedRequestSubmitted*, the distributor status will change to *WaitForProduct*, and the manufacturer status will change to *AgreeToSell*. The contract notifies the change in the entity activity in the owner chain, otherwise the contract status and other activity participants will return to the initial state and the transaction will terminate.

Algorithm 1 Manufacturer sells product to Distributor

Input: *D is the list of registered Distributor*
Etherenumaddress(EA) of Distributor,
Etherenumaddress(EA) of Manufacturer,
DateManufactured, Quantity , ProductPrice

1 Contractstate is ***Created***
2 State of the Distributor *is* ***ProductsRequested***
3 Manufacturer state is ***Ready***
4 Restrict access to only $f \in F$ i.e., registered Distributors
5 if Distributor = *registered and ProductPrice = paid* **then**
6 Contract state changes to *ProductRequestSubmitted*
7 Change State of the *Distributor* to *WaitForProduct*
8 Manufacturer state is *AgreeToSell*
9 Create a notification message stating sale of product
10 end
11 else
12 Revert contract state and show an error.
13 end

Once the distributor receives the product, it can sell it to the retailer. As shown in Algorithm 2, At this stage, the contract status is *ProductSoldToDistributor* and the distributor status is *ProductReceived-FromManufacturer*. The retailer's status is *ReadyToPurchase*.

Contract restrictions can only be accessed by registered retailers and check for acceptance of the sales agreement and completion of product payments. If these con-

ditions are met, the contract performs the transaction that the distributor delivers the product to the retailer. The contract status changes to *SaleRequestAgreedSuccess*, the distributor status changes to *ProductSoldToRetailer*, and the retailer status changes to *ProductDeliveredSuccessful*. Otherwise, for failures, the contract status changes to *SaleRequestDenied*, the distributor status changes to *RequestFailed*, the retailer status changes to *ProductDeliveryFailure*.

Algorithm 2 Distributor Ships Product to Retailer

Input: *'r' is the list of registered Retailers*
Etherenumaddress(EA) of Distributor, Etherenumaddress(EA) of Retailer,
DateManufactured, Quantity Sold, DatePurchased

1 Contractstate is ***ProductSoldToDistributor***
2 Distributor state is ***ProductReceivedFromManufacturer***
3 i Retailer state is ***ReadyToPurchase***
4 Restrict access to only r retailer
5 if *Sale = agreed and ProductPayment = successful* **then**
6 Contract state changes to *SaleRequestAgreedSuccess.*
7 Distributor state changes to *ProductSoldToRetailer.*
8 Retailer state is *ProductDeliveredSuccessful*
9 Create a 'success' notification message.
10 end
11 else
12 Contract state changes to *SaleRequestDenied.*
13 Distributor state changes to *RequestFailed.*
14 Retailer state is *ProductDeliveryFailure*
15 is Create a request failure notification message.
16 end
17 else
18 Revert contract state and show an error.
19 end

5 Traceability scheme performance analysis

Table 1 shows the gas value consumed by different nodes when they call the smart contract to trade after deploying the smart contract to the block chain, and the corre-

sponding ether consumption in the Ethernet. From Table 1, it can be seen that users only need to spend very little ether when making transaction requests on the chain, and the transaction costs will not cause losses to the interests of the nodes, so it is implementable.

Table 1. Transaction consumption.

Transaction Type	consumption(gas)	consumption(ETH)
updateProductInformation()	21272	0.000021
sellProductInformation()	23896	0.000024
buyProductFromMfacturer()	26264	0.000026
sellProductToRetailer()	25048	0.000025
buyProductFromDistributor()	21271	0.000021
sellEndProduct()	21272	0.000021

In Table 2, we compare the performance of the intelligent contract designed in this paper with that of Xu's[5] data reading delay in the paper, the time unit is milliseconds. From Table 2, we can see that because the block chain data is read locally but not sent to the block chain network, the data can be read quickly. Compared with Xu's scheme, the smart contract data read latency designed in this paper is lower and the performance is a little better.

Table 2. Reading latency(ms).

