BIG DATA SHARING AND HIGH-EFFICIENCY TRACEABILITY FOR BLOCKCHAIN-BASED SUPPLY CHAIN MANAGEMENT

WU HANQING

PhD

The Hong Kong Polytechnic University

2022

The Hong Kong Polytechnic University

Department of Computing

Big Data Sharing and High-efficiency Traceability for Blockchain-based Supply Chain Management

Wu Hanqing

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
December 2021

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: ____________________

Name of Student: Wu Hanqing

Abstract

Although supply chain management is of remarkable market value and plays a vital role in the global economy, the underlying technologies are underdeveloped, especially from the computer science perspective. In particular, the data from various supply chain stakeholders are not interoperable, leading to high operation costs. Moreover, the traceability service is not provided in most modern supply chains, which brings severe concerns in terms of product quality.

In this thesis, we propose employing the latest blockchain technology in supply chain management, namely blockchain-based supply chain management, which connects various supply chain stakeholders and provides traceability services. Blockchain provides distinctive features, such as immutability and authenticity, for supply chain management. In particular, the product information stored on blockchain cannot be tampered with once stored. Moreover, blockchain systems can authenticate product records without centralized authorities. We have developed novel methodologies of big data sharing and high-efficiency traceability for supply chain management.

First, we present a survey about blockchain-based supply chain management and propose a system architecture. Supply chain management is fundamental for gaining financial, environmental, and social benefits in the supply chain industry. Although there are some proof-of-concept studies and surveys on blockchain-based supply chain management from logistics, the underlying technical challenges are not identified. We provide a comprehensive analysis of potential opportunities, new requirements, and principles of designing blockchain-based supply chain management systems. We summarize and discuss four crucial technical challenges in scalability, throughput, access control, data retrieval and review the promising solutions. Finally, a case study of designing a blockchain-based food traceability system is reported to provide more insights into tackling these technical challenges in practice.

Second, we introduce a big data sharing solution for blockchain-based supply chain management. Nowadays is the big data era. A large amount of data are generated, which can be valuable for business, healthcare, transportation, etc. Researchers have been trying to design and develop data-sharing platforms to promote the dissemination of valuable data. However, the existing platforms fail to address at least one of the three issues: trustworthiness, data heterogeneity, and authenticability. To this end, we propose TSAR, a fully distributed Trustless data ShARing platform. In detail, we architect TSAR on Blockchain to remove the dependency on reliable third parties, which realizes the trustworthiness. Moreover, we propose a general data schema to represent raw data, which handles data heterogeneity. Finally, we record the data transaction as well as user-group information on blockchain to achieve authenticability.

Finally, we proposed a high-efficiency traceability algorithm for blockchain-based supply chain management. Supply chain traceability refers to product tracking from the source to customers, demanding transparency, authenticity, and high efficiency. In recent years, blockchain has been widely adopted in supply chain traceability to provide transparency and authenticity while the efficiency issue is inadequately studied. In practice, as the numerous product records accumulate, the time- and storage- efficiencies will decrease remarkably. For the first time, we studied the efficiency issue in blockchain-based supply chain traceability. Compared to the traditional method, which searches the records stored in a single chunk sequentially, we replicate the records in multiple chunks and employ parallel search to boost the time efficiency. However, it is challenging to allocate the record searching primitives to the chunks with maximized parallelization ratio. To this end, we model the records and chunks as a bipartite graph and solve the allocation problem using a maximum matching algorithm.

We believe this thesis is a significant step towards enhancing the supply chain efficiencies, reducing the operation cost, and most importantly, storytelling the consumers about the provenance and journey of products.

Publications Arising from the Thesis

1. Hanqing Wu, Shan Jiang, Jiannong Cao, “High-efficiency Blockchain-based Supply Chain Traceability”, under *Major Revision* in *IEEE Transactions on Intelligent Transporation Systems* (2021).

2. Hanqing Wu, Jiannong Cao, Yanni Yang, Cheung Leong Tung, Shan Jiang, Bin Tang, Yang Liu, Xiaoqing Wang, Yuming Deng, “Data Management in Supply Chain Using Blockchain: Challenges and A Case Study”, in *The 28th International Conference on Computer Communications and Networks* (2019).

3. Hanqing Wu, Jiannong Cao, Shan Jiang, Ruosong Yang, Yanni Yang, Jianfei He, “TSAR: A Fully-Distributed Trustless Data ShARing Platform”, in *The 4th IEEE International Conference on Smart Computing* (2018).

4. Shan Jiang, Jiannong Cao, Hanqing Wu, “Dynamic Ring Signature: Achieving Provable Anonymity in Blockchain-based E-voting”, under *Major Revision* in *ACM Transactions on Software Engineering and Methodology* (2021).

5. Shan Jiang, Jiannong Cao, Hanqing Wu, Yanni Yang, “Fairness-based Packing of Industrial IoT Data in Permissioned Blockchains”, in *IEEE Transactions on Industrial Informatics* (2020).

6. Shan Jiang, Jiannong Cao, Hanqing Wu, Yanni Yang, Mingyu Ma, Jianfei

He, “BlocHIE: A BLOCkchain-Based Platform for Healthcare Information Exchange”, in *The 4th IEEE International Conference on Smart Computing* (2018).

7. Jiannong Cao, Shailey Chawla, Yuqi Wang, Hanqing Wu, “Programming Platforms for Big Data Analysis”, in *Handbook of Big Data Technologies* (2017).

Acknowledgments

First and foremost, I would like to express my sincere gratitude my supervisor, Prof. Jiannong CAO, for his continuous support of my PhD study at PolyU. He has provided inspiring guidance and incredible help on every aspect of my research. From the very beginning of choosing a research topic to work on several research projects, from technical writing to doing presentation, I have learnt so much from him not only on knowledge but also on attitude in working. I will always appreciate his advice, encouragement and support at all levels.

My sincere thank to Dr. Shan Jiang for the stimulating discussions, for many sleepless nights we were working together on Huawei, Alibaba collaboration projects and research topics, and for the advice on research career and life. I would also like to thank Lei Yang, Xuefeng Liu, Yuqi Wang, Linchuan Xu, Yaguang Huangfu for their valuable guidance on my early research exploration. I would like to express sincere gratitude to my group members, Wengen Li, Jiaxing Shen, Yanni Yang, Bin Tang, Yu Yang, Ruosong Yang, Zhiyuan Wen, Juncen Zhu, and Yinfeng Cao, who gave me advice and kind help.

My special thanks go to my dear friends, Qianyun Liu, Shaoqi Shen, Yilian Wang, Conglin Li, Rong Xiang, Tianshu Zhu, Ye Chen, Mingyu Ma and many others for all the wonderful memories over these years. Without them, my life would never be so enjoyable.

Last but not least, I would like to thank my dear parents for supporting me both financially and spiritually. Without their deep love and constant support, this thesis would never have been completed.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background & Motivation

In commerce, supply chain management (SCM) refers to the flow management of goods and services along the whole supply chain, including raw materials, manufacturing, warehousing, transportation, inventory, consumption. According to statistics from Allied Market Research, the global SCM market size was valued at 15.85 billion US dollars in 2019 and is projected to reach 37.41 billion by 2027, growing at a CAGR of 11.2% from 2020 to 2027. There is no doubt that supply chain management plays a vital and indispensable role in the global economy.

Despite the importance of SCM, its enabling technology is underdeveloped, especially from the computer science perspective. On the one hand, the data is not interoperatable among the supply chain stakeholders, leading to high operation costs. The stakeholders and regulatory authorities have to make much effort to define and adjust the data format from other stakeholders before usage. Tremendous human resources are wasted in connecting the stakeholders. On the other hand, traceability service is rarely provided in modern supply chains. Supply chain traceability is crucial in terms of allowing product tracking from the sources to end consumers. It provides opportunities to enhance the supply chain efficiencies, meet the regulatory requirements, and most importantly, to story-tell the consumers about the provenance and journey of products. In terms of the products whose safety is critical, e.g., food and pharmaceuticals, supply chain traceability is critical and has been pursued for decades by the industries [49].

In this thesis, we propose a framework and mechanism leveraging blockchain technology to record supply chain data, connect various stakeholders, and provide traceability services. Blockchain provides distinctive features, such as immutability and authenticity, for SCM. In particular, the product information stored on the blockchain cannot be tampered with once stored. Moreover, blockchain systems can authenticate product records without centralized authorities. We will develop novel methodologies for big data sharing and high-efficiency traceability.

1.2 Research Framework

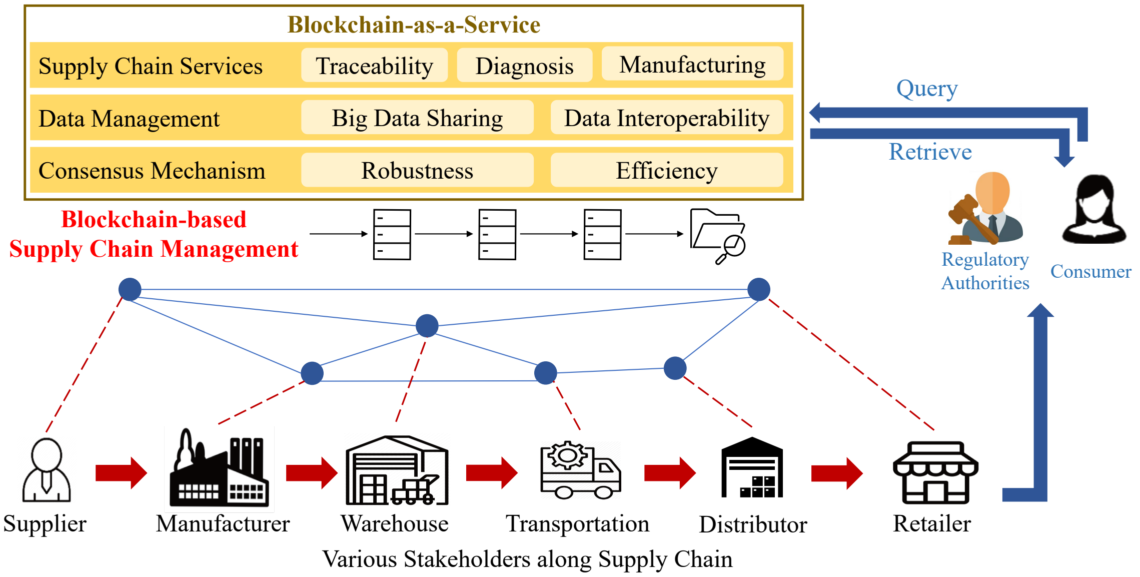

Figure 1.1: Research Framework

Fig. 1.1 depicts the research framework of this thesis.

- At the bottom layer is the supply chain. Various stakeholders, e.g., suppliers, factories, warehouses, transportation, and retailers, form the supply chain and provide regulatory authorities and customers with content services.
- The supply chain stakeholders join and maintain the blockchain network, which is blockchain-based SCM. The product records will be stored on blockchain.
- The blockchain-based SCM system consists of three layers, i.e., consensus mechanism, data management, and supply chain services. The consensus mechanism focuses on robustness and high efficiency. The data management layer includes big data sharing and data interoperability. The supply chain services include traceability, diagnosis, and manufacturing.
- The customers and regulatory authorities can query the blockchain-based supply chain system to enjoy the supply chain services.

1.3 Thesis Organization

The rest of this thesis is organized as follows:

- Chapter 2 presents a survey about blockchain-based SCM and the system architecture of blockchain-based SCM. SCM is fundamental for gaining financial, environmental, and social benefits in the supply chain industry. However, traditional SCM mechanisms usually suffer from a broad scope of issues such as lack of information sharing, long delays for data retrieval, and unreliability in product tracing. Recent advances in blockchain technology show great potential to tackle these issues due to its salient features, including immutability, transparency, and decentralization. Although there are some proof-of-concept studies and surveys on blockchain-based SCM from logistics, the underlying technical challenges are not identified. This chapter provides a comprehensive analysis of potential opportunities, new requirements, and principles of designing blockchain-based SCM systems. We summarize and discuss four crucial technical challenges in scalability, throughput, access control, data retrieval and review the promising solutions. Finally, a case study of designing a blockchain-based food traceability system is reported to provide more insights into tackling these technical challenges in practice.

- Chapter 3 introduces a big data sharing solution for blockchain-based SCM. Nowadays is the big data era. A large amount of data are generated, which can be valuable for business, healthcare, transportation, etc. Researchers have been trying to design and develop data-sharing platforms to promote the dissemination of valuable data. However, the existing platforms fail to address at least one of the three issues: trustworthiness, data heterogeneity, and authenticability. To this end, we propose TSAR, a fully distributed Trustless data ShARing platform [77]. In detail, we architect TSAR on Blockchain to remove the dependency on reliable third parties, which realizes the trustworthiness. Moreover, we propose a general data schema to represent raw data, which handles data heterogeneity. Finally, we record the data transaction as well as user-group information on blockchain to achieve authenticability. To demonstrate the practicability and effectiveness of TSAR, we implement it in a minimal-viable-product fashion and evaluate the performance in terms of throughput and response time.

- Chapter 4 develops a high-efficiency traceability algorithm for blockchain-based SCM. Supply chain traceability refers to product tracking from the source to customers, demanding transparency, authenticity, and high efficiency. In recent years, blockchain has been widely adopted in supply chain traceability to provide transparency and authenticity while the efficiency issue is inadequately studied. In practice, as the numerous product records accumulate, the time-

and storage- efficiencies will decrease remarkably. This chapter is the first work studying the efficiency issue in blockchain-based supply chain traceability to the best of our knowledge. Compared to the traditional method, which searches the records stored in a single chunk sequentially, we replicate the records in multiple chunks and employ parallel search to boost the time efficiency. However, it is challenging to allocate the record searching primitives to the chunks with maximized parallelization ratio. To this end, we model the records and chunks as a bipartite graph and solve the allocation problem using a maximum matching algorithm. The experimental results indicate that the time overhead can be reduced by up to 85.1% with affordable storage overhead.

- Chapter 5 concludes this thesis with the summarization of the main contents and our ideas in terms of the future of blockchain-based SCM.

Chapter 2

System Design of Blockchain-based Supply Chain Management

The global supply chain market surged over $13 billion in 2017 and is expected to soar past $19 billion by 2021 with the additional revenue opportunity from Software as a Service (SaaS). Although the supply chain industry has excellent potential for development, it suffers from a wide gamut of issues in SCM. To name a few, the lack of transparency and information sharing to delays in data retrieval affecting every stage of a logistics network. Furthermore, due to the centralized and separated systems in the current SCM, product authentication and traceability cannot be achieved decently, which the industry is struggling to handle.

The revolution of SCM relies on reliable and efficient data management, in which the data collected from supply chains are supposed to be stored, integrated, and retrieved with reliability and high efficiency. Facing the issues mentioned above, people are heading towards the application of blockchain technology on SCM. Blockchain, originated from Bitcoin, is an implementation of an append-only ledger. It stores fully traceable and immutable records, which can be transformed from the data along with the supply chain, e.g., product and sales information. As such, blockchain enables the authenticity of the digitalized data in the supply chain. Meanwhile, it can be used as an overall system to integrate the data flow and each step in the supply chain for efficient data management.

With much attention attracted from the blockchain for SCM, the researchers have conducted conceptual analysis [14][28][36] on the potential opportunities, advantages, and concerns when applying blockchain in different supply chains. From the perspective of logistics, blockchain has been positively recognized by the community. There have been some specific systems developed for some particular supply chain applications, for example, luxury and food supply chain [81][50][72]. Despite these significant efforts, few studies focus on the technical challenges of applying blockchain for SCM in practice.

In this chapter, we emphasize figuring out the technical challenges in blockchain regarding its application in SCM. In particular, the large number of stakeholders leads to the scalability issue of the blockchain network. In terms of the vast amount of data generated from the supply chain, the overall system throughput and latency of every single transaction should be guaranteed to make the system more user-friendly. Next, we discuss the issues about the access control in the blockchain system to prevent some data from being exposed to competitors. Finally, the efficiency and reliability of data retrieval to trace the history information in the supply chain are investigated.

After identifying the four technical challenges, we present a case study about the food supply chain and develop a blockchain-based food tracing system. The case study is significant because food safety is a significant concern for the whole society. The system is built upon the Hyperledger Fabric, a permissioned blockchain system. Functions, including the user access control, food data submission, and data query, are realized with the smart contract. Based on the developed system, we test the performance concerning the throughput for data submission and the speed of data query to discuss the effects that are key to the system performance when designing the blockchain.

In summary, the main contribution of this chapter lies in three aspects as follows:

- We provide insight into the potential opportunities to apply blockchain technology in SCM and present an exhaustive survey of existing blockchain-based SCM systems.
- We summarize the pain spots of existing SCM systems and four technical challenges in the design of blockchain SCM systems in practice.
- We implement a food tracing system based on permissioned blockchain for the food supply chain scenario.

