Blockchain-Coordinated Frameworks for Scalable and Secure Supply Chain Networks

Aaliya Sarfaraz

A thesis in fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Engineering & Information Technology

Faculty of Engineering

The University of New South Wales

February 2023

Welcome to the Research Alumni Portal, Aaliya Sarfaraz!

You will be able to download the finalised version of all thesis submissions that were processed in GRIS here.

Please ensure to include the **completed declaration** (from the Declarations tab), your **completed Inclusion of Publications Statement** (from the Inclusion of Publications Statement tab) in the final version of your thesis that you submit to the Library.

Information on how to submit the final copies of your thesis to the Library is available in the completion email sent to you by the GRS.

Thesis submission for the degree of Doctor of Philosophy

Thesis Title and Abstract | **Declarations** | Inclusion of Publications Statement | Corrected Thesis and Responses

ORIGINALITY STATEMENT

☑ I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

COPYRIGHT STATEMENT

☑ I hereby grant the University of New South Wales or its agents a non-exclusive licence to archive and to make available (including to members of the public) my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known. I acknowledge that I retain all intellectual property rights which subsist in my thesis or dissertation, such as copyright and patent rights, subject to applicable law. I also retain the right to use all or part of my thesis or dissertation in future works (such as articles or books).

For any substantial portions of copyright material used in this thesis, written permission for use has been obtained, or the copyright material is removed from the final public version of the thesis.

AUTHENTICITY STATEMENT

☑ I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis.

Welcome to the Research Alumni Portal, Aaliya Sarfaraz!

You will be able to download the finalised version of all thesis submissions that were processed in GRIS here.

Please ensure to include the **completed declaration** (from the Declarations tab), your **completed Inclusion of Publications Statement** (from the Inclusion of Publications Statement tab) in the final version of your thesis that you submit to the Library.

Information on how to submit the final copies of your thesis to the Library is available in the completion email sent to you by the GRS.

Thesis submission for the degree of Doctor of Philosophy

Thesis Title and Abstract | Declarations | **Inclusion of Publications Statement** | Corrected Thesis and Responses

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

Publications can be used in the candidate's thesis in lieu of a Chapter provided:

- The candidate contributed **greater than 50%** of the content in the publication and are the "primary author", i.e. they were responsible primarily for the planning, execution and preparation of the work for publication.
- The candidate has obtained approval to include the publication in their thesis in lieu of a Chapter from their Supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis.

☑ The candidate has declared that **some of the work described in their thesis has been published and has been documented in the relevant Chapters with acknowledgement**.

A short statement on where this work appears in the thesis and how this work is acknowledged within chapter/s:

> Chapter 3 is compromised of the journal paper: AccessChain: An Access Control Framework to Protect Data Access in Blockchain-Enabled Supply Chain which is under the second review in the Journal of Future Generation Computer Systems.
>
> Chapter 4 is compromised of the work: The implications of blockchain-coordinated information sharing within a supply chain: A simulation study which is published in the Research and Applications journal.
>
> Chapter 5 is compromised of the journal paper: RPoC: An efficient and scalable consensus algorithm for SCM applications which is under the second review in the Journal of Ambient Intelligence and Humanized Computing.
>
> Chapter 6 is compromised of the journal paper: "A tree structure-based improved blockchain framework for a secure online bidding system" which is published in Computers & Security journal.
>
> All the chapters include acknowledgments.

Candidate's Declaration

I declare that I have complied with the Thesis Examination Procedure.

Abstract

Supply chains have progressed through time from being limited to a few regional traders to becoming complicated business networks . As a result, supply chain management systems now rely significantly on the digital revolution for the privacy and security of data. Due to key qualities of blockchain, such as transparency, immutability and decentralization, it has recently gained a lot of interest as a way to solve security, privacy and scalability problems in supply chains. However conventional blockchains are not appropriate for supply chain ecosystems because they are computationally costly, have a limited potential to scale and fail to provide trust. Consequently, due to limitations with a lack of trust and coordination, supply chains tend to fail to foster trust among the network's participants. Assuring data privacy in a supply chain ecosystem is another challenge. If information is being shared with a large number of participants without establishing data privacy, access control risks arise in the network. Protecting data privacy is a concern when sending corporate data, including locations, manufacturing supplies and demand information.

The third challenge in supply chain management is scalability, which continues to be a significant barrier to adoption. As the amount of transactions in a supply chain tends to increase along with the number of nodes in a network. So scalability is essential for blockchain adoption in supply chain networks. This thesis seeks to address the challenges of privacy, scalability and trust by providing frameworks for how to effectively combine blockchains with supply chains.

This thesis makes four novel contributions. It first develops a blockchain-based framework with **A**ttribute-**B**ased **A**ccess **C**ontrol (ABAC) model to assure data privacy by adopting a distributed framework to enable fine-grained, dynamic access control management for supply chain management. To solve the data privacy challenge, *AccessChain* is developed. This proposed AccessChain model has two types of ledgers in the system: local and global. Local ledgers are used to store business contracts between stakeholders and the ABAC model management, whereas the global ledger is used to record transaction data. AccessChain can enable decentralized, fine-grained and dynamic access control management in SCM when combined with the ABAC model and blockchain technology (BCT). The framework enables a systematic approach that advantages the supply chain, and the experiments yield convincing results. Furthermore, the results of performance monitoring shows that AccessChain's response time with four local ledgers is acceptable, and therefore it provides significantly greater scalability.

Next, a framework for reducing the bullwhip effect (BWE) in SCM is proposed. The framework also focuses on combining data visibility with trust. BWE is first observed in SC and then a blockchain architecture design is used to minimize it. Full sharing of demand data has been shown to help improve the robustness of overall performance in a multi-echelon SC environment, especially for BWE mitigation and cumulative cost reduction. It is observed that when it comes to providing access to data, information sharing using a blockchain has some obvious benefits in a supply chain. Furthermore, when data sharing is distributed, parties in the supply chain will have fair access to other parties' data, even though they are farther downstream. Sharing customer demand is important in a supply chain to enhance decision-making, reduce costs and promote the final end product. This work also explores the ability of BCT as a solution in a distributed ledger approach to create a trust-enhanced environment where trust is established so that stakeholders can share their information effectively.