	Xu's scheme	Mine
Minimum	8	6
Fist Quartile	10	12
Median	11	1
Third Quartile	13	10
Maximum	129	120
Average	17	13

6 Conclusion

In this article, we first introduced the difficulties and challenges of current mainstream food traceability program. Then, through in-depth analysis of the main needs of users, we designed a food traceability scheme based on blockchain and EPC technology. To alleviate the data explosion problem, we used collaborative management of on- and-off-chain data to reduce the amount of data for a single node. Constructed a smart contract module for trading on chain to ensure the security and effectiveness of node transactions. The experiment proved that the intelligent contract designed in this paper had good performance and was implementable.

References

1. Fors E., Thankur M., Solem K., Svarva R.: State of traceability in the Norwegian food sectors. Food Control, 62–69 (2015). http://dx.doi.org/10.1016/j.foodcont.2015.03.027
2. Ricardo BM., Mishra P., Ruiz-Garcíaa L.: Food Traceability: New Trends and Recent Advances. A Review. Food Control, 393-401 (2015). https://doi.org/10.1016/j.foodcont.2015.05.005
3. Distributed ledger technology: Beyond blockchain. In: Technical Report, 2016.UK Government Chief Scientific Adviser.
4. Staples M., Chen S., Falamaki S., Ponomarev A., Rimba P.: Risks and opportunities for systems using blockchain and smart contracts. In: Technical Report, Sydney, 2017 Data61(CSIRO). http://dx.doi.org/10.1007/978-3-030-16184-2
5. Xiwei Xu., Qinghua Liu., Yue Liu., Liming Zhu., Haonan Yao., Athanasios V.Vasilakosd.: Designing blockchain-based applications a case study for imported product traceability. Future Generation Computer Systems, 399–406(2019). https://doi.org/10.1016/j.future.2018.10.010
6. Abeyratne SA., Monfared R.: Blockchain ready manufacturing supply chain using distributed ledger. In: International Journal of Research in Engineering and Technology, 05(09), pp. 1-10(2016).
7. Niya SR., Dordevic D., Nabi AG., Mann T, Stiller B.: A Platform-independent, Generic-purpose, and Blockchain-based Supply Chain Tracking. 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). IEEE,2019:14-17.
8. Toyoda K., Mathiopoulos P T., Sasase I., Ohtsuki T.: A Novel Blockchain-Based Product Ownership Management System (POMS) for Anti-Counterfeits in The Post Supply Chain. IEEE Access, 2017:1-1. https://doi.org/10.1109/ACCESS.2017.2720760
9. Tian F.: An agri-food supply chain traceability system for China based on RFID & blockchain technology, 2016 13th International Conference on Service Systems and Service Management (ICSSSM). IEEE, 2016:1-8. https://doi.org/10.1109/ICICTA.2009.700
10. Zyskind G., Nathan O., Alex.: Decentralizing Privacy: Using Blockchain to Protect Personal Data. 2015 IEEE Security and Privacy Workshops (SPW). IEEE Computer Society, 2015. https://doi.org/10.1109/SPW.2015.27
11. Zhu L., Wu Y., Gai K., & Choo, K. K. R. Controllable and trustworthy blockchain-based cloud data management. Future Generation Computer Systems, 2019, 91:527-535.
12. Gai K., Wu Y., Zhu L., Qiu M., Shen M. Privacy-Preserving Energy Trading Using Consortium Blockchain in Smart Grid . IEEE Transactions on Industrial Informatics, 2019.
13. Gai K , Wu Y , Zhu L , Xu L. Permissioned Blockchain and Edge Computing Empowered Privacy-preserving Smart Grid Networks. IEEE Internet of Things Journal, 2019:1-1.
14. Wang S.: Internet of things based on EPC technology and its application in logistics. 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC). IEEE, 2011. https://dx.doi.org/10.1109/AIMSEC.2011.6010861
15. Nizamuddin N., Salah K., Ajmal Azad M., J Arshad., M H Rehman.: Decentralized document version control using ethereum blockchain and IPFS. Computers & Electrical Engineering, 2019, 76:183-197.
16. Bahga A., Vijay K., Madisetti A.: Next-Generation Smart Contract and Decentralized Application Platform. Journal of Software Engineering and Applications, 2016.
17. Nakamoto S.: Bitcoin: A peer-to-peer electronic cash system. Consulted, 2008.