The rest of this chapter is organized as follows. Section 2.1 introduces the background of SCM and blockchain with the discussion on the application of blockchain for SCM. Section 2.2 presents the existing works on the investigation and studies of using blockchain for SCM. Section 2.3 raises the technical challenges in designing the blockchain for fulfilling the requirements in SCM. The illustration of the food tracing system as a case study is given in Section 2.4. Finally, Section 2.5 summarizes this chapter.

2.1 Supply Chain Management with Blockchain

In this section, we first introduce the background knowledge of supply chain and its management in brief. Then, the problems in SCM and the potential opportunities for blockchain to overcome the problems are discussed.

2.1.1 Brief Introduction of SCM

Supply Chain (SC) is a system of all the activities, and associated information flows involved in moving products or services from the supplier to the customer. SCM involves managing the activities and information related to sourcing, procurement, conversion, and all logistics. SCM can bring many financial, environmental, and social benefits, e.g., improved resource utilization, the reduced cycle time from order to delivery, and early problem detection.

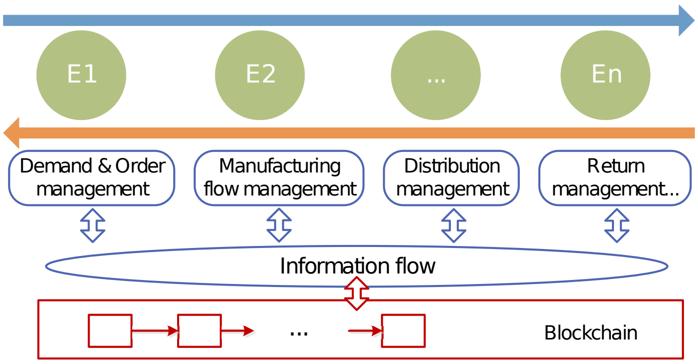

Figure 2.1: SCM with Blockchain

One of the key issues to realize the SCM functions, including demand and order management, manufacturing management, and distribution management, are the data management [74]. The types of data in the supply chain involve but are not limited to inventory information, marketing information, and customer feedback. The circulation of those data within the supply chain is the core of its process and management. In Fig. 2.1, E1, E2, and En denote the stakeholders in the supply chain, e.g., supplier and distributor. For different management functions of the whole process, the information flow provides essential inputs. However, data management in current SC is occlusive and inefficient. In particular, some stakeholders store the data in a stand-alone and offline way. Moreover, the information is exchanged via the postal system. A more effective way is to use the electronic data interchange (EDI) to automate the information flow. However, stand-alone and centralized data management brings some problems. First, data can be tampered which could intrude data authenticity. Besides, system security can be violated. Finally, the retrieval of the data is time-consuming. Therefore, to promote the supply chain data management, we consider the following guidelines: (1) improve the coordination and information sharing within SC; (2) protect the data authenticity; (3) speed up the data retrieval.

2.1.2 Blockchain for SCM

The blockchain is an append-only list of blocks, each of which includes multiple pieces of data, managed by a peer-to-peer network adhering to a protocol for inter-node communication. The magic of blockchain lies in the protocol of validating new blocks, in short, consensus mechanism. The majority of nodes will agree on the presence of each block via consensus algorithm after validating the data in the block. It can be pretty hard to tamper with the data on the blockchain since most of the nodes will not admit it. There are various blockchain systems, including permissionless and permissioned blockchains, targeting different application scenarios.

The nature of blockchain technology brings about the features of system decentralization at the same time with data immutability. These features provide potential opportunities for fulfilling the requirements of supply chain data management. First, by using the blockchain to store and manage the data flow in the supply chain, the information cannot be easily tampered with and treated as reliable proof of existence. Second, data from different stakeholders in the supply chain can be integrated into the blockchain system rather than separately stored in individual systems, which helps for the data sharing and saves cost and time for data retrieval. In the next section, we will introduce the existing works for applying blockchain for SCM.

2.2 Existing Works on SCM with Blockchain

Some existing works are performing conceptual studies on how to improve SCM by applying blockchain technology. The adoption of blockchain for SCM is commonly related to supply chain data management. As shown in Fig. 2.1, the blockchain systems are connected to the information flow. The data for SCM is the input to the blockchain system. For data management, especially for the supply chain, the first issue is to collect the data where the sensing technologies can play their roles[17][51]. Many works are introducing the usage of IoT devices, e.g., RFID [70], and NFC [4], to collect the data from the physical world and convert it into digital information.

Second, after data collection, researchers try to understand and identify the application of blockchain in SCM via different approaches. Some works collect feedbacks from the people in the community of logistics and economics by using questionnaires [36][70] so that they can gain more insights and the requirements of SCM from the industry. Some discuss the effects of blockchain in SCM based on different theories and frameworks, like Attributed of Innovation Framework [28] and Unified Theory of Acceptance and use of technology [32]. In general, the essential property of the blockchain, which is decentralization, brings out the key features that are appealing to the SCM, including trustlessness, security, and authentication. Then, we can help to deal with the issues in SCM, e.g., traceability [14], cost-efficiency [36], and transparency [12].

Apart from the systematic analysis of the usage of blockchain in SCM, there are many deployed and conceptual systems for various specific SC scenarios, among which the food supply chain is the hottest topic. The food provenance and safety issues are important problems to take care of [71]. There are ongoing projects for food supply chain or food traceability, e.g., FarmShare, Provenance, and ripe.io. More particular applications are about the wine [14] and agricultural food [68]. Pharma and drug industry also attract help from blockchain [72][17] since healthcare is also an important social problem. There are also blockchain applications in the post SC, sand SC, and excipient SC.

However, current studies, especially for those from multiple disciplines, are mainly carried out from the perspectives of logistics, economics, and management. They appreciate the advantages of importing the blockchain. However, they also argue that barriers are still there if blockchain is put into practical use for SCM in terms of the immaturity of the current blockchain technology and deployment cost. Meanwhile, only a few works consider the problems, for instance, security and privacy issues [81][55] in the technical design of the blockchain system. Few studies are focusing on how to make the blockchain system satisfy the requirements of SCM in practice. To this end, this chapter emphasizes figuring out the technical challenges in blockchain regarding its application in SCM. At the same time, we provide some experience in designing the blockchain for the food supply chain as a representative case study.

2.3 Technical Challenges in Blockchain for SCM

Supply chains typically raise issues that are highly dependent on freight failure, human error, intended fraud, and others. However, when applying blockchain technology to the SCM, more factors will impact the system. These factors pose significant challenges in designing and implementing such systems. In the following, we present and analyze four challenges of blockchain technology that explicitly or implicitly affect the supply chain.

2.3.1 Scalability

Many stakeholders are participating in modern supply chains, which are of the global range, together with a massive flow of newly created and time-sensitive information. All the data is poorly handled by modern SCM systems since there are few shared common databases for stakeholders and the supply chain. Blockchain can contribute significantly by providing a networked and decentralized database for all supply chain parties to join. However, the blockchain works in a decentralized fashion with stakeholders alongside the supply chain participating and interacting, which differs from the traditional SCM Electronic Data Interchange (EDI) systems which work in a centralized way with the system admin controlling the read and write access to the data. The prime challenge is how the system can scale and operate with the increasing number of stakeholders and a large amount of generated transactional data.

Network Scalability Blockchain platforms possess a specific mechanism that ensures data immutability along with the ledger called consensus. The scope of the consensus is to keep a general agreement among blockchain nodes about all submitted transactions. Such transaction information can be the timestamp, thus the order they occurred, the addresses of the sender and receiver, the amount transacted, the tags or electronic seal that accompany the ware, and others. The essence of blockchain technology innovation and uniqueness originates from the use of consensus mechanisms. Today, blockchain platforms support different types of general agreement tools depending on the ledger level of access. A ledger can be public or private, and typical consensus algorithms constitute PoW (Proof-of-Work), PoS (Proof-of-stake), PBFT (Practical Byzantine Fault Tolerance), PoET (Proof of Elapsed Time), and PoA (Proof of Authority), respectively. There is no standard tool on modern supply chains that organizes and secures each step of the product. Hence errors, fraud, and ware failure are possible. Consensus algorithms constitute the core of the blockchain trust and node general agreement, with well-known methods known as “mining”. On the contrary, such techniques hide dangers, such as 51% attack and reveal problems such as selfish mining, which already happened in real life.

Storage Scalability With the number of transactions increasing day by day, the blockchain becomes heavy since all transactions must be stored to validate the transaction. Currently, Bitcoin and Ethereum blockchains have exceeded 230GB and

208GB storage, respectively. A large block size can increase the system throughput temporarily. However, the increased block size slows the propagation speed down, which leads to blockchain forks. So scalability is quite a challenging problem in the blockchain. A novel cryptocurrency scheme was proposed to solve the bulky blockchain problem. In the new scheme, the network removes old transaction records, and a database named account tree is used to hold the balance of all non-empty addresses. In this way, nodes do not need to store all transactions to check whether a transaction is valid or not. Besides, the lightweight client could also help fix this problem. A novel scheme named VerSum [73] was proposed to provide another way of enabling lightweight clients. It allows lightweight clients to outsource expensive computations over large inputs. It ensures that the computation result is correct by comparing results from multiple servers.

2.3.2 Throughput

Various actions and procedures occur during the journey of a product inside the supply chain. They are often prone to human errors or even fraud or ware failure, which as a result, diminish system performance. With blockchain, the majority of activities can be represented as electronic transactions submitted on the ledger. In that case, those activities execute faster and without errors increasing SCM system performance. However, it is not easy to guarantee the system throughput in the blockchain.

While previous work has identified additional metrics, system throughput is the bottleneck issue and more challenging to address from a research perspective. Bitcoin's transaction throughput is a function of its block size and inter-block interval. With its current block size of 1MB and 10-minute inter-block interval, the maximum throughput is capped at about seven transactions per second; and a client that creates a transaction has to wait for at least 10 minutes on average to be sure that the transaction is included in the blockchain. In contrast, mainstream payment-processing companies like Visa confirm transactions within a few seconds and have a high throughput of up to 24,000 transactions per second.

Current research is focused on developing solutions to improve blockchain performance while retaining its decentralized nature significantly. Reparametrization of Bitcoin's block size and inter-block interval can improve performance to a limited extent, estimated by a recent study at 27 transactions per second and 12 seconds, respectively. However, significant performance improvement requires a fundamental redesign of the blockchain paradigm.

2.3.3 Fine-grained Access Control

The modern supply chains suffer from participants' societal fears about loss of privacy and data protection where any kind of data is available and can be tampered with. It makes many large companies unwilling to share their data which creates data silos along the supply chain. Blockchains offer a decisive contribution when it comes to those issues. The blockchain ledger contains immutable data and participants' privacy can be highly respected and preserved with corresponding access control measures.

We classify blockchain data into two categories: user identity and transactional data. For the user identity, permissionless blockchains, e.g., BTC and ETH, offer its users pseudonymity because users only make transactions with newly generated addresses rather than real identities to avoid identity exposure. Thus, there is no longer any central party keeping users' private information. While inside a permissioned blockchain, total anonymity can be assured that participants are joining in an anonymized way while being authenticated in advance by an off-chain system, e.g., the government, FDA, of the supply chain. In this way, the supply chain system functions appropriately with participants' real identities kept safely from the other network users, and it is guaranteed that they are legal participants. Nevertheless, in [13], the authors presented a method to link user pseudonyms to IP addresses even when users are behind NAT or firewalls. Moreover, each client can be uniquely identified by a set of nodes it connects to, which can be learned and used to find the origin of a transaction.

However, the more valuable assets in the supply chain are transactional data. Under some circumstances, participants would prefer to reveal their identities while the transactional data, e.g., manufacturing logs, retailer and consumer sales information, needs to be protected with fine-grained access control. Transactions in a permissionless blockchain are visible to the public, while the read permission depends on the permissioned blockchain. The supply chain consortium could decide whether the stored data is public or restricted with levels of access control. Some works have been addressing this issue in the past few years. In [84], the authors propose a decentralized personal data management system implemented on the blockchain that ensures the user ownership of their data. In [58], it proposed a new approach based on blockchain to publish the policies expressing the right to access a resource and to allow the distributed transfer of such rights among users. In [45], a blockchain-based platform for healthcare information exchange is proposed to satisfy the requirements of both privacy and authenticity.

2.3.4 Data Retrieval

When adopting blockchain technology with SCM, sharing accurate and timely information throughout a supply chain yields significant benefits to all participants along the supply chain. Every recordable data requires peer-to-peer verification, which can be time-consuming with the number of blocks involved when tracing the linkage of data backward. With the data passing through the peer-to-peer verification and the block containing the data appended to the blockchain, it is essential to retrieve the desired data from the blockchain efficiently and effectively.

Participants have different requirements and expectations for data retrieval. Wholesalers want to trace forward the product to get the accurate situation of sales in order to make a better marketing strategy and increase the revenue in return. The consumers want to know the authenticity of the product quickly to decide the purchase without too much hesitation confidently. Those requirements place challenges in data retrieval. The efficiency of data retrieval means that the queried data should return the results within a reasonable time range. The reliability of data retrieval means that the return results should not be incomplete and tampered with.

Retrieval by Full Node In existing mainstream blockchain applications, it is not easy to achieve efficient or reliable data retrieval. For example, in both BTC and ETH, a user who wants to do the query has to be a full node that fully downloads every block and transaction and checks them against blockchain's consensus rules which may take days for the startup. It poses a heavy burden on both data storage and network bandwidth, and it is user-unfriendly. After that, the blockchain data is locally available and imported into designated databases to rebuild indexes for retrieval. However, it is evident that this local data retrieval manner is inefficient or even practical for nodes with low-end hardware.

Retrieval by lightweight node Online data retrieval is possible because a node can also be a lightweight node that does not download the complete blockchain. Instead, the light nodes download the block headers only to validate the authenticity of the transactions. Because of this reason light nodes are easy to maintain and run. However, light nodes need full nodes to connect to the network, and therefore, the effectiveness of data retrieval completely depends on the full nodes to function. More downsides come from privacy and security consideration. For privacy, the light nodes typically send transactions to a trusted third party, allowing the third party to spy on all the users' past and future activities. For security, the light nodes may skip several security steps, leaving the user vulnerable, and the trusted third party can be a malicious node to launch the middleman attack.

Besides data storage, the blockchain is also supposed to provide data retrieval services to the stakeholders along the supply chain. However, it is non-trivial to search for data on blockchain efficiently while preserving data privacy. In particular, the data stored on blockchain are typically encrypted, which hinders efficient search algorithms. Moreover, the search operations are special in SCM, for example, forward search and backward search.

2.4 Case Study

In this section, we first introduce the system framework to build a blockchain-based SCM system. Then, we demonstrate the implementation based on Hyperledger Fabric with the functionality design, system architecture, and implementation details. Finally, we conduct extensive experiments to analyze the system performance regarding response times of user registration, data submission, and data query.

2.4.1 System Overview

Food safety is the primary concern for society nowadays. Problems including food fraud, illegal production, and food-borne illness in the food supply chain have resulted in much damage to customers' health and loss in the food industry. Governments and many organizations have paid close attention to the food safety problems and taken measures to deal with them. However, there is a long and arduous way to go.

Our system adopts the concept of blockchain as a Service (BaaS) [46]. BaaS is an offering that allows customers to leverage cloud-based solutions to build, host, and use their blockchain apps and functions. With the help of BaaS, clients only need to be concerned about the functions they want to realize. From the technical view, blockchain service providers provide flexible choices about different data, consensus, or smart contract components in the blockchain infrastructure. In addition, it could be much easier to operate multiple chains simultaneously.

Therefore, our blockchain-based food traceability system is built upon the Hyperledger Fabric, an open-source and permissioned distributed ledger technology platform. We use Fabric because it provides modular and configurable architecture, which is potential for the realization of BaaS. Besides, it is permissioned, similar to the federated blockchain with moderate trust among different parties, and suitable for supply chain applications.

The general framework of our food traceability system is shown in Fig. 2.2. The bottom layers are the network layer and data layer. The submitted data are aggregated into blocks and serialized into the form of a chain. When preparing each data block, cryptographic functions can be used for data representation and connecting consistent blocks. The proposed data and blocks are broadcasted via different protocols. The second layer is the consensus layer, including leader election and transaction packing [44]. There are many consensus algorithms, e.g., PoW and PBFT. In the Fabric, we can insert our own designed consensus algorithms. Currently, we use the PBFT for leader election in the system. Then, the selected node would select and pack data pieces into one block. The third layer is the contract layer. Many functions can be realized in the smart contract. The upper layer is the application layer, of which the characteristics are subject to the purposes of different applications.