To provide visibility and coordination along with a blockchain consensus process, a new consensus algorithm, namely Reputation-based proof-of cooperation (RPoC), is proposed for blockchain-based SCM, which does not involve validators to solve any mathematical puzzle before storing a new block. The RPoC algorithm is an efficient and scalable consensus algorithm that selects the consensus node dynamically and permits a large number of nodes to participate in the consensus process. The algorithm decreases the workload on individual nodes while increasing consensus performance by allocating the transaction verification process to specific nodes. Through extensive theoretical analyses and experimentation, the suitability of the proposed algorithm is well grounded in terms of scalability and efficiency.

The thesis concludes with a blockchain-enabled framework that addresses the issue of preserving privacy and security for an open-bid auction system. This work implements a bid management system in a private BC environment to provide a secure bidding scheme. The novelty of this framework derives from an enhanced approach for integrating BC structures by replacing the original chain structure with a tree structure. Throughout the online world, user privacy is a primary concern, because the electronic environment enables the collection of personal data. Hence a suitable cryptographic protocol for an open-bid auction atop BC is proposed. Here the primary aim is to achieve security and privacy with greater efficiency, which largely depends on the effectiveness of the encryption algorithms used by BC. Essentially this work considers Elliptic Curve Cryptography (ECC) and a dynamic cryptographic accumulator encryption algorithm to enhance security between auctioneer and bidder. The proposed e-bidding scheme and the findings from this study should foster the further growth of BC strategies.

Keywords

Blockchain

Supply chain

Access control

ABAC

Authorization

Data security and privacy

Supply chain collaboration

Bullwhip effect

Lead time

Trust

Consensus algorithm

Efficiency

Decentralization

Tree data structure

Scalability

Encryption algorithms

Dynamic accumulator

Open-bid auction

Acknowledgment

First and foremost, my sincere gratitude goes to ALLAH the Almighty, the most gracious and benevolent. All praises are due to Him for showering His blessings throughout this PhD journey.

I wish to express my utmost gratitude and deepest appreciation to my supervisor, Dr. Daryl Essam and Dr. Ripon Chakrabortty for all their excellent advice, continuous support, endless motivation, patience, encouragement and immense knowledge that was given throughout this work. Having the opportunity to work with these two prominent people has been a wonderful lifetime experience, which I am highly fortunate and grateful for.

I would like to thank the University of New South Wales, Canberra at the Australian Defense Force Academy, to provide me with a scholarship to study at that institution. I would also like to thank my brother at UNSW, Dr. Adnan Farooq Awan for his immense support.

Special thanks to my parents, husband and siblings for their understanding, patience, encouragement and numerous sacrifices. I wish to thank my friend, Tracey Radbourne for the warm friendship and support during the course of my study and for being the coolest grandma to my daughter.

Above all, I owe my love to my beloved daughter Anaya Ahmed, thanks for being the source of my strength and for making my life more meaningful.

Aaliya Sarfaraz

Dedication

To my beloved
Anaya Ahmed Khan

Publications

Journals

The work conducted under this thesis has led to the following publications (listed in chronological order).

- Sarfaraz, A., Chakrabortty, R. K., & Essam, D. L. (2021). A tree structure-based improved blockchain framework for a secure online bidding system. Computers & Security, 102, 102147.
- Sarfaraz, A., Chakrabortty, R. K., & Essam, D. L. (2022). The implications of blockchain-coordinated information sharing within a supply chain: A simulation study. Blockchain: Research and Applications, 100110.
- Sarfaraz, A., Chakrabortty, R. K., & Essam, D. L. (2023). Reputation based proof of cooperation: an efficient and scalable consensus algorithm for supply chain applications. Journal of Ambient Intelligence and Humanized Computing, 1-17.
- Sarfaraz, A., Chakrabortty, R. K., & Essam, D. L. (2022). AccessChain: An Access Control Framework to Protect Data Access in Blockchain Enabled Supply Chain. Available Online: `https://www.preprints.org/manuscript/202202.0106/v1` -*under second review in Future Generation Computer Systems Journal.*

Conference

- Sarfaraz, A., Chakrabortty, R. K., & Essam, D. L. (2022, December). Towards a Scalable Permissioned Blockchain Framework for Supply Chain Management. In 2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (pp. 960-964). IEEE.

Contents

List of Figures

List of Tables

Abbreviations

ABAC Attribute Based Access Control

ACL Access Control List

AES Advanced Encryption Standard

BCT Blockchain Technology

BFT Byzantine Fault Tolerance

BWE Bullwhip Effect

CA Certification Authority

CCAAC Capability-based Context-Aware Access Control

DAC Discretionary Access Control

DCapAC Distributed Capability-based Access Control

DES Data Encryption Standard

DPoS Delegated Proof of Stake

ECC Elliptic Curve Cryptography

ECDSA Elliptic Curve Digital Signature Algorithm

IS Information Sharing

PBFT Practical Byzantine Fault Tolerance

PoA Proof of Authority

PoC Proof of Capacity

PoC Proof of Concept

PoI Proof of Importance

PoS Proof of Stake

PoT Proof of Trust

PoW Proof of Work

RBAC Role Based Access Control

ROP Reorder point

RPoC Reputation-based proof-of-cooperation

RSA Rivest Shamir Adleman

SC Supply Chain

SCM Supply Chain Management

SICAP Secure Identity-Based Capability

TPS Transactions Per Second

Chapter 1

Introduction

In this chapter, a brief summary of the research that was done for this thesis is provided. Additionally, it explains the significance of blockchain technology, the challenges in implementing it in supply chain management, the goals of this thesis and its contributions to academia. It also provides the thesis' organizational structure.

1.1 Supply Chain Management

Supply chain management (SCM), which spans various operations that convert raw materials into finished items, is the handling of the flow of goods and services. It entails actively optimizing an industry's procurement operations in order to maximize customer value and achieve an edge over its competitors. SCM is grounded in the notion that every product that reaches a customer is a result of the work of several businesses that make up a supply chain [1]. A supply chain is the network of people, businesses, assets, processes and technology utilized in the production and distribution of a good or service [2]. The businesses that make up the supply chain are connected together by means of both physical and informational flows. Physical flows include the processing, transportation and management of goods and resources, while information flow aids in the coordination and

control of the regular flow of resources and commodities across the supply chain. Most businesses recognize the value that supply chains can contribute to their organizations. Modern supply chains, in contrast to earlier ones, are now more focused on managing data, services and goods than they are on the accessibility, transportation and cost of physical assets [3]. There is much more that modern supply chain management systems demand than just the where and when [4].

Supply Chain vs Supply Chain Network

The establishment of the supply network (SN) is a new development in the field of SC research. A SC is a collection of primarily coordinated operations and connections that connect businesses in the value-creation process, in order to provide the final customer with an appropriate worth composition of products and/or services. While SN is described as a set of active members within an organization's SCs, as well as inactive participants to which an organization is related, that can be called upon to contribute to an SC if a need arises [5]. The SC idea entails cooperating businesses providing input on a product's development, design, delivery and commercialization. It stands for a narrowly concentrated analytical and business practice unit. Some of its success factors are offering a competitive product/service mix and having operational and market orientations [6]. The ontological perspective is expanded inside the SN paradigm to take into account, analyze and prepare for more complicated, dynamic and interconnected phenomena essential to the business world and managing relationships between trading parties.

Importance of Data Sharing in Supply Chains

SCM has progressed rapidly as a result of advancements in data accessibility, automation and digital technologies. Supply chains are fundamentally dynamic since they consist of several tiers of organizations connected globally. SCM can be thought of as the coordination of distributed decision-making among individuals or organizations regarding the flow of information, finances, and products throughout the supply chain [7]. Coordination may result in long-term benefits for everyone involved in SC. For instance, a manufacturer might eventually cut costs by re-configuring processes using better information and

retaining consumers by fostering customer loyalty through coordination. A retailer could minimize inventory and/or increase inventory turns, decrease labor costs, and maintain operations with an uninterrupted supply of goods [8]. The coordination of supply chains strengthens when every tier of the chain performs actions that are aligned. Supply chain coordination necessitates that each level of the supply chain communicates information and consider the influence of its actions on subsequent stages. The supply chain's many stages may have competing aims, or information may move between them slowly and inaccurately, leading to a lack of coordination. If each stage of a supply chain is owned by a different party, the goals of the various phases may conflict. Each level strives to maximize its own earnings as a result, which frequently leads to decisions that lower overall profit margins in the supply chain [9].

SCM has grown significantly reliant on digitization since the introduction of different technologies, such as big data, Internet of things (IoT) and blockchain technology [10]. Blockchain technology can have a substantial beneficial influence on a business [11]. In particular, it could lower business costs while increasing their overall efficiency. The tremendous growth of supply networks and digitization has simplified overall data collaboration processes. Which in turn, can improve the performance of the entire supply chain by lowering inventories and regulating operations [12]. Supply chain performance is the key since modern competition is not really among businesses, but rather across supply networks. Data sharing brings many advantages for a supply chain's partners:

1. Collaboration: Coordination between organizations is facilitated through data exchange. One of the key obstacles of SCM is remoteness, whether geographical, temporal or informational [13]. The development of data-sharing tools has now made it feasible to address this issue and foster trust among SC participants.

2. Risk management: Data sharing allows supply chain participants to evaluate processes and regulations to determine if they are efficient and compliant. This reduces the possibility of fraudulent activity, while also streamlining processes and preventing performance bottlenecks. Data sharing can assist to decrease delays because firms will be able to identify any issues early and take rapid action to address them,

as it provides insights on cutting down on lead time variations and shipment delays [14].

3. Cost reduction: Information sharing among supply chain participants improves the efficiency of the management of operations and any resources (financial or material) [15]. Manufacturers can reduce their inventory and operating costs by having an accurate knowledge of demand data, production and sales patterns.

4. Quality & production efficiency: If data from each phase of a project's life cycle is shared with each supply chain participant, it can contribute to data authentication. Collaboration in strategic sourcing paves the way for more efficient operations, lowers costs and quicker response times for enterprises. It also gives participants flexibility in determining collectively if a disruptive situation arises or to optimize operations [16]. Furthermore, businesses can assure that the final product fulfills customer criteria for reliability and efficiency.

Blockchain for Supply Chains

Blockchain technology has the ability to transform supply chains by establishing a more efficient and transparent approach to monitoring the movement of products.

In a conventional supply chain, several stakeholders typically have siloed and fragmented information regarding the movement of products. This may result in forecasting errors, delays and disputes, as well as a lack of accountability and transparency. Blockchain technology has the potential to solve these issues by creating a decentralized, transparent ledger of all supply chain transactions. This ledger is accessible by all supply chain participants and serves as a single source of information for all transactions [17].

Every time a product changes hands in a supply chain powered by blockchain, a new block is added to the network. Each block holds a digital record of the transaction, including details on the product, the time and date of the transaction, and the names of the parties involved. These building elements are connected in a chain to produce an immutable record of the whole supply chain. This transparency and immutability can

help to minimize errors, delays, and fraud in the supply chain [18]. It can also enhance visibility into the movement of goods, allowing for more efficient inventory management and faster response times to disruptions in the supply chain. Section 2.2.1 contains a more thorough discussion on this subject matter.

Blockchain vs Distributed Ledger Technology

Blockchain and distributed ledger technology (DLT) are two related concepts, but they have some key differences. Blockchain is a specific type of DLT that was originally developed for use in the cryptocurrency Bitcoin [19]. Distributed ledger technology (DLT) is a broader term that encompasses all types of decentralized databases that are distributed across a network of nodes. DLT can include technologies such as blockchain, but it also includes other types of decentralized databases that do not use a blockchain structure. One of the key differences between blockchain and DLT is that blockchain is a specific type of DLT that uses a particular structure to record and validate transactions. In contrast, DLT is a more general term that refers to any type of decentralized database that is distributed across a network of nodes. Additionally, blockchain is typically more secure than other types of DLT because of its use of cryptographic hashing and consensus algorithm. Among DLT, blockchain is one particular kind that is intended to keep track of transactions or any digital communication and give businesses the transparency, efficiency, and security they were looking for [20].

1.2 Motivation

Blockchain is the foundation of cryptocurrencies like Bitcoin [19] or Ethereum [21], however, they can be utilized for a variety of services, including administrating supply chain management and delivering financial services. Therefore, the ownership of physical assets can be handled via blockchain technology. Large volumes of data are involved in supply chains, particularly when items are transported across international borders. It might be challenging to identify the root cause of issues while using conventional data-storing

techniques. As a result, the privacy of the digital data of physical assets and events in blockchains is becoming more and more difficult. This section highlights the need for additional research on the subject while identifying the fundamental problems with the state-of-the-art.