In our system, there are three kinds of identities: the regulatory authorities, members, and customers. Regulatory authorities have the highest level of access to the blockchain. All the information can be visible to regulators for censorship. In practice, the regulator can be a food safety association or other official organization. Members refer to the suppliers, companies, and other parties in the food supply chain. They have rights to write in, i.e., submitting information to the blockchain, and only the “one back, one up” food information is visible to them. It means that they can only search the last source and next step of the food transferred in the supply chain. Customers do not need to write in, and the information exposed to them only tells the information about the leading producer of the food product and the origin areas of

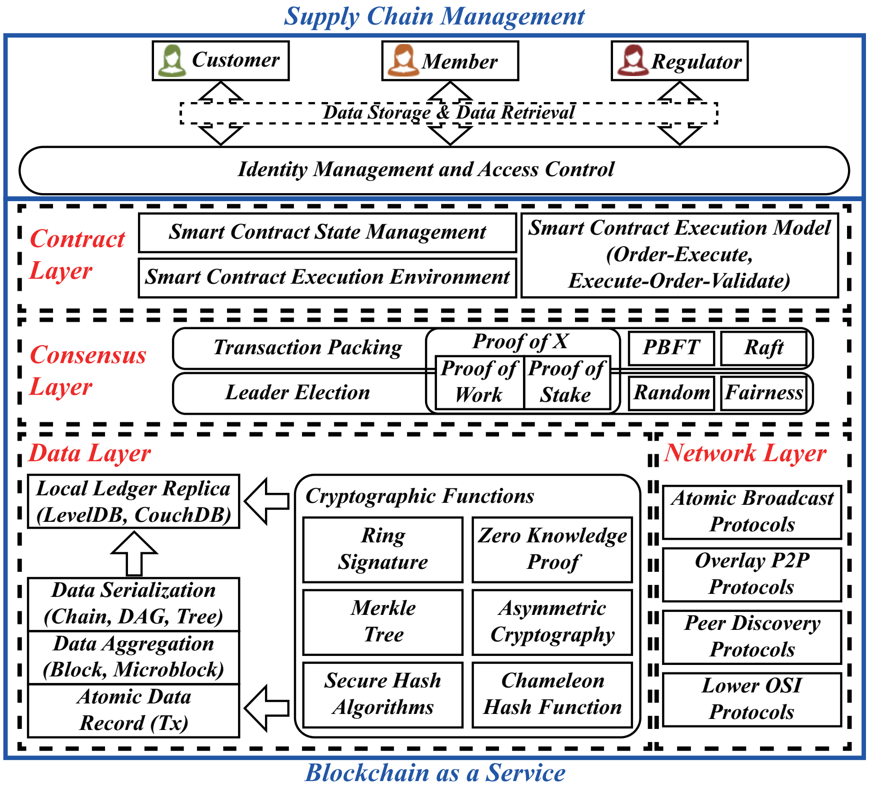

Figure 2.2: Framework of the Blockchain-based Food Traceability System

the raw food. The intermedia information will be hidden from the customers. The access control of the three identities is realized by recording their identity information in the blockchain. Their private key is used for log-in. The identify checking is implemented by the smart contract.

The functionality of the system is briefed in Fig. 2.3. Regulatory authorities and federated members are required to log into the system before further operation. New federated members should register for legitimate access to the system given by the regulators. The registration records and access records would all be stored on the blockchain as proof. Federated members can submit food information and query previously submitted data by invoking smart contract. The submitted food information will be verified by the peers and published on the blockchain. For the regulatory authorities, they search any information that existed in the ledger. While for customers, only can query the information about the food they bought from retailers. There are different smart contracts from implementing the query and search request sent by the regulatory authorities, members and, customers.

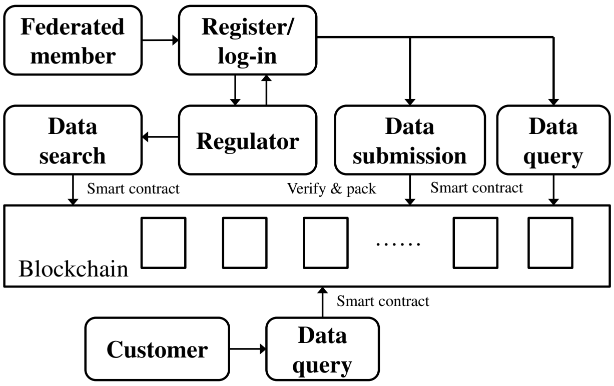

Figure 2.3: Functionality of the food Traceability System

For setting the nodes for the food blockchain, there should be full nodes and light nodes in the network. The full nodes can be set by the regulatory authorities, while the members create the light nodes. One reason for doing this is that only the regulatory authorities could see all the information across the food supply chain due to the privacy concerns of business competitors. Another reason is that if only regulatory authorities have the full nodes, the information can be trustless for the members if regulatory authorities tamper with the original data. Therefore, a group of members could have light nodes that store their food information.

2.4.2 System Implementation

To demonstrate the effectiveness and practicability of the proposed system from a case study, we implement the food traceability system in a minimal-viable-product version. The system is developed under the framework of Hyperledger Fabric.

The current system is developed under Hyperledger Fabric v1.1. The system is implemented on our virtual machines with Docker. Each virtual machine consists of one core two threads of Intel Core i7-8809G 4.2Ghz CPU with 4GB of DDR4 DRAM and 40GB of NVMe SSD, running on Ubuntu 16.04.5.

For the *System and Data Setup*, we deploy four nodes within the department intranet to evaluate the system. The node can run four docker containers for different roles, i.e., 2 for client peer, 1 for orderer, and 1 for endorser, at the same time. The responsibility of network roles is explained as follows. Each node has individual IP such that they can communicate with each other.

- Client: a client that submits an actual transaction-invocation to the endorsers and broadcasts transaction proposals to the ordering service.
- Orderers are nodes that commit transactions and maintain the state and a copy of the ledger
- Endorsers are nodes running the communication service that implements a delivery guarantee, such as atomic or total order broadcast.

2.4.3 System Analysis

In the *System Performance Test*, for the function of *Member Registration*, we invoke the chaincode (smart contract) with 10, 30, and 50 concurrent jobs at the same time and repeat the experiment 10 times to get the average time for the registration, which is successfully finished and recorded on the blockchain, as depicted in Fig. 2.4 as cumulative distribution function (CDF) plot. We also apply the same testing parameters to the function of *Transaction Uploading* and *Data Retrieving* as depicted in Fig. 2.5 and Fig. 2.6 as CDF plots respectively. For the *data uploading*, including system registration and transaction uploading, these two functions will invoke both 1436 individual jobs. For the *data researching*, the total number of the query is 1000.

With the CDF plots from the experimental data, it is not hard to find that the system response time will increase linearly with the increased number of concurrent jobs within one chaincode. Still, the time of data retrieving will increase significantly with the increase of concurrent jobs. When we send ten queries to the blockchain, we can get the most results within 0.3 seconds. However, the average response time of 50 queries takes close to 1 second. The reason is that the system registration and transaction uploading are treated as writing operations to the blockchain, while data retrieving is treated as reading operations. The query may involve multiple transactions' retrieving and validation since these transactions are linked in the blockchain.

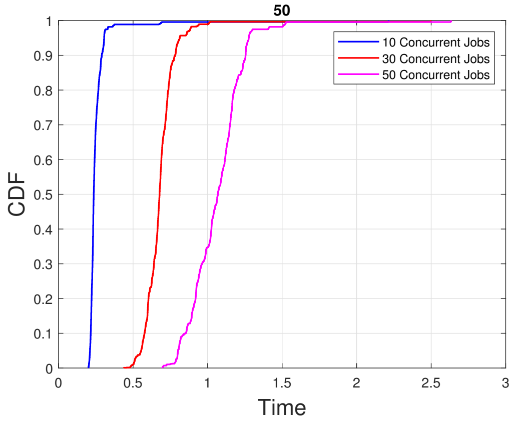

Figure 2.4: CDF of Member Registration Time

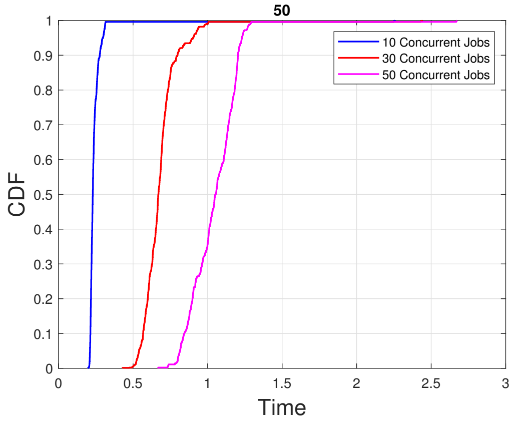

Figure 2.5: CDF of Transaction Uploading Time

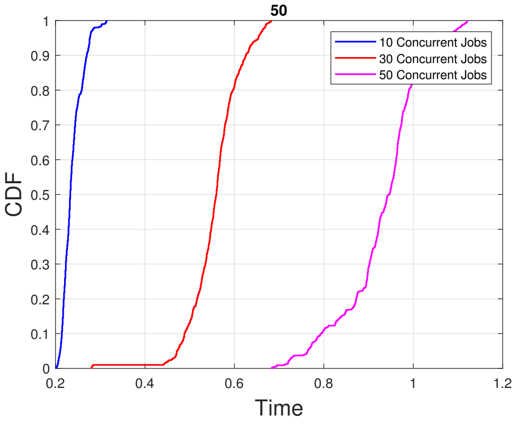

Figure 2.6: CDF of Data Retrieving Time

Throughput Analysis

The current system TPS is not high due to the following factors. First, the current Hyperledger Fabric framework is a generic framework supporting different applications, and it is not optimized for high TPS. Second, the transaction process flows require that the system process and verify transactions one by one instead of parallel. Finally, the transaction size is big and not optimized yet.

In particular, we analyze the *Transaction Size* as a significant impact factor to the system. When using naive design, i.e., one food item submission in each transaction, the system's throughput and write speed are low. The possible solutions can be a follows. On the one hand, we can bundle multiple item submissions in the transaction. On the other hand, we can increase the number of transactions per block. We can also achieve a higher utility rate (payload/total block size) to increase the efficiency by packing related food items together in a single transaction atomically. Moreover, a more complex packing algorithm with a higher computational requirement is needed to lower the real-time capacity. Overall, further study is needed to propose a framework for adjusting the block size and number of transactions to balance system throughput and real-time capability.

Privacy Analysis

In the current system, we focus the privacy protection with the following safety precautions: 1) All the data, e.g., transactions, logs, system events, on the blockchain are encrypted with private-public key pairs. Only the user with the corresponding private key, in the form of owning or being given access rights, can decrypt and view the data; 2) The transaction data can only be seen from neighbor hops, by system default configuration, when doing the data retrieval. It guarantees the traceability of the system while safeguarding the privacy of data. Only the administrator, e.g., the government, FDA, can trace the entire history when a consumer sends an appealing request about the disputed product.

2.5 Chapter Summary

In this chapter, we have introduced blockchain technology in supply chain data management. In particular, we looked into the potential opportunities of applying blockchain for SCM and summarized the existing works on blockchain for SCM. We studied the requirements from SCM when adopting blockchain technology and demonstrated the critical technical challenges in the design of the blockchain for meeting the demands of the supply chain in practice. A case study is presented to address the issues above, and we introduce our proposed food safety tracing system. We implemented this system based on the permissioned supply chain for the food supply chain scenario.

We consider integrating real-world supply chain data with the current system and deploying the system on more federated nodes for future work.

Chapter 3

TSAR: A fully-distributed Trustless Data ShARing Platform

Human beings benefit a lot from big data analytics. For example, the companies analyze the behaviors of their customers based on the collected data and produce products more suitable for the customers [29]. The hospitals take advantage of the gene data, and daily data of the individual for more precise disease treatment and prevention [63]. The airports schedule the boarding of thousands of planes more efficiently by fusing data of weather, ground transportation, etc. [8]

Concerning big data analytics, usually, data from multiple sources are required for one application. Take the taxi as an example. To allocate the taxi drivers more appropriately, various data are needed such as weather data, POI data, traffic data and so on [56] [57] [42]. These data are from various institutions such as the bureau of weather, road transport, and geology. In such a case, data sharing is in urgent need to achieve taxi driver allocation.

However, data sharing is performed in a primitive way in most of the big data analytics applications [53]. The companies figure out what data is needed and ask other institutions whether they can offer the desired data. Then, the question is why the data owner does not want to publish their data directly. It is because of the following reasons. On the one hand, the institutions that own the data are not aware of the value. On the other hand, it is complicated, inefficient, or unsafe for the data owners to publish. Also, it is not guaranteed that the agency will not leak the data. Indeed, the data owners have an alternative to put the data on their official websites, which can hardly be guaranteed that the demander can find the sites.

To this end, it is essential and urgent to build a data-sharing platform. SCM also benefits a lot from big data sharing in the sense that the data from different stakeholders will be interoperable, which improves the efficiency of supply chain management and reduces the operation cost. In research community and market, there are few data sharing platforms [21][24][61]. However, they all suffer from at least one of the following issues. First, they require full trustworthiness from the data owners. Second, the shared data can be heterogeneous, thus requires significant efforts to be managed. Third, the users can have various requirements on the access control policy, such as sharing the data with a specific organization whenever they request it.

In this chapter, we propose TSAR, a fully distributed Trustless data ShARing platform. TSAR addresses the above three issues successfully as follows. TSAR is architected on Blockchain [45], which achieves decentralization. Since there is no third party involved, TSAR is trustless, requiring no trust from the data owners. Moreover, we propose a metadata schema for the data owners to publish their data. In this way, only the metadata can be accessed publicly, not the raw data stored locally. Finally, TSAR provides a Blockchain-based authentication mechanism, which automates the access control of the shared data. The data owners can specify the access control rules in the shared metadata.

The main contributions of this chapter are summarized as follows:

- We propose TSAR, a fully distributed data-sharing platform, which addresses three critical issues in existing systems: trustworthiness, data heterogeneity, and lack of automatic access control mechanism.

- A blockchain-based authentication mechanism is developed in TSAR. It allows the data owners to specify the access control rules in the shared metadata, which enhances the user-friendliness of TSAR.
- We implemented TSAR in a minimal-viable-product fashion. The implementation demonstrates its practicability. We further evaluate the performance of TSAR concerning throughput and response time.

The rest of this chapter is organized as follows. Sec. 3.1 introduces related works. Sec. 3.2 demonstrates the system and module design, and the architecture of TSAR, and three main functions. Sec. 3.3 shows the system implementation and evaluation. Finally, Sec. 3.4 concludes this chapter.

3.1 Related Work

3.1.1 Data Sharing Platforms and Tools

The need to provide easy-to-use tools for sharing big data has resulted in many platforms and tools. The large-scale organized communities, like High Energy Physics, have already developed their data management systems out of the scope for this research. We also consider general file hosting services such as Google Drive out of scope. Four closely related systems that have emerged in the past few years are discussed in detail below: Zenodo, CKAN, Figshare, and IPFS.

Zenodo[2] is a research data repository created and hosted by OpenAIRE and CERN to provide a place for researchers to deposit datasets. Zenodo code is open source and is built on the foundation of the Invenio digital library. It is a general-purpose open access repository, and it supports all types of files. Data can be published under different types of licenses, and it can be flexibly controlled. Zenodo assigns a unique DOI to the data and provides APIs for uploading data and harvesting metadata. The Comprehensive Knowledge Archive Network (CKAN)[1] is a web-based open source management system for storing and distributing open data. It has developed into a powerful data catalog system mainly used by public institutions seeking to share their data with the general public. CKAN supports permanent URIs for citation, e.g., DOIs, by extension packages. It supports RESTful JSON API with essential tools for querying and accessing data. Figshare is an online digital repository where researchers can preserve and share their research outputs, including figures, datasets, images, and videos. In adherence to the principle of open data, it is free to upload content and access it. Users can upload files in any format, and items are attributed to a DOI. Figshare has different functionalities depending on being authenticated user or not. InterPlanetary File System (IPFS) [11] is a content-addressable, p2p hypermedia distribution protocol. Nodes in the IPFS network form a distributed file system. It is a p2p distributed file system that seeks to connect all computing devices with the same file system. IPFS could be seen as a single BitTorrent swarm, exchanging objects within one Git repository. In other words, IPFS provides a high throughput content-addressed block storage model with content-addressed hyperlinks. IPFS combines a distributed hashtable, an incentivized block exchange, and a self-certifying namespace.

3.1.2 Traditional Data Sharing and Transaction Models

Since there is a need for data sharing and data transactions, traditional models exist and are providing services for data sharing and trading. However, the traditional models can not protect the security and interests of both data suppliers and data demanders. Here we classify the traditional data sharing and transaction models into twofold: *Data Hosting Center* (DHC) and *Data Aggregation Center* (DAC).

In the DHC model, each agency will host, upload, and publish its data to the central database controlled and maintained by the DHC. The DHC is responsible for data exchanging and trading with an external agency. After the data is hosted, the data is entirely owned by the DHC. All the follow-up applications of the data are independent of the agency. This model is widely used in the current data-sharing platform due to its convenience, easy operating, and low cost.

In the DAC model, the Center links data services through the API interface among agencies. Data agencies do not need to report, upload to the DAC in advance. The data is still owned and managed by the data agencies. When an agency needs to search the data, it will use real-time interaction with the DAC to send the data request. The DAC will relay and broadcast this request to other agencies. Once other agencies with the target data respond to this request and return it, the DAC will collect all the data and send it back to the data demander. However, it is not hard to find that the DAC has the ability and the opportunity to retain the data. The DAC can accumulate the data during sharing, and it will gradually become a DHC.