1.2.1 Challenges in Adopting Blockchain in SCM

Blockchain technology is an appealing proposition for SCM security and scalability because of the many advantages it offers. However, the following challenges must be overcome in order to successfully adapt the current blockchain design to the SCM environment.

1. Latency

   Latency sometimes referred to as "Block time", is the amount of time needed to create the next block of transactions on the blockchain. In comparison to current Web2 standards, most existing blockchain networks are extremely slow [22]For instance, one transaction on the Bitcoin network takes over ten minutes, whereas confirmation on the Ethereum network might take up to 30 seconds or more depending on the network load. The latency issue with blockchain is one of the most significant roadblocks to wider adoption.

2. Throughput

   In the blockchain, the amount of transactions that can be stored per second is referred to as throughput. The transaction throughput of a decentralized network is determined by the consensus algorithm used by a blockchain network. For instance, a proof-of-stake (PoS) [23] network like Cardano has a higher throughput than a proof-of-work (PoW) [19] blockchain like Bitcoin. For instance, with a typical blockchain implementation, confirming a transaction might take up to 30 minutes for Bitcoin and 5 minutes for Ethereum. Likewise, Bitcoin has a throughput of 7 transactions per second (TPS) while Ethereum has a throughput of 30 TPS. The quantity of supply chain events fluctuates along with the frequency of trade in a supply chain. With

a higher trade frequency, the increasing transaction load may consequently result in increased network latency and decreased throughput. The block size, traffic, storage, number of nodes and complexity of transactions on a blockchain, are other elements that influence throughput.

3. Privacy

   A blockchain-based transaction is immutable and tamper-proof and transparent to the whole network. However keeping track of every user transaction on a blockchain would threaten the security of user privacy [24]. If a trade secret is revealed in the supply chain, it could give a business an edge over its competitors and the privacy of stakeholders could be compromised.

4. Scalability

   Scalability is the ability of a system to manage resources and handle increasing workloads. In a traditional blockchain network, every node receives and verifies every block. This is a significant challenge since network congestion and computing overheads grow with network node size. Hundreds of stakeholders may be involved in supply chain operations on a regular basis. There's a potential that a single blockchain system would not be able to handle the growing number of transactions. A scalable blockchain network should be able to accommodate an exponentially growing number of participants, without compromising system performance.

5. Resource intensive consensus algorithms

   It is ultimately the consensus mechanism that defines a network's security level, throughput and scalability. Compared to a traditional database, an energy-intensive consensus algorithm like PoW needs a huge amount of power. All miners compete to be the first to solve a mathematical puzzle in order to validate a transaction. This raises the price of blockchain-based transactions and significantly burdens the ecosystem with carbon emissions. On the other hand, alternative blockchain consensus approaches, such as Proof-of-Stake (PoS), require significant resources from the network participants, which is considerably more than what the majority of SCM stakeholders can provide.

6. Access Control

   In computer security, access control systems are used to control who has access to resources. Access control policies are often used to indicate a subject's entitlement to access such resources. Typically, a permissionless blockchain allows anybody to join and allows users to log transactions anonymously. Many characteristics of permissionless blockchain systems are not appropriate in the case of supply chains as supply chain stakeholders are trusted business partners. In contrast, with a permissioned blockchain, everyone can see trade flows and so it would breach data privacy. Therefore, some access control is required in order to establish rules defining who is permitted to read and write data on the blockchain.

7. Trust

   Blockchain technology is sometimes referred to as a "trustless" technology since it substitutes the requirement for a central authority with a system of publicly verifiable proofs [25]. Lack of trust among partners is a significant barrier to blockchain adoption. Since supply chain organizations may not trust other participants in a public blockchain network, whereas, in a private blockchain network with no anonymous users, businesses can experience better levels of trust. Platforms like *TradeLens* [26], a global logistics permissioned network, shows what may happen when businesses work together to find solutions to common problems in order to boost consumer trust. TradeLens is a blockchain-based platform for global trade that was developed by IBM and Maersk, a leading global shipping company. The platform was launched in 2018 with the goal of providing more efficient and transparent supply chain management for the global trade industry. The TradeLens platform is designed to streamline the supply chain management process by eliminating many of the manual and paper-based processes that are currently used in the industry. By doing so, it can reduce delays, errors, and costs, while improving transparency, security, and efficiency. Since its launch, TradeLens has attracted a growing number of participants from across the global trade industry, including shipping companies, ports, customs authorities, and more. Today, it is one of the leading blockchain

platforms for global trade, and it has the potential to transform the way that goods are shipped and managed around the world.

1.3 Thesis Objective

These evident facts serve as the driving force behind this thesis's overall objective, which is to develop an efficient, scalable and secure blockchain models for supply chain management. To achieve the main objective of this study, the following sub-objectives are considered:

- Finding an SCM model that meets the requirements of the research;
- Analyzing the effect of the various features of the blockchain technology on supply chains, like the consensus algorithm and encryption/decryption algorithms;
- Developing an efficient access control model to depict viable solutions for data privacy;
- Proposing a real-time data sharing model that fosters trust among supply chain participants;
- Proposing a secure consensus algorithm capable of scaling while maintaining high accuracy;
- Developing an e-auction framework that works well for open bid auctions and offers efficiency and security;
- Testing each of the models that have been proposed:
- Evaluating how the developed frameworks perform in comparison to previously designed networks.

1.4 Contributions

This thesis makes a substantial contribution to the adoption of blockchain technology in supply chains, by addressing the issues raised earlier in Section 1.2.1. In particular by achieving scalability and privacy objectives. The subsequent sections provide an overview of the specific contributions this thesis has made.

1.4.1 AccessChain: An Access Control Framework to Protect Data Access in Blockchain-Enabled Supply Chain.

Due to the concern over compromising sensitive information, supply chain stakeholders are hesitant to share information via blockchain. *AccessChain* attempts to address the issues of access control, data privacy and scalability. In this contribution, a two-tiered framework is proposed that provides scalability for handling transaction load by integrating several regional blockchains while protecting data privacy by limiting access privileges to competitive partners. The proposed model considers the use of business contracts and an ABAC policy to restrict access to data on the blockchain. When a stakeholder attempt to access data, the design ensures that only users who are members of the same business contract can see trade flows.

A quantitative and qualitative analysis has been conducted in order to evaluate *AccessChain*. *AccessChain's* resistance to relevant threats is demonstrated through quantitative security and privacy analysis, which also suggests countermeasures. According to the findings of simulations, *AccessChain* offers high performance, better data accessibility, data privacy and scalability compared to existing blockchain frameworks.