3.2 System and Module Design

In this section, we first describe the architecture of TSAR in subsection. 3.2.1. Then, we introduce the three modules, i.e., data publishing, data retrieval, and data sharing, in detail from subsection. 3.2.2 to subsection. 3.2.4.

3.2.1 System Overview

The system architecture is illustrated in Fig. 3.1. For each user who is using TSAR, he/she uses five local components to perform three network functions. The five components are the raw data, the metadata, the metadata chain, the sharingdata chain, and the TSAR interface, while the three network functions are data publishing, data retrieval, and data sharing.

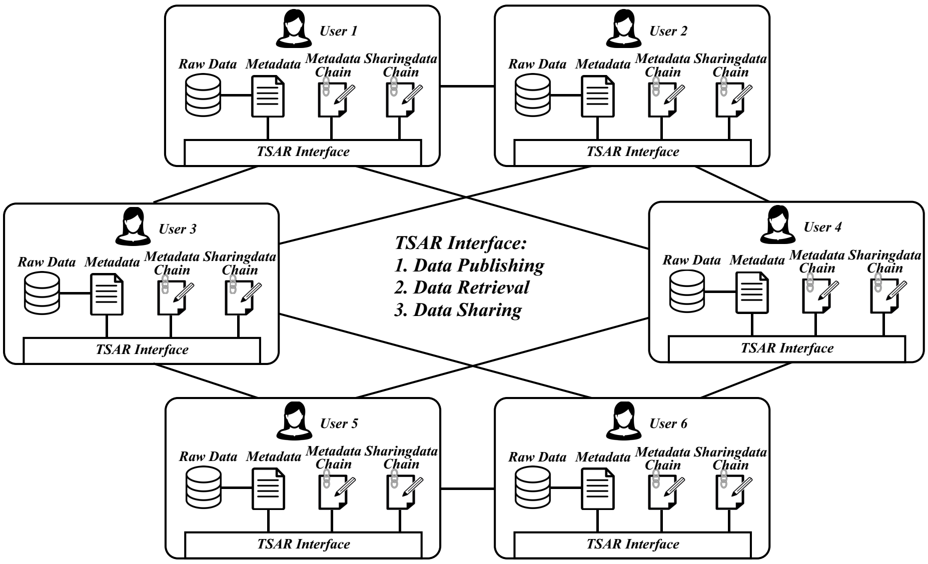

Figure 3.1: System Architecture of TSAR

First, if a user owns some raw data to be shared, the user needs to notify the other users in the TSAR network that there is a piece of newly published data. The process of data publishing involves the components of raw data, metadata, and metadata chain. Specifically, the raw data stored by each user locally is transferred into metadata, and the metadata is published on the metadata chain, which is accessible by all the users on the TSAR network. The metadata chain is a Blockchain, which stores metadatas as transactions. Second, the data retrieval function is required when a user wants to search data with some keywords. Finally, when a user wants to get specific data, data sharing is needed.

3.2.2 Data Publishing

In the traditional data-sharing platform, the users have to upload their raw data or metadata to achieve publishing data. Under this schema, a centralized server collects the uploaded data and displays them. This method heavily relies on a trustworthy service provider. By saying trustworthy, the service provider is not supposed to make any modifications to the uploaded data. To remove such a centralized service provider, we propose to use a metadata chain for publishing data in this subsection.

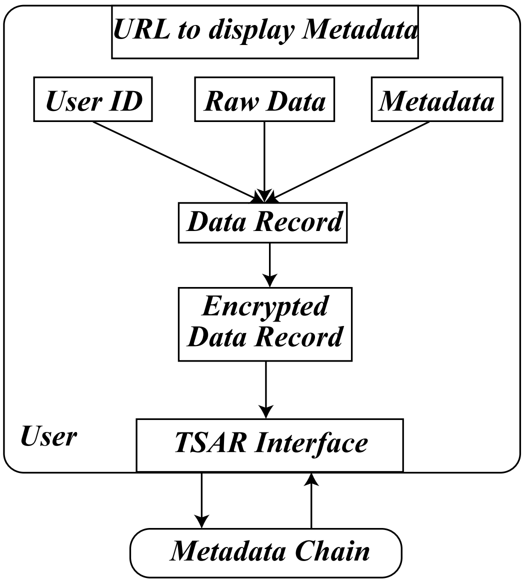

Figure 3.2: Flowchart of Data Publishing

Fig. 3.2 shows the proposed procedure for a user to publish data which is divided into three steps: 1) packing raw data into data record with signature; 2) broadcasting and verifying the data record; 3) synchronize of metadata chain.

The input of the data publishing procedure is the user's raw data. The raw data can be gigabytes and even terabytes. If the raw data is directly published, it is nearly impossible to guarantee its copyright. Also, it is a massive burden to the network. To this end, we define a new data type, metadata, to describe and publish the raw data.

The metadata is a description of the raw data, which contains the data schema, a set of keywords, a small amount of sample data, the acquisition time, and the data size. The format of the metadata is defined to describe the raw data and for high-performance retrieval wholly. The size of a piece of metadata is several hundreds of kilobytes. Compared to the huge-size raw data, it significantly reduces the burden of the network.

After transforming the raw data to metadata, the metadata is published via the HTTP service. In this way, everyone in the network can view the metadata via the corresponding URL. Also, the published metadata cannot be modified by others. However, there are still two issues to handle. The first issue is how to make other users in the network aware of the newly published data. The second issue is how to guarantee that its owner does not modify the metadata after publishing. We propose a metadata chain mechanism for the decentralized recording of data publishing to address these two issues.

After a user generates the URL to display the metadata, a data record composed of user ID, the checksum of raw data, checksum of metadata, and the URL to display the metadata is generated. The data record is encrypted using the user's private key and broadcast to the whole network using the TSAR interface.

If a user receives a data record, the data record will be verified as follows:

- identify the user using the signature in the record;
- acquire the public key of the data publisher;
- use the public key to decrypt the data record;
- whether the data format is as defined;
- whether the signature is the same with the publisher;
- whether the URL contained in the data record is accessible;
- whether the metadata in the URL is as defined;
- whether the metadata checksum is the same with the one in the data record;

The conditions are checked one by one. If there is any unsatisfied condition, the data record will be aborted. Everyone in the network will check every data record to make the data records consistent. If a user verifies a data record, it will be put into the user's local metadata pool. Note that it does not mean that a data record is published in the metadata pool. At a fixed frequency, the data records in the metadata pool will be packed into MetadataChain. If a data record is packed into the MetadataChain, it is published. Each node in the network will synchronize the MetadataChain.

3.2.3 Data Retrieval

As for data retrieval, a central server exists to respond to users' queries in the traditional data-sharing platform. Moreover, for the P2P network, each user sends their query to neighbor peers, and the neighbor peers respond to the query and send it to neighbor peers. The central one needs a central server and balancing load is a complex problem, and the latter one, broadcasting the query to all the peers, is a time-consuming operation, and users may get a response with significant delay.

Our system is fully decentralized and does not need a server and broadcast the query. As mentioned previously, each metadata would be published on a metadata chain, and the users' client of our system responds to its query according to the Metadata Chain. The procedure is as follows:

1. Client Metadata Chain synchronization
2. Word extension and similarity
3. Data retrieval and show results

The users who have attended the system will have performed metadata chain synchronization as mentioned in the former section. However, new users, or users who only want to search for the data they need and have not published any data, have not synchronized the Metadata Chain in their client. So in data retrieval, client Metadata Chain block Synchronization is the first step. The procedure of Metadata Chain Synchronization is the same as the former section. However, synchronizing the whole chain may be a time-consuming cost. We may consider how to avoid downloading the whole chain in the future.

Word Extension and Similarity

Data retrieval aims to respond to the user's query, and in the Metadata Chain, especially in each metadata recorded in each block, there are some keywords to describe the semantic information of the data. The retrieval process is to backtrack each metadata and match the query to the keywords.

This section will introduce how to extend query keywords and match the query keywords and the metadata keywords.

In a practical system, like a search engine, users query is always short and contain very little information. So only using the query words could not get available results, and a standard method to solve this problem is to extend the query words with extra knowledge. There are so many human-designed knowledge bases such as WordNet, which contains synonyms, antonyms, word definitions, etc. In our system, for each query word w_i, we extract the synonyms 3.1 of the word through WordNet and use all these words as the query words. And in order to avoid different word form, such as "traffic" and "traffics", we utilize the NLTK interface to get the stemming form of each key words. 3.2 is the final query keywords set and we use Q to search for the related data.

$$S = \{w_k | w_k \in Syn(w_i)\} \tag{3.1}$$

$$Q = \{Stem(w) | w \in S\} \tag{3.2}$$

$$T = \{Stem(w) | w \in R = Syn(w_{tag})\} \tag{3.3}$$

$$Jacarrd(Q, T) = \frac{Q \cap T}{Q \cup T} \tag{3.4}$$

Then it is important to match the query keywords with each metadata keyword. In order to publish more data in each block, each metadata has very few tags as keywords, and directly comparing these tags with query keywords is also difficult to find the semantically related one. We extend the tags of the metadata using a similar method that with query words. We could get the final tags T 3.3. Moreover, we use the Jacarrd Similarity 3.4 to calculate the distance between query keywords and metadata tags.

Data Retrieval and Show Results

Data retrieval is similar to Search Engines, and we want to return a list of data that may be semantically similar to the user's query. In this subsection, we introduce the whole procedure of data retrieval and how to rank the data. Furthermore, to speed the retrieval process and according to the locality principle, we build a cache for each user to record recent search results. The key idea of data retrieval is to traverse the cache data. After getting preliminary results of the semantically similar data, we want to re-rank the results so that the data with a smaller rank number will be more needed. Following a simple idea, the data is published more early, and the data will be less critical. So we add a weight decay to the data similarity according to the published date, shown in Formula 3.5. F_S means the final similarity, S means the similarity calculated by the former algorithm, D means the publish date of the current data, D_R means the latest publish date during all the list DL.

$$F_S = S * e^{(D - D_R)} \tag{3.5}$$

3.2.4 Data Sharing

The goal of distributed big data-sharing platform is to ensure the authenticity of the data, reliability of the sharing mechanism, and legality of user behavior. The module of data sharing is composed of two functions: (1) data usage under flexible and safe control; (2) reliable data sharing mechanism. With the above two functions, users can freely set different permission modes for sharing data and protect data ownership. Meanwhile, for the data requester, they can also guarantee the authenticity of the data they want.

Data Usage Mechanism

The data usage module of the system is mainly designed to ensure the controllability and reliability of data sharing. For the controllability of data sharing, the user can set different permissions to the data requester to obtain and use the data through the ID of the requester. The way of data usage is divided into the following two models according to the identity of the data requester.

Unlimited Data Usage

Data requesters with unlimited data usage permission search the intended data through MetadataChain, retrieve the data, and send a data-sharing request to the owner. Once the owner has approved the request, the intended data can be sent to the requester. Since TSAR is distributed without a centralized party involved, there are times when the data sent by the data owner does not match the original data information posted by the data owner in the Metadata Chain. In order to guarantee the authenticity of data obtained by the requester, the data authenticity verification function is involved in the TSAR interface, as shown in Fig. 3.3. After B retrieves Data A1 from the Metadata Chain, he broadcasts his request for Data A1 to the entire network. Data owner A sends the data to B via the TSAR interface after receiving B's request. Before Data A1 is sent to B, the TSAR interface first checks the data. If the metadata parsed from the data is the same as the metadata A posted before, TSAR will encrypt Data A1 and send it to B. B will get the encrypted data of Data A1. If the metadata parsed by TSAR does not match Metadata A1, the system will reject the data-sharing action and ask A to send the original data of Data A1.

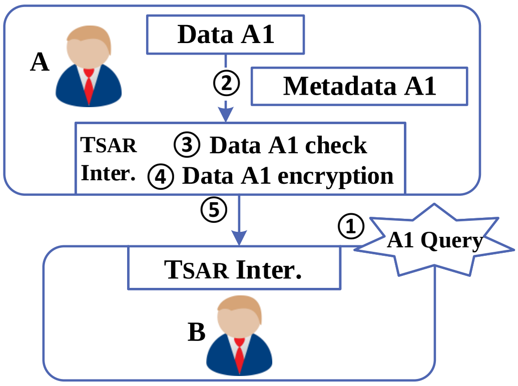

Figure 3.3: Mechanism to Guarantee Data Authenticity

Limited Data Usage

Users with limited data usage permission cannot directly access the original data owner but can obtain the desired processing result by sending an operation instruction to the data owner. In this case, the system needs to ensure that the data owner's data is not maliciously acquired or damaged and ensures that the data requester can obtain the data processing result more conveniently and accurately.

Data Sharing Mechanism - Sharingdata Chain

Data ownership protection is one of the critical functions in distributed big data-sharing platforms. There are cases when the data requester tampers and republishes the data obtained from other users or even derives profits from it, which seriously infringes the ownership of the data owner. Therefore, through the mechanism of Sharingdata Chain, the system stores the record of data sharing information in the Sharingdata Chain. The data sharing records on the Sharingdata can be regarded as evidence for data ownership, and it can also be used for tracking the behavior of data sharing.

The design of the Sharingdata Chain follows the concepts of blockchain. Sharingdata Chain makes some innovations based on blockchain technology for the scenario of big data sharing. The primary role of the Sharingdata Chain is to store the data recording record as proof for data ownership. A data-sharing record in the Sharingdata Chain contains the following information:

1. Data owner, data requester and their signatures
2. metadata pointer, verification code, and URL of the shared data
3. Sharing time and the permission mode
4. Other additional terms

For users with permission of unlimited data usage, the workflow of the Sharingdata Chain is shown in Fig. 3.4). Data requester B publishes the request and data sharing contract for requesting Data A1 to the entire network. After receiving the request, user A checks and sends the encrypted Data A1 to user B. During this period, A packs the decryption key (Decryption) of the encrypted Data A1 data and the signed data-sharing contract into the Sharingdata Chain. After the Sharingdata block containing the data sharing record is authenticated, B can obtain the decryption key and recover the encrypted Data A1.

For users with permission of limited data usage (as shown in Fig. 3.5), data requester B publishes the request and contract for using Data A1 to the whole network and then sends the code and its results on the data sample. Afterward, user A processes Data

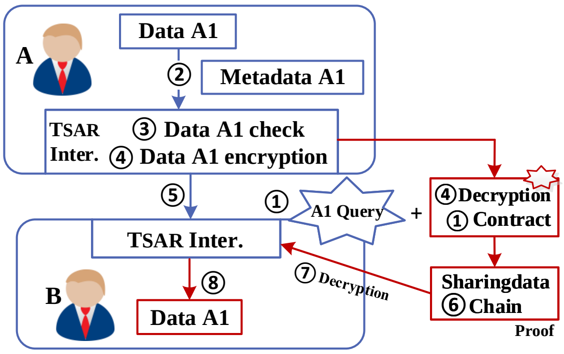

Figure 3.4: Workflow of Sharingdata Chain for Unlimited Data Usage

A1 with the code and sends the encrypted processing result to B through the TSAR interface. At the same time, user A packs the decryption key (Decryption) together with the signed data-sharing contract as a data usage record into the Sharingdata Chain. As long as the block with this record is authenticated, user B will get the decryption key to retrieve the encrypted data result.

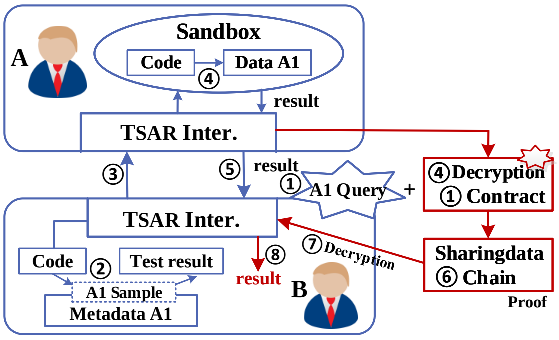

Figure 3.5: Workflow of Sharingdata Chain for Limited Data Usage

3.3 System Implementation & Evaluation

The critical challenge in implementation is the two Blockchains, i.e., metadata chain and sharingdata chain. In TSAR, we implement blockchain under the framework of gRPC. A transaction in TSAR is defined to consist of six fields, namely timestamp, source, destination, hash value, type, and body. The timestamp and hash values are the approximate submitting time and the SHA256 hash value of the transaction.