1.4.2 Blockchain-coordination for SCM

This thesis proposes an information-sharing, permissioned framework to reduce the bullwhip effect in a complex retail supply chain. While *AccessChain* solves issues with data

privacy and scalability, it does not resolve the trust issue related to the data source. The secrecy of trade secrets cannot be guaranteed, even when permissioned blockchains restrict read-only and write access to the ledger to approved supply chain stakeholders. The contribution of this framework is twofold: (a) investigating the effects of complete information sharing on reducing bullwhip effect, inventory variance and costs; and (b) proposing a trust mechanism that takes data authenticity into account and allows stakeholders to communicate information effectively in a trust-enhanced environment. A security analysis is performed on the significant threats, and defenses are recommended. When compared to traditional blockchain configurations, the suggested framework's customized implementation reduces the bullwhip effect by 99% and the inventory cost by up to 75%.

1.4.3 An Efficient and Scalable Consensus Algorithm for SCM Applications.

A reputation-based proof-of-cooperation (RPoC) consensus mechanism is proposed in this thesis work, which randomly selects consensus nodes from a large pool of nodes based on their reputation score and willingness to stake their identity in order to organize them into clusters. This process involves breaking down the huge mining effort into manageable segments and helps RPoC increase scalability and efficiency while retaining peer trust in the system. RPoC increases the scalability of the blockchain by grouping supply chain participants into clusters and assigning sole responsibility for blockchain management to the cluster master nodes. The maintenance of blockchains entails the verification of individual transactions or blocks of transactions. A transaction is referred to as the communication primitive for transmitting business events between nodes. A block is created by combining transactions, and that block is then added to the blockchain to create the distributed ledger. The proposed consensus mechanism is more effective, decentralized and scalable since it engages all network nodes, instead of a few as mining candidates.

Several sets of experiments are conducted to evaluate RPoC. Quantitative security and privacy studies demonstrate RPoC's resistance to relevant attacks. According to simu-

lation data, RPoC performs better than existing blockchain configurations, in terms of latency, processing time and scalability.

1.4.4 A Tree Structure-based Improved Blockchain Framework for a Secure Online Bidding System.

A secure private blockchain-based open-bid platform has been proposed in this work in order to address the key problems with high performance and privacy in blockchain that are detailed in Section 1.2.1. Then, a novel protocol is proposed that ensures high efficiency in searching and processing by replacing the old linear data structure with a tree data structure. A data structure is an approach to collect, organize and manage data and allows users to access, add, update and search the data stored within it. Since blockchain's linear nature makes it more suited for sequential or single-user processes, its data structure has been modified. The goal of modifying the data structure is to develop a solution to improve searchability.

The privacy, anonymity and security of open-bid auctions are challenging. This is due to the fact that it must offer a way to conduct a secure e-auction and protect the data and anonymity of the auctioneer. Additionally, coalition, anonymity and linkability of the digital world may contribute to a lack of trust in the network. In order to improve security and privacy in open-bid auction systems, an Elliptic-curve cryptography (ECC) technique with a dynamic accumulator design is developed to address these issues. To establish strong authentication mechanisms, the proposed architecture takes advantage of dynamic accumulators. In the bidding process, the bidder creates the key pair and the key accumulator gathers the keys for authentication and verification. The key accumulator confirms the bidder's identification once the bidder sends an encrypted bid. The auctioneer scans the signature, decrypts the bid and seeks to verify the tender. At that point, the identification of the auctioneer is confirmed by the key accumulator. The bid is approved by the auctioneer and recorded on the blockchain if there is no authentication error. The implementation result demonstrates that the suggested framework improves the security,

privacy and performance of auctions. The proposed framework is proven to be resilient to a variety of threats through security analysis.

1.5 Thesis Organization

Following is an outline of the thesis:

Chapter 1: This chapter presents an introduction about this thesis which includes motivation and scope of research and contribution to scientific knowledge.

Chapter 2: This chapter includes the background and literature review of related research.

Chapter 3: This chapter introduces AccessChain, a fine-grained access control framework that protects data access and solves the scalability bottleneck.

Chapter 4: This chapter presents a blockchain-coordinated framework, a permissioned blockchain for information sharing and enabling trust between supply chain entities.

Chapter 5: This chapter introduces RPoC, a scalable, efficient and trustworthy consensus mechanism for supply chain applications.

Chapter 6: This chapter introduces an online bidding framework, a private permissioned blockchain to provide a secure bidding scheme with a new protocol by replacing its usual linear data structure with a tree data structure to improve performance.

Chapter 7: This chapter summarizes the thesis and explores potential future research directions.

Chapter 2

Background Study and Review

This chapter provides background knowledge on blockchain technology and explores related studies from the perspective of its utilization in supply chain management. We begin by outlining a comprehensive overview of blockchain technology in Section 2.1. In the following section, we emphasize the most cutting-edge blockchain-based solutions for SCM scalability, privacy and trust and each section is concluded with a summary of the reviewed approaches. A thorough examination of existing consensus algorithms in the blockchain is provided in Section 2.2.2 and section 2.2.3 evaluates the existing research on privacy preservation in blockchain-enabled frameworks. The trust-focused blockchain solutions discussed in Section 2.2.4, provide a measurable statistic of the level of trust in the data stored in blockchain-supported systems. Finally, Section 2.3 provides an in-depth analysis of the e-auction research.

2.1 Blockchain Overview

Blockchain technology has gained widespread adoption for the variety of advantages it offers. In general, blockchain technology is characterized by decentralization, accountability, immutability and security [27]. In 1991, the notion of "blockchain" was first introduced, to

prevent backdating or editing of electronic data, a group of academics set out to develop a program that timestamps data. Later, Satoshi Nakamoto refined and innovated Bitcoin [19], a blockchain-based project, that was the first cryptocurrency and peer-to-peer payment system ever established. A blockchain can be characterized as a collection of blocks that are linked together and carry particular information in a safe and authentic manner. In other words, blockchain is a collection of interconnected computers, rather than a single, centralized server, making the entire network decentralized. Through this decentralized network, data security, trust and transparency are all guaranteed. Although commonly used in the finance industry, blockchain technology is also expected to have applications in a broad range of other industries, such as supply chain, healthcare and the internet of things. Blockchain is a list of linked blocks where each block is made up of two main components: the block header and transactions, as shown in Figure 2.1. The transmission of any sort of payment or data between nodes is documented in a blockchain transaction. Depending on the kind of blockchain, a block may include different types of data. For instance, transaction information such as sender, recipient and currency, are stored on the Bitcoin blockchain. A transaction has as its unique identifier, *Transaction ID*, which is a hash of all the fields in a transaction. *prev TID* stands for the previous transaction ID and connects any prior transaction done by the same node. The initial transaction in each ledger is known as the genesis transaction. There could be dependencies between transactions, where particular fields created in one transaction's outputs, are referred to as inputs in another transaction. The *Input/Output* fields store the inputs and outputs. A *public key* is used for the identification of each transaction of a node. It is generated in order to enhance anonymity. The *H(PK)* field contains the hash of the public key. Lastly, the *Sign* field keeps the signature of the node that generated the transaction.