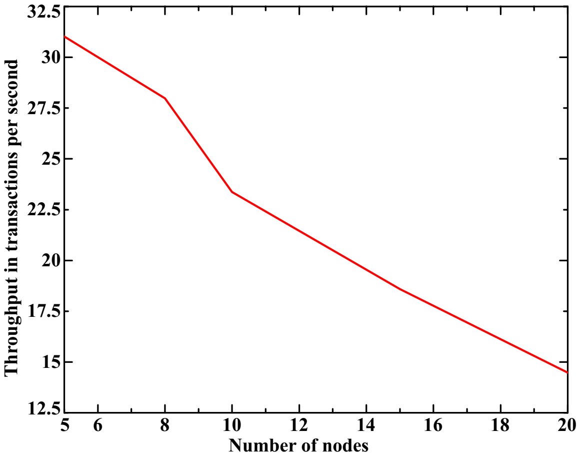

Figure 3.6: Throughput vs. Number of Nodes

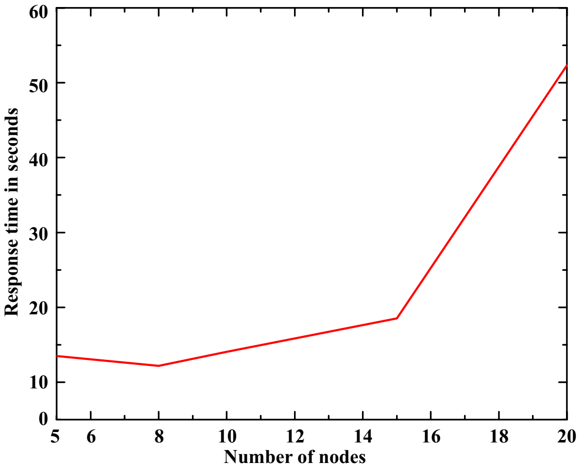

Figure 3.7: Response Time vs. Number of Nodes

We evaluate the performance of the implemented blockchain by investigating how the number of nodes affects the throughput and response time. We conduct experiments on 5, 8, 10, 15, and 20 nodes, respectively. In each set of experiments, each node serves as both client and server. That is, each node generate transactions and pack transactions at the same time. The transaction generation rate for each node is five transactions per second. The evaluation result is shown in Fig. 3.6 and Fig. 3.7. The evaluation results indicate that as the number of nodes increases, the system throughput decreases, and the response time increases. It is true since the system becomes more robust if there are more replicas of the data in the network. However, it requires more network resources, which results in degradation of the system performance.

3.4 Chapter Summary

In this chapter, we propose TSAR, a fully distributed Trustless data ShARing platform. There are three key innovation points in the design of TSAR. First, we architect TSAR on Blockchain, which removes the need for trustworthy third parties. Second, we propose sharing metadata, which is a description of the data, rather than raw data, which decreases the demand of network resources and copes with data heterogeneity. Third, we record the data transaction on blockchain, which achieves non-repudiability. We implement TSAR in a minimal-viable-product fashion and evaluate the system performance concerning throughput and response time. The experimental results indicate the practicability and effectiveness of TSAR.

Chapter 4

High-efficiency Blockchain-based Supply Chain Traceability

In 2019, the global supply chain market value surpassed 14.6 trillion US dollars, having increased at a compound annual growth rate of 10.8% since 2015 [9]. There is no doubt that the supply chain plays a vital role in the global economy. SCM, which refers to the flow management of goods and services, including all the processes that transform raw materials into final products along the supply chain, is essential for boosting customer services, reducing operation costs, improving financial positions, etc. [59]

Among the services of SCM, traceability is crucial in terms of allowing product tracking from the sources to end consumers [54]. Supply chain traceability provides opportunities to enhance the supply chain efficiencies, meet the regulatory requirements, and most importantly, to story-tell the consumers about the provenance and journey of products. In terms of the products whose safety is critical, e.g., food and pharmaceuticals, supply chain traceability is critical and has been pursued for decades by the industries [49].

In recent years, blockchain has been regarded as a promising solution for supply chain traceability because of the distinctive features of immutability, transparency, and auditability [35][19]. Generally, blockchain is an append-only list of blocks, each containing a set of transactions maintained by a decentralized peer-to-peer network. The product records stored on the blockchain are publicly available and cannot be modified, making the stored information reliable. The auditability makes it possible to track the product information on blockchain. To summarize, blockchain empowers supply chain traceability with high reliability, and auditability [75, 62, 78].

Besides the applications of blockchain-based supply chain traceability in big enterprises such as IBM and Walmart [47], the blockchain solutions for supply chain traceability are also extensively studied in academia. On the one hand, the concept of blockchain-based supply chain traceability and the corresponding system design are discussed in many research works [6, 37, 16, 64, 22]. On the other hand, the researchers find blockchain technology can be used together with other technologies, such as the Internet of Things (IoT), to provide the traceability service [67, 66, 60, 41]. However, all these works in industry and academia focus on the design of the traceability system while leaving the efficiency issue alone. In practice, the time- and storage-efficiencies are significantly affected by the ubiquitous IoT devices' considerable and increasing number of product records.

To the best of our knowledge, we present the first work studying high-efficiency blockchain-based supply chain traceability. In particular, we demonstrate the system architecture of a blockchain-based supply chain and model the product records as a directed acyclic graph. In this way, the traceability problem is defined as a graph searching problem over the blockchain. To address the problem, we propose replicating the product records in multiple chunks in a database and developing a novel parallel search algorithm based on the maximum matching algorithm to improve searching efficiency significantly.

The main contributions of this chapter are as follows:

- To the best of our knowledge, we are the first to study and formally model the high-efficiency issue in blockchain-based supply chain traceability.
- We propose a novel parallel search algorithm based on the maximum matching algorithm, which significantly boosts the product tracking efficiency.
- We conduct extensive experiments on the proposed algorithm, which indicates up to 85.1% time reduction for product tracking.

The rest of this chapter is organized as follows. Sec. 4.1 introduces the related work. Sec. 4.2 provides the preliminaries of the problem. In Sec. 4.3, we explain the system model and formally define the problem of high-efficiency blockchain-based supply chain traceability. Sec. 4.4 gives the traditional approach and the proposed algorithm to solving the traceability problem. Sec. 4.5 demonstrates the experimental results. Finally, Sec. 4.6 concludes this chapter.

4.1 Related Work

In this section, we survey the related work about high-efficiency blockchain-based supply chain traceability, i.e., supply chain traceability and searching over blockchain, and articulate the motivations and novelty of this work.

4.1.1 Supply Chain Traceability

The research on supply chain traceability can be roughly divided into two categories, i.e., unified data representation methods for various stakeholders along the supply chain and digital technologies to facilitate reliable and ubiquitous information storage.

A large number of stakeholders along the supply chain have their data management systems with diversified data formats. Supply chain traceability needs to retrieve the data from the stakeholders, and a unified data representation method is demanded. The unified data representation methods for supply chain information have been studied for years. In [10], Bechini et al. investigate the issues for supply chain traceability, introduce a traceability data model and a set of suitable patterns, and discuss the suitable technological standards to define, register, and enable business collaborations, and implement a real-world food supply chain traceability system. In [39], Hu et al. propose a Unified Modeling Language (UML) model for traceability along with a set of suitable patterns, develop a series of UML class diagrams to conceive a method for modeling the product, process, and quality information along the supply chain, and conduct a case study on vegetable supply chain traceability.

In terms of the digital technologies for supply chain traceability, radio-frequency identification (RFID) and blockchain are representative. In particular, RFID is a sensing technology that helps to collect the data along supply chains ubiquitously, while blockchain is a distributed ledger technology to provide secure and reliable data storage services.

The usage of RFID in supply chain traceability can be traced back to as early as 2003 [48], at which time Karkkainen proposed to develop an RFID-based data capture system to solve the problems associated with the logistics of short shelf-life products. In 2007, RFID was widely recognized as a promising technology for supply chain traceability [7, 49] because the passive RFID tags on the products are cheap, do not need to be within the line of sight of the RFID reader (compared with barcodes), and do not need batteries (compared with other sensors). Later on, there are also surveys about RFID-enabled supply chain traceability [65, 79, 23].

The potential of using blockchain technology for supply chain traceability was investigated by Tian in 2016 for the first time [67], in which a traceability system was designed for agri-food supply chains combining the RFID and blockchain technologies. The work is conceptual without real-world deployment but a pioneer. In particular, the product information recorded on the blockchain is immutable, i.e., cannot be modified once stored, making the traceability results reliable. Similar works include [3, 69, 15, 18, 31, 76] in the supply chains of construction, wine, etc., some of which are implemented in real-world settings.

4.1.2 Searching over Blockchain

Blockchain-based supply chain traceability requires the blockchain data to be searched given a product item. We present the related work about searching over blockchain in this subsection. In particular, searching over blockchain refers to the process that the users (with no local storage) request blockchain full nodes (with full storage) to search data on the blockchain, in which the search requests can be keyword search, range query, etc. In literature, integrity, privacy, and efficiency are the three concerned performance metrics of searching over blockchain, in which integrity means whether the searching results are sound and complete, privacy means whether data leakage happens during searching, while efficiency means the time and communication overhead.

The naive procedure of searching over the blockchain is as follows. First, the user sends a searching request to a blockchain full node. Then, the full node proceeds the request by scanning the data on blockchain block by block and transaction by transaction and recording all the data satisfying the searching request. Finally, the full node returns the searching result. As we can see, the integrity of the searching result cannot be guaranteed, the privacy can be disclosed because of the raw data on the blockchain, and the efficiency is low because scanning transactions one by one takes a long time. The research community has been developing solutions to improve integrity, privacy, and efficiency.

Smart contract and verifiable computation are the two approaches to guarantee searching integrity. The basic idea of smart contract is to send the searching requests to all the blockchain nodes rather than a single one. The incentive mechanism of blockchain will motivate the majority of the blockchain nodes to return the sound and complete searching result, which guarantees integrity. The advantage of using smart contract is that the method is general and can be easily adapted to all kinds of data and queries. However, the drawback lies in the high cost of executing smart contracts. In terms of verifiable computation, the searching result returned to the user will be accompanied by proof for integrity verification. Using verifiable computation can fine-tune the efficiency by designing subtle data structure [80, 83, 38, 82, 33, 26]. In contrast, the disadvantage is that there is no general data structure for all types of data and queries.

Searchable encryption is the major approach for privacy preservation during searching over the blockchain. Compared with the naive search approach, the data, queries, and search results are encrypted. The research community has been developing efficient searchable encryption scheme for various types of data and queries [40, 43, 34, 27, 20].

To summarize, existing works about blockchain-based supply chain traceability mainly focus on the system design while leaving the efficiency issue alone. When we reduce blockchain-based supply chain traceability to a problem of searching over blockchain, we find that the reduced graph searching problem on the blockchain is new.

4.2 Preliminaries

This section introduces the preliminary knowledge about blockchain data structure and maximum matching algorithm for the bipartite graphs. Note that the maximum matching algorithm is used for maximizing the parallelization ratio in Sec. 4.4.

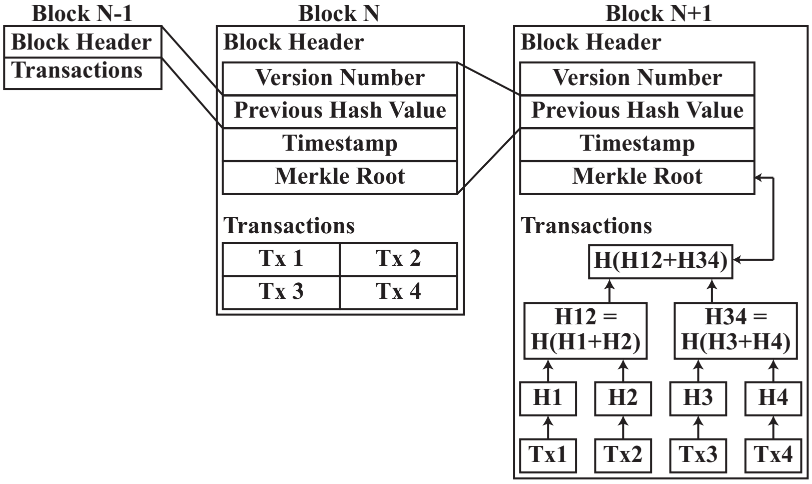

Figure 4.1: Structure of a Typical Blockchain

4.2.1 Blockchain Data Structure

A blockchain is a data structure of an append-only list of blocks linked by cryptographic values, in which each block contains a set of transactions maintained by a decentralized peer-to-peer network. Fig. 4.1 depicts the structure of blockchain with description. To be specific, a single valid block consists of a block header and a list of transactions. The following fields briefly document the block details:

- *Block Header* provides the important information inside the block. Generally speaking, it includes the Version Number, Previous Hash Value, Timestamp, Merkle root, etc. Each block header is hashed, unique, and cryptographically secured, which supports the property of immutability of the blockchain. For example, in Bitcoin [25], "target difficulty" and "nonce" are included as part of the Proof of Work (PoW) consensus algorithm that is used when mining.

- *Version Number* indicates which version of block validation rules to follow. If the block version number differs from other blocks, this block is running on a different chain, commonly known as a hard fork.

- *Previous Hash Value* is a byte field containing the hash of the previous block header, serving as a pointer to the previous block. Such a field ensures that no previous block can be modified without also changing the current block header.

- *Timestamp* is the time of generating this block which is more commonly known as the time when the miner started hashing the current block header. The average block propagation time is calculated based on the timestamp.

- *Merkle root* is derived from the hashes of all transactions included in the current block. It is a tamper resistance measure that those transactions cannot be modified without changing the Merkle Root value, furthermore, the changing of the entire header. Merkle root is also a fast and efficient way to verify the data. In Fig. 4.1, the Merkle root of block $N+1$ is computed as the hierarchical hash results upon the transactions inside.

- *Transactions* contains the transactions broadcast by the nodes to the network and collected into the current block. For example, in Bitcoin [25], a typical transaction references previous transaction outputs as the new transaction inputs and sends Bitcoin values to other addresses as new outputs.

4.2.2 Maximum Matching

In graph theory, a matching in an undirected graph is a set of edges without common vertices. The maximum matching problem is to find a matching that uses as many edges as possible given an undirected graph.

A bipartite graph is a graph whose vertices can be divided into two disjoint and independent sets U and V such that every edge connects a vertex in U and a vertex in V. The maximum matching problem on a bipartite graph is well studied and can be solved efficiently (in polynomial time) using the Hungarian algorithm [52], Ford-Fulkerson algorithm [30], etc.

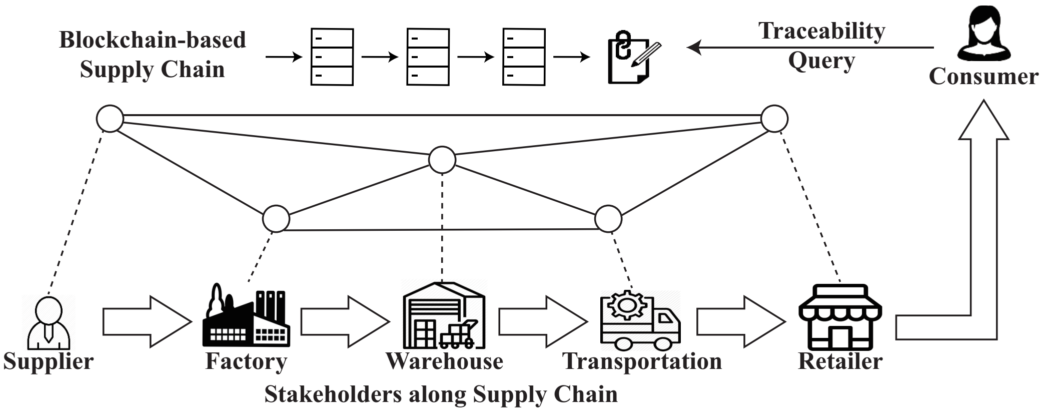

Figure 4.2: System Model of Blockchain-based Supply Chain

4.3 System Model and Problem Definition

In this section, we first give the system model of the blockchain-based supply chain and then formally define the problem of high-efficiency blockchain-based supply chain traceability.

4.3.1 System Model

Fig. 4.2 elaborates the system model of the blockchain-based supply chain. In particular, the stakeholders along the supply chain, e.g., raw material suppliers, factories, warehouses, transportation companies, and retailers, form a peer-to-peer network and maintain a permissioned blockchain. The regulatory authorities can also join and expand the blockchain network. Our system prefers permissioned blockchain to the public one because the nodes not hosted by the supply chain stakeholders should be forbidden by joining the blockchain network. Note that each stakeholder may contribute a set of blockchain nodes, and the whole blockchain network will be of large scale. In our system, the stakeholders will upload the product information to the blockchain motivated by the following reasons. First, the transparent product information on the blockchain will strengthen the consumers' confidence in the products.

Second, the inter-related information helps improve the efficiency of SCM. Finally, the product information will better meet the frequent regulation requests. Note that the blockchain system can only guarantee that the information cannot be tampered with once stored. If a stakeholder provides incorrect records, the blockchain can provide non-tamperable and permanent proof of the incorrectness. In terms of the end consumers, they will enjoy the services provided by the supply chain and query the product tracking information through the blockchain.

The product information recorded on the blockchain will contain at least the following fields:

- Time: the timestamp when the record is submitted.
- Location: the location when the record is submitted.
- Publisher: the one who submitted the record.
- SrcItems: the unique identifiers of the source (original) food items.
- DesItems: the unique identifiers of the destination (result) food items.