Apart from the transaction, the block header structure also varies depending on the blockchain type, as shown in the top portion of Figure 2.1. The block header contains an identifier called *Block ID*, which represents the hash of all the information in a block. *H(Prev block)* is the hash of the preceding block, thereby generating a chain between them and keeping the blockchain immutable. So, a block header carries the hash of the block itself and the hash of the previous block, like a fingerprint, a hash is always unique and

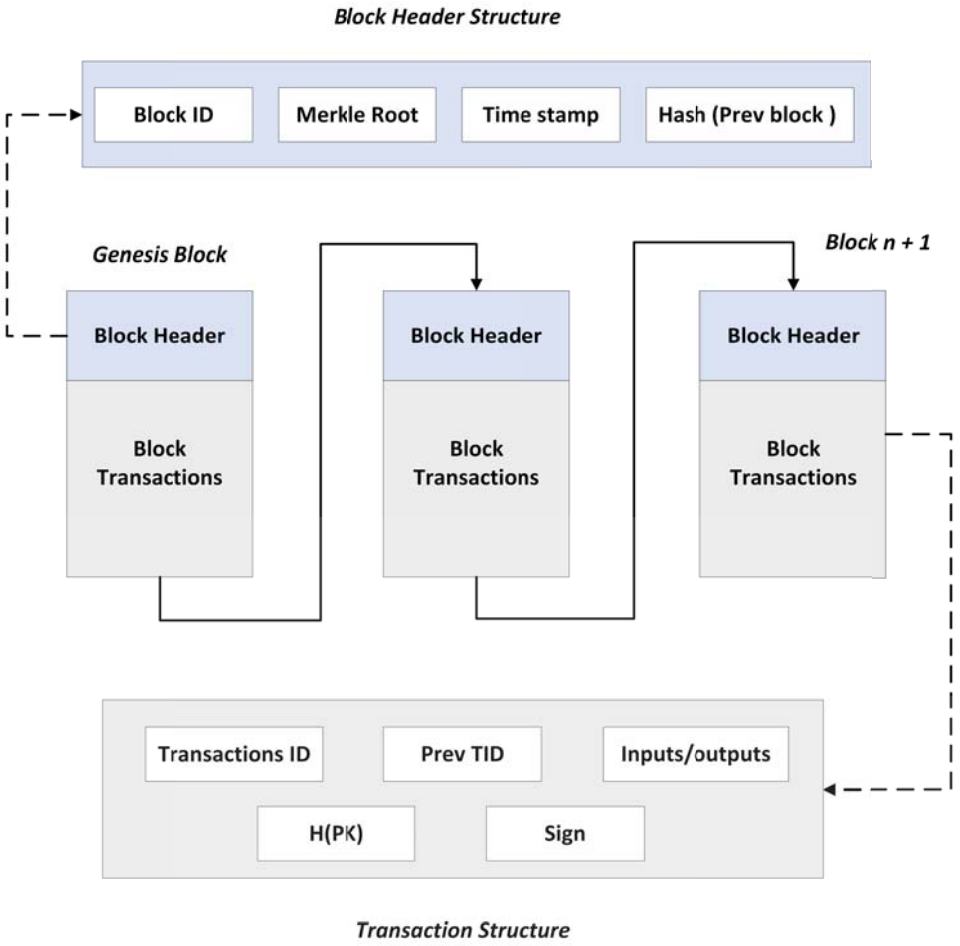


Figure 2.1: Blockchain structure, where H=hash, ID=identifier, TID = transaction ID, PK= public key, sign = signature.

identifies a block and all of its contents. A block's hash is determined after it has been produced. Any modification to the block will lead to modifications to the hash. This essentially establishes a chain of blocks and contributes to the security of a blockchain. The *Timestamp* field indicates the time the block was created. The block's Merkle root is also included in the block header. A *Merkle tree* [28], also known as a binary tree, is employed to effectively compact a massive amount of information into hash values. In a hash tree, each leaf node is labelled with the cryptographic hash of a data block and each non-leaf node is labelled with the cryptographic hash of the labels of its child nodes.

Although most hash tree designs are binary, there could be a large number of child nodes.

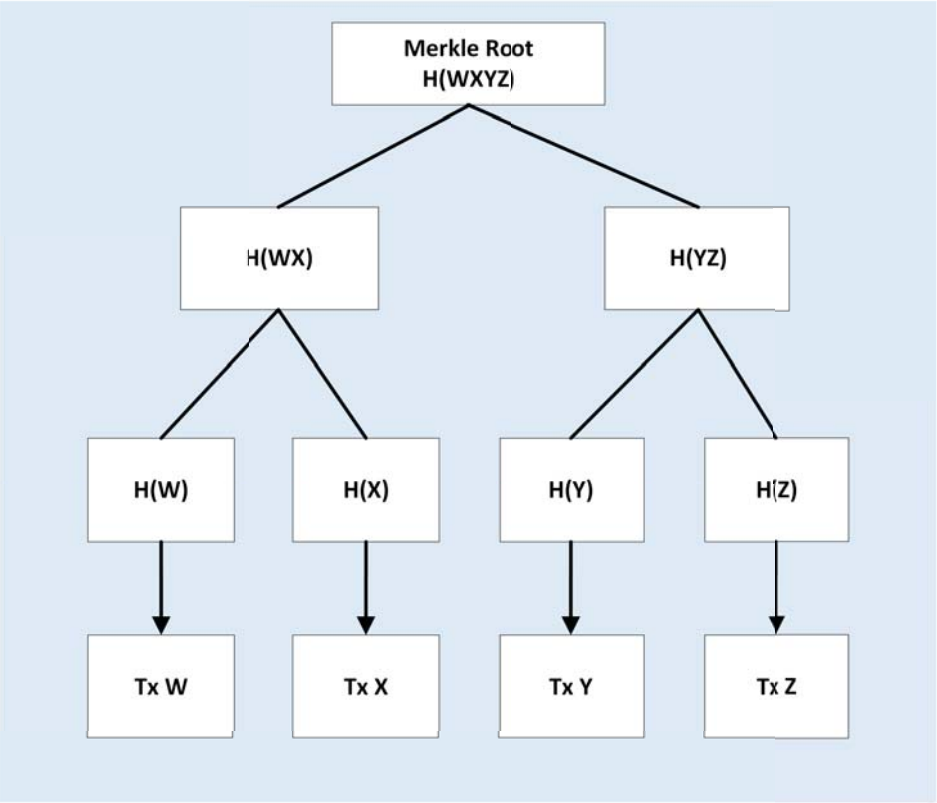


Figure 2.2: Merkle tree structure.

Figure 2.2 depicts a basic Merkle tree with four transactions, *Tx W*, *Tx X*, *Tx Y* and *Tx Z*. Afterwards, each of these transactions is subsequently hashed. The first step is to calculate the hash of each transaction, producing the values *H(W)*, *H(X)*, *H(Y)* and *H(Z)*. Then the hashes are rehashed in consecutive pairs, yielding *H(WX)* and *H(YZ)*. Lastly, *H(WX)* and *H(YZ)* are hashed once more to provide *H(WXYZ)* Merkle root. Merkle roots are fundamental to the processing needed to maintain cryptocurrencies, such as bitcoin, in operation [29].

2.1.1 Mining

The process of adding new transactions to the ledger that is shared by all blockchain nodes is known as mining. Mining, as used in reference to cryptocurrencies, denotes the process through which networks of specialized machines produce and distribute new coins, as well as authenticate new transactions [30]. The network nodes, known as *miners*, authenticate transactions after they are generated and broadcast to the network. Coins are rewarded to network miners in exchange for their services. A mining process ensures that new coins are generated on the blockchain, and keeps the miners motivated to preserve system security. All validated transactions are placed in a transaction pool, and miners are in charge of putting all of the pending transactions together into a block. Finally, this new block is appended to the blockchain by utilizing a consensus algorithm. The consensus algorithm is a technique that helps all network nodes to agree on a single ledger state. The consensus mechanism guarantees randomization among the miners and consistency of the blockchain among network participants. Randomness enhances blockchain security by preventing fraudulent miners from mining blocks. Section 2.2.2 will further explore the consensus mechanisms utilized in the blockchain.

Mining is crucial to the security of any blockchain, in addition to adding new coins to circulation. It authenticates and protects the blockchain, enabling cryptocurrencies to operate as a peer-to-peer decentralized network without the need for third-party supervision [31]. Additionally, it encourages miners to offer their computing capabilities/resources to the network.

Blockchain Forks

Forks in a blockchain network are basically partitions in the network. The blockchain may be split into two different versions if there is Unanimous consensus [32]. Forks are connected to the fact that all nodes follow the same consensus method in order to preserve the blockchain's record. Due to the blockchain protocol's backwards compatibility and the time at which a new block is created, two different sorts of forks can happen. They include the following types:

1. Soft fork: A soft fork is a fork in the blockchain that can happen when out-of-date network nodes do not comply with a protocol that is agreed to by the more recent nodes. This could lead to unnoticed out-of-sync situations or old nodes accepting data that seems wrong to newer nodes.

2. Hard fork: When there is a hard fork, a new currency is created along with the original currency, as happened with Ethereum (original: Ethereum, new: Ethereum Classic) and Bitcoin (original: Bitcoin, new: Bitcoin cash). The nodes that decide to upgrade their software are given an equivalent amount of currency, preventing any monetary losses. It is up to each full node to decide whether to join a certain chain. Newer transactions must be made legitimate in order for a node to join the new chain; otherwise, it will continue to operate normally.

2.1.2 Symmetric and Asymmetric Encryption

Blockchain technology has been created using a variety of different cryptographic principles. In the blockchain, cryptography is primarily employed to maintain data integrity and secure user privacy and transaction data. Cryptography is a collection of protocols that shields information from unauthorized individuals during a communication process [33]; symmetric and asymmetric encryption are two of the fundamental methods of cryptography. Since the time of the Roman Empire, symmetric encryption has been one of the most popular and oldest methods of encrypting confidential and sensitive data. It encrypts and decrypts data using a single, secure cryptographic key. A symmetric encryption algorithm reads the plain text and a key before starting to encrypt it. To encrypt the original sensitive information, the key collaborates with the algorithm to convert a plain text to cipher-text. This is effective for storing data that has to be decrypted at a later stage. When utilizing symmetric encryption to send and receive data, senders and receivers must both be aware of the secret key. However, the use of a single key for encryption and decryption raises concerns, because if the key became compromised, all the data it has encrypted would also be at risk. The Data Encryption Standard (DES), Advanced Encryption Standard (AES) and TLS/SSL protocols are popular instances of

symmetric encryption.

Asymmetric encryption provides a solution to the single secret key problem by utilizing a key pair for encryption. Asymmetric encryption, commonly referred to as public-key cryptography, employs mathematically linked public and private key pairs to encrypt and decrypt sensitive data. The mathematical relationship between the keys is that the private key cannot be determined from the public key, whereas the public key can be obtained from the private key. The private key is kept hidden, while the public key serves as the owner's identification, allowing them to demonstrate ownership of the private key. This is due to the fact that encryption uses the public key, whereas decryption uses the private key. The sender will receive the recipient's public key, which will be used to encrypt the data which guarantees that only the receiver can decode the data using their own private key. Similarly, a sender can encrypt data with their private key and then anyone with their public key may decrypt and read it.