The fields of SrcItems and DesItems indicate the relationships among the product. That is, the products in SrcItems are the raw materials of the ones in DesItems. When talking about supply chain traceability, the products in SrcItems should be output if any product in DesItems is set as the input.

4.3.2 Problem Definition

In this section, we define the problem of high-efficiency traceability formally.

Generally speaking, the function of blockchain is to serialize a set of transactions to an ordered list.

Definition 1. *A* ***blockchain*** $\mathcal{B} = (t_1, t_2, \cdots)$ *is defined to be an append-only list of transactions, in which* t_i*s are transactions.*

The transactions t_is in the blockchain are totally ordered, which means t_j is confirmed after t_i for sure if $i < j$.

Definition 2. *In a blockchain, a* ***transaction*** $t_i = (id_i, \mathcal{P}_i)$ *is defined to be a tuple of identifier and direct predecessors, in which* id_i *is the identifier while* $\mathcal{P}_i$ *is the set of identifiers of direct predecessor transactions.*

In the context of traceability, the predecessor means the relationship of dependency, e.g., a bag of potato chips is made from a package of potatoes. Note that for a given transaction t_i, the predecessors in $\mathcal{P}_i$ must be already there in the blockchain, e.g., the transaction of potatoes must appear before the transaction of potato chips in the blockchain. Formally speaking, if $id_j \in \mathcal{P}_i$, then we can infer that $j < i$.

The relationship among the transactions can be represented as a direct acyclic graph (DAG) for better understanding. The construction of the DAG given a blockchain is as follows:

- for each transaction t_i, add a vertex v_i; and
- for each transaction t_i and each identifier $id_j \in \mathcal{P}_i$, add a directed edge from v_j to v_i.

An example set of transactions and its corresponding DAG are shown in Tab. 4.1 and Fig. 4.3, respectively. In the example, the transactions with identifiers 1 and 4 are the direct predecessors of transaction 5. Meanwhile, transaction 3 is a predecessor (indirect) of 5 as in Fig. 4.3. In this chapter, we define *traceability* as a function to find all the predecessors (both direct and indirect) of a given transaction in a given blockchain. The formal definitions of *direct predecessor*, *predecessor* and *traceability* are given are follows.

Table 4.1: Example Transactions in Blockchain

Identifier	Direct Predecessors
1	$\emptyset$
2	$\{1\}$
3	$\{1\}$
4	$\{2, 3\}$
5	$\{1, 4\}$

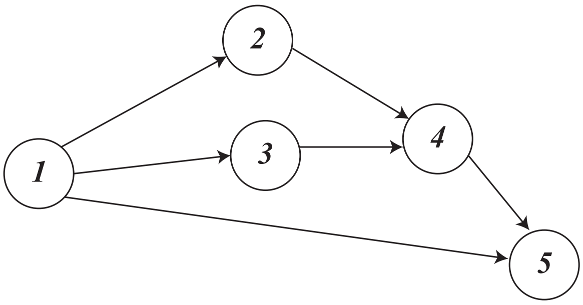

Figure 4.3: Example DAG based on Transactions

Definition 3. *A transaction t_i is defined to be a **direct predecessor** of another transaction t_j if $id_i \in \mathcal{P}_j$.*

Definition 4. *A transaction t_i is defined to be a **predecessor** of another transaction t_j if there is a list of transactions $t_{k_1}, t_{k_2}, \cdots, t_{k_l}$ such that $id_i \in \mathcal{P}_{k_1}$, $id_{k_1} \in \mathcal{P}_{k_2}$, $\cdots$, $id_{k_{l-1}} \in \mathcal{P}_{k_l}$, and $id_{k_l} \in \mathcal{P}_j$.*

In a blockchain, we assume that there is a function called GetPredecessors which takes a transaction identifier as input, and output all the direct predecessors of the input transaction. We assume that GetPredecessors takes $t(n)$ time in which n is the number of transactions in the blockchain. Note that the expression $t(n)$ depends on the implementation of GetPredecessors. For example, if GetPredecessors is implemented using a binary search tree, then $t(n) = O(\log n)$.

Definition 5. *Problem **Traceability**: given a blockchain and a transaction identifier*

id_i, output the identifiers of all the predecessors of t_i.

Table 4.2: Example Input and Output of Traceability

Input	Output
1	$\emptyset$
2	$\{1\}$
3	$\{1\}$
4	$\{1, 2, 3\}$
5	$\{1, 2, 3, 4\}$

Following the definition of *traceability*, if the input is transaction 4, then the output should be transactions 1, 2, and 3. Other examples of input and output can be found in Tab. 4.2.

4.4 Proposed Algorithm & Analysis

In this section, we present the traditional algorithm and the proposed algorithm to solving the problem *traceability*.

4.4.1 Traditional Approach

The naive approach to solving *traceability* is breadth-first search (BFS) as shown in Algo. 1.

In this straightforward solution, the searching is time-consuming for accessing the index and block repeatedly. Detailed speaking, we have a set called All Predecessors, $\mathcal{AP}$, which is the expected output of the given transaction id_i. The $\mathcal{AP}$ is empty at the very beginning. A first-in-first-out queue, Q, is created to hold all the elements that need to be processed. While the Q is not empty, we pick out one element u at a

Algorithm 1 Breadth-first search algorithm to solving the problem *traceability*

```
Input: $\mathcal{B} = (t_1, t_2, \cdots, t_n)$: a blockchain of $n$ transactions; $id$: identifier of a transaction
Output: $\mathcal{AP}$: all the predecessors of $t_i$
$\mathcal{AP} \leftarrow \emptyset$
$Q \leftarrow$ a first-in-first-out queue with a single element $id$
while $Q$ is not empty do
    $u \leftarrow$ PopQueue($Q$)
    $\mathcal{P}_u \leftarrow$ GetPredecessors($u$)
    for each $v \in \mathcal{P}_u$ do
        if $v \notin \mathcal{AP}$ then
            PushQueue($Q$, $v$)
            $\mathcal{AP} \leftarrow \mathcal{AP} \cup \{v\}$
        end if
    end for
end while
return $\mathcal{AP}$
```

time, PopQueue this element u from the queue. The function GetPredecessors is called to get the direct predecessor or predecessors of u and temporarily cached at $\mathcal{P}_u$.

For each element $\mathcal{P}_u$, if it is not in $\mathcal{AP}$, which means it is a new element, $\mathcal{P}_u$ will be PushQueue to the queue Q. At the same time, we update $\mathcal{AP}$ with the new element $\mathcal{P}_u$. If $\mathcal{P}_u$ is already in $\mathcal{AP}$, which means the element has already been processed, no further operation will be needed. This procedure stops when the queue Q is empty, at which time the entire blockchain has been gone through. It is evident that this procedure processes all the elements linearly, one element at a time. Although the breadth-first search-based solution achieves the objective of *traceability*, the efficiency is quite low because only one element can be processed at a time.

4.4.2 Proposed Solution

The critical drawback of the traditional approach lies in the frequent operations of GetPredecessors of high time overhead. To improve the time efficiency, we gain the insight that the operations of GetPredecessors can be parallelized when tracing the transactions. In particular, we use α chunks to store the transactions, which can be accessed in parallel. Each transaction is replicated for $\beta - 1$ times in the chunks to enhance the degree of parallelization.

To store the transactions into α chunks, we need to find a way to evenly distribute all transactions into chunks without causing an imbalance of the chunk storage. Here we propose Algo. 2, a transaction allocation mechanism. We directly modulo each transaction ID with α and store this transaction pair $(id_i, \mathcal{P}_i)$ into the corresponding chunk.

Algorithm 2 Transaction allocation

Input: id_i: identifier of the new transaction; $\mathcal{P}_i$: the set of direct predecessors of the new transaction

Output: The allocation scheme of the new transaction

```
for $i \leftarrow 0$ to $\beta - 1$ do
    Store $(id_i, \mathcal{P}_i)$ in $\mathcal{CK}_{id_i \bmod \alpha}$
end for
```

With transactions evenly allocated to α chunks, the parallelized search algorithm is shown in Algo. 3. To recap, with given transaction ID id_i, we aim at getting all the predecessors $\mathcal{AP}$ of id_i.

The $\mathcal{AP}$ is empty at first, the same with Algo. 1. Then, a set S is created to hold the elements that need to be processed. While the S is not empty, a scheduling algorithm Schedule will be called to generate optimal transaction-chunk pairs $\mathcal{R}$ from the S. This $\mathcal{R}$ contains a list of transaction-chunk pairs which has no conflict with each other. The essential purpose is to process different elements stored at different chunks simultaneously since processing them one by one hinders the searching efficiency, as discussed in Sec. 4.4.1. The details of the procedure SCHEDULE is explained in Algo. 4.

For each pair $(id_i, \mathcal{CK}_i)$ from $\mathcal{R}$, a new thread is forked exclusively to handle this transaction-chunk pair. GETPREDECESSORS will be called to get the direct predecessors of id_i, and id_i will be removed from S. We wait until all the threads, forked from each pair, terminate. For the results returned from each thread, if the element is not in $\mathcal{AP}$ which means it is a new element, the element is added to both S and $\mathcal{AP}$. Otherwise, it is a processed element that needs no further operation. Please note that the condition is whether id is not in $\mathcal{AP}$ since the $\mathcal{AP}$ is the expected result set while the S is a set with elements awaiting to be processed. The $\mathcal{AP}$ is the superset of S. For example, a processed element will be in $\mathcal{AP}$ not S.

Algo. 4 explains the particulars of generating transaction-chunk pairs which can be parallelized from a set of transactions. It is a common bipartite graph maximum matching problem. The input is S, a set of transactions. The expected output is $\mathcal{R}$, the set of transaction-chunk pairs representing the queries that can be parallelized. The $\mathcal{R}$ is empty at the beginning, and G is an empty bipartite graph. At line 3-7, we add vertex v_i to $\mathcal{V}$ for each chunk $\mathcal{CK}_i$. Also, for each transaction belongs to S, we add a vertex u_i to $\mathcal{U}$ representing transaction id_i. Next, from line 8-11, for each given transaction, we first calculate the allocated chunk index based on Algo. 2 and then add an edge to graph G representing the pair of transactions and its allocated chunk index.

Until this step, we have transformed the original transaction data into the graph format in the form of transaction and its corresponding chunk index pairs, stored in G. Then the problem has been modeled as a MAXIMUM-MATCHING problem, which aims to find out the maximum number of edges that share no vertex. We use the Hungarian algorithm to solve the problem. In particular, for a complete bipartite graph G, the Hungarian algorithm finds the maximum-weight matching where sometimes it is also called assignment problem. A bipartite graph can easily

Algorithm 3 Parallelized search algorithm to solving the problem *traceability*

Input: $\mathcal{B} = (t_1, t_2, \cdots, t_n)$: a blockchain of n transactions; id: identifier of a transaction

Output: $\mathcal{AP}$: all the predecessors of t_i

```
$\mathcal{AP} \leftarrow \emptyset$
$S \leftarrow$ a set with a single element $id$
while $S$ is not empty do
    $\mathcal{R} \leftarrow$ SCHEDULE($S$)
    for each $(id_i, \mathcal{CK}_i) \in \mathcal{R}$ do
        fork thread: $\mathcal{P}_i \leftarrow$ GETPREDECESSORS($id_i, \mathcal{CK}_i$)
        $S \leftarrow S \setminus \{id_i\}$
    end for
    Wait until all the threads terminate
    for each $\mathcal{P}_i$ returned by the threads do
        for each $id \in \mathcal{P}_i$ do
            if $id \notin \mathcal{AP}$ then
                $S \leftarrow S \cup \{id\}$
                $\mathcal{AP} \leftarrow \mathcal{AP} \cup \{id\}$
            end if
        end for
    end for
end while
return $\mathcal{AP}$
```

be represented by an adjacency matrix, where the weights of edges are the entries. The method operates on this key idea: if a number is added to or subtracted from all of the entries of any one row or column of a cost matrix, then an optimal assignment for the resulting cost matrix is also an optimal assignment for the original cost matrix. Based on this MAXIMUM-MATCHING Algorithm, we get the result of $\mathcal{R}'$, which is the edge set of non-conflict edges. At line 14-16, we transform the returned eligible edges back into $(id_i, \mathcal{CK}_i)$ key pairs. Finally, we return the result $\mathcal{R}$ which is the input of line 5 at Algo. 3.

Algorithm 4 Procedure SCHEDULE as in Algo. 3 to Generate Parallelized Query

Input: S: a set of transactions

Output: $\mathcal{R}$: a set of transaction-chunk pairs representing the queries that can be parallelized

```
 1: $\mathcal{R} \leftarrow \emptyset$
 2: $G \leftarrow$ an empty bipartite graph with vertex sets $\mathcal{U}$ and $\mathcal{V}$, and edge set $\mathcal{E}$
 3: for $i \leftarrow 0$ to $\alpha - 1$ do
 4:     Add a vertex $v_i$ to $\mathcal{V}$ representing chunks $\mathcal{CK}_i$
 5: end for
 6: for $id_i \in S$ do
 7:     Add a vertex $u_i$ to $\mathcal{U}$ representing transaction $id_i$
 8:     for $i \leftarrow 1$ to $\beta$ do
 9:         $j \leftarrow id_i \bmod \alpha$
10:         Add an edge $(u_i, v_j)$ to $\mathcal{E}$
11:     end for
12: end for
13: $\mathcal{R}' \leftarrow$ MAXIMUM-MATCHING($\mathcal{G}$)
14: for each $(u_i, v_j) \in \mathcal{R}'$ do
15:     $\mathcal{R} \leftarrow \mathcal{R} \cup \{(id_i, \mathcal{CK}_j)\}$
16: end for
17: return $\mathcal{R}$
```

4.4.3 Time Complexity Analysis

In this subsection, we analyze and compare the time complexities of Algo. 1 and Algo. 3.

We assume the number of returned predecessors to be m, i.e., $\mathcal{AP} = m$. The time complexity of Algo. 1 is $O(m \log m + mt(n))$ when maintaining the set $\mathcal{AP}$ and invoking the procedure GetPredecessors for m times.

Because Algo. 4 is a function called by Algo. 3, we analyze the time complexity of Algo. 4 first. Algo. 4 constructs a graph and run the Maximum-Matching algorithm. Note that the time complexity of the Maximum-Matching algorithm is $O(V \cdot E)$, in which V and E are the numbers of vertices and edges of the graph, respectively [52]. When constructing the graph, α vertices are added from line 3 to 5, and $|S|$ vertices and $|S| \cdot \beta$ edges are added from line 6 to 12. Here, $|S| = O(m)$ because S from Algo. 3 is a subset of $\mathcal{AP}$. Therefore, the number of vertices and edges are $O(\alpha + m)$ and $O(m\beta)$, respectively. To this end, the time complexity of Algo. 4 is $O((\alpha + m)m\beta)$. Because α is a constant compared to m, the time complexity is reduced to $O(\beta \cdot m^2)$.

In Algo. 3, we also assume that p transactions are searched in parallel at line 6 on average. We find that the main loop from line 3 to 18 is entered for $\frac{m}{p}$ times. Inside the main loop, line 4 takes $O(\beta \cdot m^2)$ as analyzed previously and line 5-9 takes $O(p + t(n) + \log m)$ time, reduced to $O(t(n) + \log m)$ because p is minor compared to $t(n)$ and $\log m$. As a result, the main loop takes $O(\frac{m}{p} \cdot (\beta \cdot m^2 + t(n) + \log m)) = O(\frac{\beta m^3}{p} + \frac{mt(n)}{p})$ because $\log m$ is minor compared to $\beta \cdot m^2$, excluding line 10-17. In terms of line 10-17, it takes $O(m \log m)$ in total because its purpose is to maintain two sets S and $\mathcal{AP}$, both of which are of size $O(m)$. As a result, the overall time complexity of Algo. 3 is $O(m \log m + \frac{\beta m^3}{p} + \frac{mt(n)}{p}) = O(\frac{\beta m^3}{p} + \frac{mt(n)}{p})$ because $m \log m$ is minor compared to $\beta m^3/p$.

Next, we compare the time complexities of Algo. 1 and Algo. 3, which are $O(m \log m +$

$mt(n))$ and $O(\frac{\beta m^3}{p} + \frac{mt(n)}{p})$, respectively. If $t(n)$ dominates the time complexity compared to m (for example, n is large), Algo. 3 will take much less time than Algo. 1 theoretically. Note that Algo. 3 achieves better performance with the sacrifice in higher storage overhead ($\beta - 1$ replicas of the transactions in α trunks).

4.5 Experimental Results & Discussion

In this section, we demonstrate the effectiveness and practicability of the proposed high-efficiency traceability solution based on implementation on Hyperledger Fabric [5] and extensive experiments evaluating the parallelization ratio and storage ratio. We also discuss the transaction allocation algorithm and the database selection, which might affect the tracing efficiency in this work.

4.5.1 Experimental Environments & Design

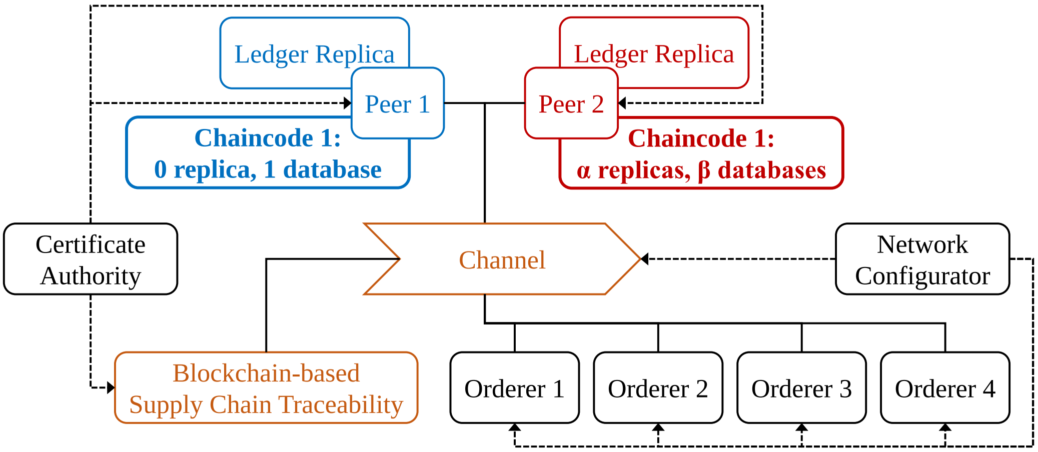

Figure 4.4: System Architecture

In this chapter, we leverage Hyperledger Fabric, an open-source permissioned blockchain platform, to implement and evaluate our proposed algorithms. Fig. 4.4 depicts the architecture of the developed system using Hyperledger Fabric, showing the components and their relationship as follow:

- The benchmark algorithm, i.e., BFS-based solution when $\alpha = 0$ and $\beta = 1$, is implemented in “Chaincode 1”, hosted by “Peer 1”.
- The proposed algorithm in this paper with varying parameters α and β is implemented in “Chaincode 2”, hosted by “Peer 2”.
- The two chain codes, i.e., “Chaincode 1” and “Chaincode 2”, are deployed on the “Channel”, supporting the application “Blockchain-based Supply Chain Traceability”.
- The transactions in “Channel” are ordered by four orderers, i.e., “Orderer 1”, “Orderer 2”, “Orderer 3”, and “Orderer 4”, running the crash fault-tolerant consensus protocol as provided by Hyperledger Fabric.
- The “Certificate Authority” dispenses identities to the application and two peers.
- The “Network Configurator” configures the networks of the channel and four orderers.

We deploy a prototype based on the system architecture using eight workstations. Each component is hosted on a workstation consisting of 4 core 8 threads of Intel Core i7-8809G 4.2Ghz CPU with 32GB of DDR4 DRAM and 1024GB of NVMe SSD, running on Ubuntu 20.04. The workstations are connected in a local area network and form a blockchain network. The deployment implies the practicability of the proposed high-efficiency traceability solution.

With the prototype system, we study how the proposed solution performs when compared with the BFS-based solution. Based on Algo. 2, Algo. 3, and Algo. 4 mentioned in Sec. 4.4.2, there are three important variables that will impact the efficiency of proposed solution: n, α, and β, representing the number of transactions, the number of chunks, and the number of replicas, respectively.

In order to reduce the impact of n on the system, we use *parallelization ratio* and *storage ratio* as two key performance metrics for demonstrating the solution effectiveness. The *parallelization ratio* and *storage ratio* are the execution time overhead and chunks storage overhead compared to BFS-based solution, respectively. In the following experiments, we will look into the parallelization ratio, storage ratio with different combinations of α and β. We will discuss the transaction allocation algorithm and selection of the database as well. For each experiment, we repeat the experiment 50 times to get the average results.

4.5.2 Evaluation of Parallelization Ratio

In this experiment, we will compare the parallelization ratio of the proposed parallelization algorithm. We try to find the pair of optimal parameters of α and β by changing their values, calculated by the number of operations. The algorithm has such termination condition that it will be terminated by final results where the transaction has no father nodes or is the genesis/origin transaction. Intuitively, with the increase of α (the number of chunks) or β (the number of replicas), the parallelization ratio shall also rise compared to the straightforward BFS-based solution.

Fig. 4.5 and Fig. 4.6 depict the change of parallelization ratio with 1 to 4 replicas and 5 to 9 replicas respectively. Note that 0 replica and 1 chunk indicate the BFS-based solution. Obviously, when the number of chunks is fixed, with the increase of replicas, the parallelization ratio increases dramatically.

Such a trend is maintained when the number of chunks is small. Precisely speaking, the turning point slightly shifts towards the right with the increase of replicas. For example, When β equals 2, the turning point of α is around 6. When β equals 6, the turning point of α is around 12. For the surge part, the reason is that when the

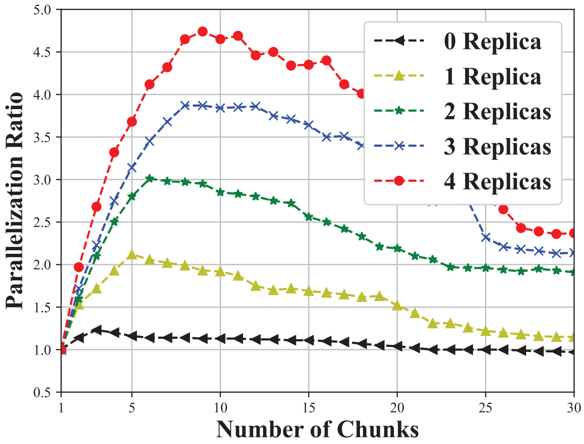

Figure 4.5: Parallelization Ratio with 0-4 Replicas

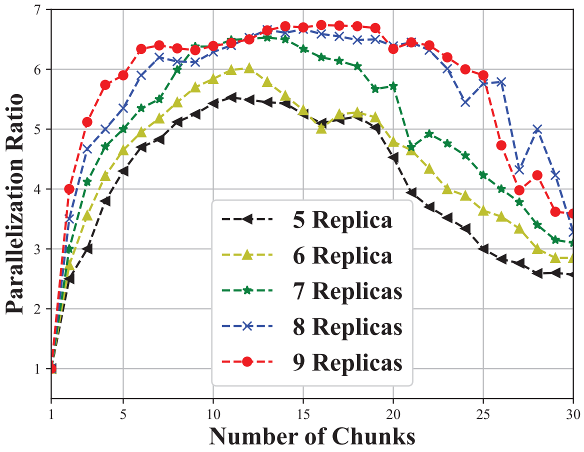

Figure 4.6: Parallelization Ratio with 5-9 Replicas

number of replicas are arising, each transaction has a higher probability of coverage to store it into different chunks. After the turning point, the curve comes down moderately. This is because the increasing number of chunks dilutes the probability of finding the exact transaction among chunks. In extreme cases, when having 8 and 9 replicas, the ratio becomes unstable at the end. The relatively large number of replicas distributed among chunks increases the complexity of finding the target transaction. Fig. 4.7 depicts the maximum parallelization ratio that can be achieved based on the different number of replicas. The growth slows down when the system has more than six replicas.

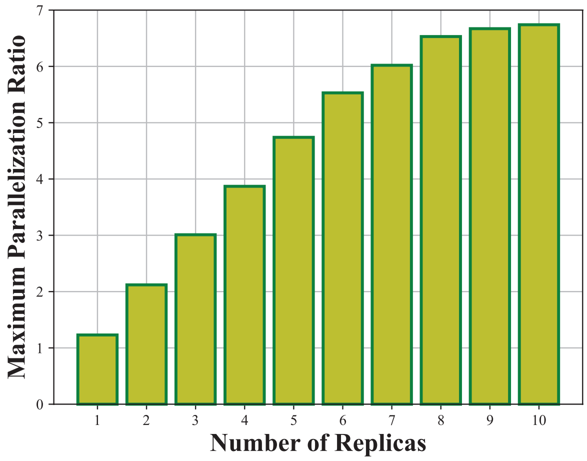

Figure 4.7: Maximum Parallelization Ratio

In Sec. 4.5.2, we plan to find out the optimal pair of α and β via changing their values. For each β, there should be an optimal value of α which should be noticed at the curve's apex. Fig. 4.8 illustrates the relationship between the number of chunks to achieve max parallelization ratio with the number of replicas. It is not hard to find that such a linear relationship maintains steadily with the increase of replicas. By simple linear regression, we can get $f(\alpha) = 1.43\beta + 1.93$. It implies that the number of chunks should be neither too small nor too large, given the number of replicas. An excess number of chunks does not contribute to the parallelization ratio, which has an upper limit as shown in Fig. 4.7. The "sweet point" of the α and β can be easily calculated, which will be helpful when we have more replicas.

4.5.3 Evaluation of Storage Ratio

In this set of experiments, we will investigate the database storage overhead of the proposed parallelization algorithm. We will fix the value of α and change the value of β. Then we do it in a reverse way by fixing the β and changing the α. We normalized the storage cost of the BFS-based solution as 1 for easier comparison.

Fig. 4.9 and Fig. 4.10 illustrate the change of storage overhead ratio with different

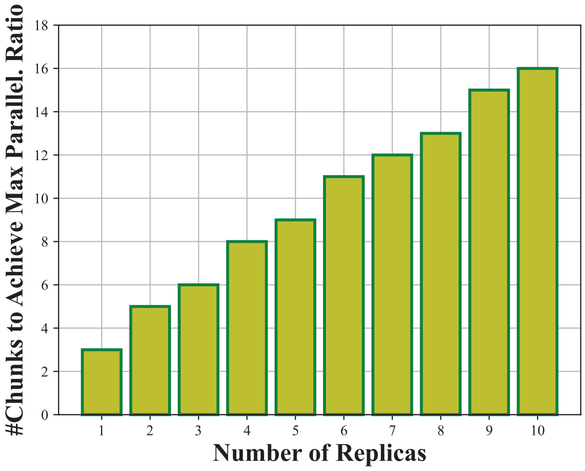

Figure 4.8: Number of Chunks to Achieve Max Parallelization Ratio

settings of the number of chunks and replicas. The results of the experiment are straightforward, largely affected by the value β, the number of replicas. With more replicas available, the storage ratio grows steadily. On the other hand, α, i.e., the number of chunks, affects little on the storage ratio, with a slight increase when more chunks are used. The reason is that more chunks require more complexity in building the index. Overall, the storage overhead climbs slowly compared with the BFS-based solution, which is acceptable to our concern.

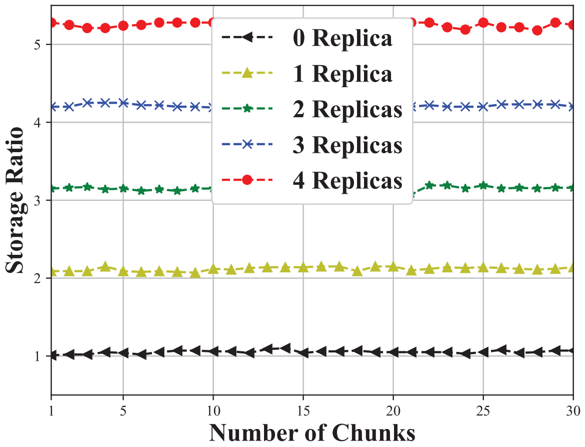

Figure 4.9: Database Storage Ratio with 0-4 Replicas

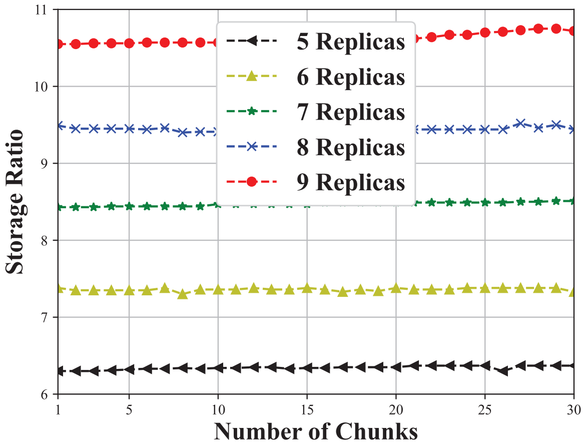

Figure 4.10: Database Storage Ratio with 5-9 Replicas

4.5.4 Discussion of Transaction Allocation Algorithm

In Sec. 4.4.2, we point out that the major drawback of the traditional approach is the frequent operations of GetPredecessors of high time overhead. Based on this insight, we propose using α chunks to store the transactions, which can dramatically improve the parallelized operations when tracing the transactions. First, we propose a straightforward transaction allocation mechanism with mod operation. However, the issue of imbalance may still occur for many reasons. For example, the uneven distribution of transaction ID can be found in many statistics reports (add ref, the law of large numbers), which cannot be directly solved using mod function.

To improve the randomness of the transaction allocation of Algo. 2, we proposed a new random-based mechanism as shown in Algo. 5. The main difference with the Algo. 2 is that before mod each transaction ID with α, we pre-process the transaction ID with a random function f_i, which eliminates the deterministic correlation between transaction ID and the target allocated chunk.

We conduct experiments to compare the degree of transaction distribution for Algo. 2 and Algo. 5. Both transaction allocation mechanisms achieve the parallelization target. The parallelization ratio in Fig. 4.5 and Fig. 4.6 evidences the idea that allocating

Algorithm 5 Transaction allocation (random method)

Input: id_i: identifier of the new transaction; $\mathcal{P}_i$: the set of direct predecessors of the new transaction

Output: The allocation scheme of the new transaction

```
for $i \leftarrow 1$ to $\beta$ do
    Store $(id_i, \mathcal{P}_i)$ in $\mathcal{CK}_{f_i(id_i) \bmod \alpha}$
end for
```

transactions into different chunks can parallel the operations to speed up the tracing. The selection of either mechanism does not affect the parallelization ratio and the storage ratio.

4.5.5 Discussion of Database Selection

Blockchain can be considered as a multi-node database maintained by a network of independent participants. It is decentralized, with no single user having the ultimate authority over the system. On the other hand, the database, unlike blockchains, are a centralized ledger that an administrator runs. Although blockchain looks contradictory to the database, they are closely connected. Conceptually, as a whole, a blockchain is distributed across the entire network peers. Fundamentally, for a single node of the network, it still relies on a specific database to maintain its local ledger, which is synchronized from peers, for verification and synchronization purposes. For example, the Bitcoin core client uses *LevelDB* database for the block index and the chain state, which is also known as the UTXO (Unspent Transaction Output) set. The Bitcoin blocks are dumped in raw on the disk without being converted and imported into another database. Ethereum uses LevelDB as well. LevelDB is also the default key-value state database embedded in Hyperledger Fabric, while *CouchDB* is a choice as it supports rich queries and indexing for more efficient queries over large datasets. During our experiments, the selection of the database does not impact the parallelization ratio. The proposed solution suits different databases with proper interface and block parser.

4.6 Chapter Summary

In this chapter, we study the problem of high-efficiency supply chain traceability. First, we depict the system model and formally define the traceability problem as a graph searching problem. Then, a parallel searching algorithm is proposed, in which the maximum flow theory is used to maximize the parallelization ratio. The experimental results show up to 85.1% reduction of the product tracking time.

In the future, we will study the algorithm to further boost the time efficiency by considering the sequence of parallel searches in blockchain-based supply chain traceability. Moreover, We will also consider integrating the real-world blockchain-based supply chain system with the proposed tracing solution for a more significant impact.

Chapter 5

Conclusion and Future Directions

Supply chain management contributes to remarkable market value, playing a vital role in the global economy. However, the underlying technologies are underdeveloped, especially from the computer science perspective. In particular, the data from various supply chain stakeholders are not interoperable, leading to high operation costs. Moreover, the traceability service is not provided in most modern supply chains, which brings severe concerns in terms of product quality.

This thesis employs the latest blockchain technology in supply chain management and develops blockchain-based supply chain management, connecting various supply chain stakeholders and providing traceability services. Product information stored on the blockchain cannot be tampered with once stored. Moreover, the product records can be authenticated without centralized authorities. We have developed novel methods of big data sharing and high-efficiency traceability for the supply chain management.

The future directions are twofold as follows. On the one hand, this thesis only presents prototypes of supply chain management, and real-world deployment needs to be done to examine our solution. The practical and technical challenges may be discovered through large-scale deployment. On the other hand, we will explore methodologies, approaches, and mechanisms to improve the quality of supply chain services.

References

[1] Ckan: Comprehensive knowledge archive network, the open-source data portal platform. `http://ckan.org`. Online; Accessed: 2022-07-12.

[2] Zenodo: A research data repository. `https://zenodo.org`. Online; Accessed: 2022-07-12.

[3] Saveen A Abeyratne and Radmehr P Monfared. Blockchain ready manufacturing supply chain using distributed ledger. *International Journal of Research in Engineering and Technology*, 5(9):1–10, 2016.

[4] Naif Alzahrani and Nirupama Bulusu. Block-supply chain: A new anti-counterfeiting supply chain using NFC and blockchain. In *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, CryBlock'18, pages 30–35, 2018.

[5] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the thirteenth EuroSys conference*, pages 1–15, 2018.

[6] Shireesh Apte and Nikolai Petrovsky. Will blockchain technology revolutionize excipient supply chain management? *Journal of Excipients and Food Chemicals*, 7(3):910, 2016.

[7] Mohsen Attaran. Rfid: an enabler of supply chain operations. *Supply Chain Management: An International Journal*, 12(4):249–257, 2007.

[8] Michael Batty. Big data, smart cities and city planning. *Dialogues in Human Geography*, 3(3):274–279, 2013.

[9] Benita M Beamon. Supply chain design and analysis: Models and methods. *International journal of production economics*, 55(3):281–294, 1998.

[10] Alessio Bechini, Mario GCA Cimino, Francesco Marcelloni, and Andrea Tomasi. Patterns and technologies for enabling supply chain traceability through collaborative e-business. *Information and Software Technology*, 50(4):342–359, 2008.

[11] Juan Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.

[12] Jonathan Biggs, Sheri R Hinish, Michael A Natale, and Matt Patronick. Blockchain: Revolutionizing the global supply chain by building trust and transparency. *Rutgers University, New Jersey*, 2018.

[13] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. Deanonymisation of clients in bitcoin P2P network. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 15–29, 2014.

[14] Kamanashis Biswas, Vallipuram Muthukkumarasamy, and Wee Lum. Blockchain based wine supply chain traceability system. In *Future Technologies Conference*, nov 2017.

[15] Kamanashis Biswas, Vallipuram Muthukkumarasamy, and Wee Lum Tan. Blockchain based wine supply chain traceability system. In *Future Technologies Conference (FTC) 2017*, pages 56–62. The Science and Information Organization, 2017.

[16] Gregor Blossey, Jannick Eisenhardt, and Gerd Hahn. Blockchain technology in supply chain management: an application perspective. In *Proceedings of the 52nd Hawaii international conference on system sciences*, pages 1–9. ScholarSpace, 2019.

[17] Thomas Bocek, Bruno Bastos Rodrigues, Tim Strasser, and Burkhard Stiller. Blockchains everywhere - a use-case of blockchains in the pharma supply-chain. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, May 8-12, 2017*, pages 772–777, 2017.

[18] Miguel Pincheira Caro, Muhammad Salek Ali, Massimo Vecchio, and Raffaele Giaffreda. Blockchain-based traceability in agri-food supply chain management: A practical implementation. In *2018 IoT Vertical and Topical Summit on Agriculture-Tuscany (IOT Tuscany)*, pages 1–4. IEEE, 2018.

[19] Shuchih E Chang and Yichian Chen. When blockchain meets supply chain: A systematic literature review on current development and potential applications. *IEEE Access*, 8:62478–62494, 2020.

[20] Lanxiang Chen, Wai-Kong Lee, Chin-Chen Chang, Kim-Kwang Raymond Choo, and Nan Zhang. Blockchain based searchable encryption for electronic health record sharing. *Future Generation Computer Systems*, 95:420–429, 2019.

[21] Cheng-Kang Chu, Sherman SM Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H Deng. Key-aggregate cryptosystem for scalable data sharing in cloud storage. *IEEE transactions on parallel and distributed systems*, 25(2):468–477, 2014.

[22] Rosanna Cole, Mark Stevenson, and James Aitken. Blockchain technology: implications for operations and supply chain management. *Supply Chain Management: An International Journal*, 24(4):469–483, 2019.

[23] Corrado Costa, Francesca Antonucci, Federico Pallottino, Jacopo Aguzzi, David Sarriá, and Paolo Menesatti. A review on agri-food supply chain traceability by means of rfid technology. *Food and bioprocess technology*, 6(2):353–366, 2013.

[24] Baojiang Cui, Zheli Liu, and Lingyu Wang. Key-aggregate searchable encryption (kase) for group data sharing via cloud storage. *IEEE Transactions on computers*, 65(8):2374–2385, 2016.

[25] Michael A Cusumano. The bitcoin ecosystem. *Communications of the ACM*, 57(10):22–24, 2014.

[26] Xiaohai Dai, Jiang Xiao, Wenhui Yang, Chaofan Wang, Jian Chang, Rui Han, and Hai Jin. Lvq: A lightweight verifiable query approach for transaction history in bitcoin. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pages 1020–1030. IEEE, 2020.

[27] Yicheng Ding, Wei Song, and Yuan Shen. Enabling efficient multi-keyword search over fine-grained authorized healthcare blockchain system. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, pages 27–41. Springer, 2020.

[28] Mario Dobrovnik, David M Herold, Elmar Fürst, and Sebastian Kummer. Blockchain for and in logistics: What to adopt and where to start. *Logistics*, 2(3):18, 2018.

[29] Sunil Erevelles, Nobuyuki Fukawa, and Linda Swayne. Big data consumer analytics and the transformation of marketing. *Journal of Business Research*, 69(2):897–904, 2016.

[30] Lester Randolph Ford and Delbert Ray Fulkerson. *Flows in networks.* Princeton university press, 2015.

[31] Kristoffer Francisco and David Swanson. The supply chain has no clothes: Technology adoption of blockchain for supply chain transparency. *Logistics*, 2(1):2, 2018.

[32] Kristoffer Francisco and Rodger Swanson. The supply chain has no clothes: Technology adoption of blockchain for supply chain transparency. *Logistics*, 2:2, 01 2018.

[33] Qingxing Guo, Sijia Deng, Lei Cai, Yanchao Zhu, Zhao Zhang, and Cheqing Jin. Blockchain pg: Enabling authenticated query and trace query in database. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, pages 529–534. Springer, 2020.

[34] Yu Guo, Chen Zhang, and Xiaohua Jia. Verifiable and forward-secure encrypted search using blockchain techniques. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2020.

[35] Amulya Gurtu and Jestin Johny. Potential of blockchain technology in supply chain management: a literature review. *International Journal of Physical Distribution & Logistics Management*, 49(9):881–900, 2019.

[36] Niels Hackius and Moritz Petersen. Blockchain in logistics and supply chain : trick or treat? *Proceedings of the Hamburg International Conference of Logistics (HICL)*, pages 3–18, 2017.

[37] Niels Hackius and Moritz Petersen. Blockchain in logistics and supply chain: trick or treat? In *Proceedings of the Hamburg International Conference of Logistics (HICL)*, volume 23, pages 3–18, 2017.

[38] Kun Hao, Junchang Xin, Zhiqiong Wang, and Guoren Wang. Outsourced data integrity verification based on blockchain in untrusted environment. *World Wide Web*, 23(4):2215–2238, 2020.

[39] Jinyou Hu, Xu Zhang, Liliana Mihaela Moga, and Mihaela Neculita. Modeling and implementation of the vegetable supply chain traceability system. *Food Control*, 30(1):341–353, 2013.

[40] Shengshan Hu, Chengjun Cai, Qian Wang, Cong Wang, Xiangyang Luo, and Kui Ren. Searching an encrypted cloud meets blockchain: A decentralized, reliable and fair realization. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 792–800. IEEE, 2018.

[41] Srinivas Jangirala, Ashok Kumar Das, and Athanasios V Vasilakos. Designing secure lightweight blockchain-enabled rfid-based authentication protocol for supply chains in 5g mobile edge computing environment. *IEEE Transactions on Industrial Informatics*, 16(11):7081–7093, 2019.

[42] Shan Jiang, Jiannong Cao, Yang Liu, Jinlin Chen, and Xuefeng Liu. Programming large-scale multi-robot system with timing constraints. In *Computer Communication and Networks (ICCCN), 2016 25th International Conference on*, pages 1–9. IEEE, 2016.

[43] Shan Jiang, Jiannong Cao, Julie A McCann, Yanni Yang, Yang Liu, Xiaoqing Wang, and Yuming Deng. Privacy-preserving and efficient multi-keyword search over encrypted data on blockchain. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 405–410. IEEE, 2019.

[44] Shan Jiang, Jiannong Cao, Hanqing Wu, and Yanni Yang. Fairness-based packing of industrial iot data in permissioned blockchains. *IEEE Transactions on Industrial Informatics*, 2020.

[45] Shan Jiang, Jiannong Cao, Hanqing Wu, Yanni Yang, Mingyu Ma, and Jianfei He. Blochie: a blockchain-based platform for healthcare information exchange. In *Smart Computing (SMARTCOMP), 2018 IEEE International Conference on (to appear)*, pages 1–8. IEEE, 2018.

[46] Shan Jiang, Jiannong Cao, Juncen Zhu, and Yinfeng Cao. Polychain: a generic blockchain as a service platform. In *International Conference on Blockchain and Trustworthy Systems (Blocksys' 2021)*, pages 1–14, 2021.

[47] Reshma Kamath. Food traceability on blockchain: Walmart's pork and mango pilots with ibm. *The Journal of the British Blockchain Association*, 1(1):3712, 2018.

[48] Mikko Kärkkäinen. Increasing efficiency in the supply chain for short shelf life goods using rfid tagging. *International Journal of Retail & Distribution Management*, 31(10):529–536, 2003.

[49] Thomas Kelepouris, Katerina Pramatari, and Georgios Doukidis. Rfid-enabled traceability in the food supply chain. *Industrial Management & data systems*, 2007.

[50] Kari Korpela, Jukka Hallikas, and Tomi Dahlberg. Digital supply chain transformation toward blockchain integration. In *50th Hawaii International Conference on System Sciences, HICSS 2017, Hilton Waikoloa Village, Hawaii, USA, January 4-7, 2017*, pages 1–10, 2017.

[51] Nir Kshetri. 1 blockchain's roles in meeting key supply chain management objectives. *International Journal of Information Management*, 39:80–89, 2018.

[52] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[53] David Lazer, Ryan Kennedy, Gary King, and Alessandro Vespignani. The parable of google flu: traps in big data analysis. *Science*, 343(6176):1203–1205, 2014.

[54] Dongmyung Lee and Jinwoo Park. Rfid-based traceability in the supply chain. *Industrial Management & Data Systems*, 108(6):713–725, 2008.

[55] Kaijun Leng, Ya Bi, Linbo Jing, Han-Chi Fu, and Inneke Van Nieuwenhuyse. Research on agricultural supply chain system with double chain architecture based on blockchain technology. *Future Generation Computer Systems*, 86:641–649, 2018.

[56] Wengen Li, Jiannong Cao, Jihong Guan, Man Lung Yiu, and Shuigeng Zhou. Efficient retrieval of bounded-cost informative routes. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2182–2196, 2017.

[57] Xiulong Liu, Keqiu Li, Geyong Min, Yanming Shen, Alex X Liu, and Wenyu Qu. Completely pinpointing the missing rfid tags in a time-efficient way. *IEEE Transactions on Computers*, 64(1):87–96, 2015.

[58] Damiano Di Francesco Maesa, Paolo Mori, and Laura Ricci. Blockchain based access control. In *Distributed Applications and Interoperable Systems - 17th IFIP WG 6.1 International Conference, DAIS 2017*, pages 206–220, 2017.

[59] John T Mentzer, William DeWitt, James S Keebler, Soonhong Min, Nancy W Nix, Carlo D Smith, and Zach G Zacharia. Defining supply chain management. *Journal of Business logistics*, 22(2):1–25, 2001.

[60] Saikat Mondal, Kanishka P Wijewardena, Saranraj Karuppuswami, Nitya Kriti, Deepak Kumar, and Premjeet Chahal. Blockchain inspired rfid-based information architecture for food supply chain. *IEEE Internet of Things Journal*, 6(3):5803–5813, 2019.

[61] Russell A Poldrack and Krzysztof J Gorgolewski. Making big data open: data sharing in neuroimaging. *Nature neuroscience*, 17(11):1510, 2014.

[62] Maciel M Queiroz, Renato Telles, and Silvia H Bonilla. Blockchain and supply chain management integration: a systematic review of the literature. *Supply Chain Management: An International Journal*, 25(2):241–254, 2018.

[63] Wullianallur Raghupathi and Viju Raghupathi. Big data analytics in healthcare: promise and potential. *Health information science and systems*, 2(1):3, 2014.

[64] Sara Saberi, Mahtab Kouhizadeh, Joseph Sarkis, and Lejia Shen. Blockchain technology and its relationships to sustainable supply chain management. *International Journal of Production Research*, 57(7):2117–2135, 2019.

[65] Aysegul Sarac, Nabil Absi, and Stéphane Dauzère-Pérès. A literature review on the impact of rfid technologies on supply chain management. *International journal of production economics*, 128(1):77–95, 2010.

[66] Michail Sidorov, Ming Tze Ong, Ravivarma Vikneswaren Sridharan, Junya Nakamura, Ren Ohmura, and Jing Huey Khor. Ultralightweight mutual authentication rfid protocol for blockchain enabled supply chains. *IEEE Access*, 7:7273–7285, 2019.

[67] Feng Tian. An agri-food supply chain traceability system for china based on rfid & blockchain technology. In *2016 13th international conference on service systems and service management (ICSSSM)*, pages 1–6. IEEE, 2016.

[68] Feng Tian. A supply chain traceability system for food safety based on haccp, blockchain & internet of things. In *2017 International Conference on Service Systems and Service Management*, pages 1–6. IEEE, 2017.

[69] Feng Tian. A supply chain traceability system for food safety based on haccp, blockchain & internet of things. In *2017 International conference on service systems and service management*, pages 1–6. IEEE, 2017.

[70] Kentaroh Toyoda, P Takis Mathiopoulos, Iwao Sasase, and Tomoaki Ohtsuki. A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain. *IEEE Access*, 5:17465–17477, 2017.

[71] Daniel Tse, Bowen Zhang, Yuchen Yang, Chenli Cheng, and Haoran Mu. Blockchain application in food supply information security. In *2017 IEEE international conference on industrial engineering and engineering management (IEEM)*, pages 1357–1361. IEEE, 2017.

[72] Jen-Hung Tseng, Yen-Chih Liao, Bin Chong, and Shih-wei Liao. Governance on the drug supply chain via gcoin blockchain. *International Journal of Environmental Research and Public Health*, 15:1055–1063, 2018.

[73] Jelle van den Hooff, M Frans Kaashoek, and Nickolai Zeldovich. Versum: Verifiable computations over large public logs. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 1304–1316. ACM, 2014.

[74] Matthew A Waller and Stanley E Fawcett. Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management. *Journal of Business Logistics*, 34(2):77–84, 2013.

[75] Samuel Fosso Wamba and Maciel M Queiroz. Blockchain in the operations and supply chain management: Benefits, challenges and future research opportunities. *International Journal of Information Management*, 52:102064, 2020.

[76] Zhaojing Wang, Tengyu Wang, Hao Hu, Jie Gong, Xu Ren, and Qiying Xiao. Blockchain-based framework for improving supply chain traceability and information sharing in precast construction. *Automation in Construction*, 111:103063, 2020.

[77] Hanqing Wu, Jiannong Cao, Shan Jiang, Ruosong Yang, Yanni Yang, and Jianfei Hey. Tsar: a fully-distributed trustless data sharing platform. In *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 350–355. IEEE, 2018.

[78] Hanqing Wu, Jiannong Cao, Yanni Yang, Cheung Leong Tung, Shan Jiang, Bin Tang, Yang Liu, Xiaoqing Wang, and Yuming Deng. Data management in supply chain using blockchain: Challenges and a case study. In *2019 28th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–8. IEEE, 2019.

[79] Yanbo Wu, Damith C Ranasinghe, Quan Z Sheng, Sherali Zeadally, and Jian Yu. Rfid enabled traceability networks: a survey. *Distributed and Parallel Databases*, 29(5):397–443, 2011.

[80] Cheng Xu, Ce Zhang, and Jianliang Xu. vchain: Enabling verifiable boolean range queries over blockchain databases. In *Proceedings of the 2019 international conference on management of data*, pages 141–158, 2019.

[81] Lei Xu, Lin Chen, Zhimin Gao, Yang Lu, and Weidong Shi. Coc: Secure supply chain management system based on public ledger. In *26th International Conference on Computer Communication and Networks, ICCCN 2017, Vancouver, BC, Canada, Jul 31 - Aug 3, 2017*, pages 1–6, 2017.

[82] Ce Zhang, Cheng Xu, Haixin Wang, Jianliang Xu, and Byron Choi. Authenticated keyword search in scalable hybrid-storage blockchains. In *2021 IEEE 37th international conference on data engineering (ICDE)*. IEEE, 2021.

[83] Ce Zhang, Cheng Xu, Jianliang Xu, Yuzhe Tang, and Byron Choi. Gemˆ 2-tree: A gas-efficient structure for authenticated range queries in blockchain. In *2019 IEEE 35th international conference on data engineering (ICDE)*, pages 842–853. IEEE, 2019.

[84] Guy Zyskind, Oz Nathan, and Alex 'Sandy' Pentland. Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE Security and Privacy Workshops (SPW)*, pages 180–184, Los Alamitos, CA, USA, may 2015. IEEE Computer Society.