Different situations necessitate the deployment of symmetric or asymmetric encryption. Symmetric encryption, which employs a single key, is best suited to data-at-rest applications. On the contrary, data transmission between nodes through the internet should be encrypted using asymmetric protocols. To further strengthen the security of the data, these encryption algorithms are combined with other techniques like digital signature, which we will discuss further in the following section. In order to ensure the integrity and security of the blockchain, cryptography is crucial since it plays a vital role in keeping the public network secured. The use of cryptography in blockchain technology has a wide range of advantages, including immutability, security, scalability, non-repudiation and prevention of hackers [34]. Blockchain employs asymmetric cryptography, which uses digital signatures for verification. Every transaction that is recorded to a block is signed by the sender using a digital signature, ensuring that the data is not compromised. The Rivest-Shamir-Adleman (RSA) [35] and Elliptic Curve Digital Signature Algorithms (ECDSA) [36] are asymmetric encryption algorithms that are frequently employed in blockchain technology. They are further explained in the section below.

2.1.2.1 ECC

In 1985, N. Koblitz and Miller [36] used an elliptical curve for implementing their elliptic curve cryptography algorithm. ECC is a sort of public key cryptography, with each user having a pair of keys, a public and a private key. ECC's mathematical operations are distributed over an elliptic curve. A public key is a point in the curve, and a random number is a private key. The public key is generated by multiplying the private key in the curve with a generator point G. G is the base point or can be called a generator point. Before starting the ECC algorithm, the two parties who wish to exchange information must first approve the use of a curve and its parameters, such as the coefficients of A and B, and a base point G to be used. The elliptic curve is defined by the constants a and b used in its defining equation. The elliptic curve equation can be written as

$$Y^2 = X^3 + AX + B \tag{2.1}$$

where $4a^3 + 27b^2 \neq 0$.

This condition means the curve on the actual axis does not have a cusp or double point. That is also called Weierstrass. The addition of two points on an ecliptic curve is defined according to a set of sample rules (e.g. point 1 plus point 2 is equal to point 3). The addition operation in an ecliptic curve is the counterpart to modular multiplication in public-key cryptocurrency and multiple additions are the counterpart to modular exponentiation [37]. Furthermore, elliptic curve cryptosystems are efficient because they use smaller key sizes, and have low computational power requirements. ECC's time complexity is$(O_{\sqrt{}}X)$, and so has a low growth rate [38]. ECC has stronger resistance to attack, lower CPU and content use [39], lower network consumption and faster encryption compared to other encryption algorithms [40]. Algorithm 1 is ECC algorithm for encryption and decryption [41].

Algorithm 1 ECC Algorithm

Global Public Elements

Step I. Eq(a, b) elliptic curve with parameters a, b, and q, where q is a prime or integer of the form 2^m.

Step II. G point on elliptic curve whose order is large value n

User Alice Key Generation

Step I. Select private key $n_A; n_A < n$

Step II. Calculate public key P_A

Step III. $P_A = n_A G$

User Bob Key Generation

Step I. Select private key $n_B; n_B < n$

Step II. Calculate public key P_B

Step III. $P_B = n_B G$

Calculation of Secret Key by User Alice

Step I. $K = n_A P_B$

Calculation of Secret Key by User Bob

Step I. $K = n_B P_A$

Encryption by Alice using Bob's Public Key

Step I. Alice chooses message P_m and a random positive integer 'k'

Step II. Ciphertext: $C_m = kG, P_m + kP_B$

Decryption by Bob using his own Private Key

Step I. Ciphertext: C_m

Step II. Plaintext: $P_m = P_m + kP_B - n_B(kG) = P_m + k(n_B G) - n_B(kG)$

2.1.2.2 RSA

RSA, proposed by Ron Rivest, Adi Shamir and Leonard Adleman [35], is one of the most influential public key encryption algorithms. This algorithm is based on the incredibly difficult decomposition of large entities and can be used for both key encryption and digital signature. One element, namely the public key, can be published as the encryption key and the combination of two large prime numbers can be set as the private key. The difficulty of getting the plain text message back from the cipher text and the public key depends on the difficulty of factoring the massive product of two prime numbers [42]. Algorithms 2 and 3 are RSA's encryption and decryption algorithms respectively [41].

Algorithm 2 RSA Encryption

1: **Input**: RSA public key (n ,e), Plain text $m \in [0, n-1]$
2: **Output**:Ciphertext c
3: Compute $c = m^e \bmod n$
4: Return c:
5: **End**

Algorithm 3 RSA Decryption

1: **Input**: Public key (n,e), Private key d, Ciphertext c
2: **Output**:Plain text m
3: Compute $c = c^d \bmod n$
4: Return m.
5: **End**

Digital Signature

To establish user identification and guarantee data security, blockchain combines public key cryptography and a digital signature technique. This solution eliminates the possibility of identity theft and fraud by deterring intruders from tampering with data. Digital signatures cryptographically connect an entity to a transaction, much like traditional/handwritten signatures do, for a certain document. Digital signatures are based on number theory

principles and are thereby almost impossible to counterfeit. A genuine digital signature guarantees the receiver that the message was created by the identified sender (authentication), that the sender cannot repudiate having sent the message (non-repudiation) and that the data was not manipulated in transmission (integrity) [43].

Digital signatures employ mathematical correlations to link two different keys, a private and a public key. The key pairs are used in cryptography to encrypt, decrypt, sign and validate transactions. The steps are as follows:

1. Private key for signing the message: Signing a transaction generates a one-way hash of the electronic data to be signed. The hash is then encrypted using the private key. The digital signature consists of the encrypted hash as well as the hashing algorithm. It is recommended to encrypt the hash rather than the full message, Since a hash function may transform any input into a set length value with a predetermined length that is typically much shorter. This saves time since hashing takes far less time than signing.

2. Public key for verifying the message: There could be two parts in this process: creating the message's hash and decrypting the signature. The hash could be decrypted by utilizing the signer's public key. If this decrypted hash matches the second computed hash of the same data, it is evidence that the data has not changed since being signed. If the two hashes do not match, either the data has been manipulated (integrity) or the signature was made using a private key that does not match the public key that the signer has provided (authentication).

Non-repudiation, authentication and data integrity are crucial goals that can be accomplished with the use of digital signature applications in blockchain [44]. Hashing and digital signatures have therefore made significant contributions to raising the security level of blockchain applications.

Hashing Functions

The distributed ledger features of blockchain technology make it impossible for unauthorized users to access the information transferred during a given transaction. It draws emphasis on a blockchain's hashing and digital signature while using cryptography to achieve the appropriate level of security. In the context of blockchain technology, digital signatures and hashing both plays significant roles.

By obscuring and encoding the original message to a distinctive string, hashes protect data integrity and essentially function as pseudonyms on the blockchain, while maintaining complete transparency. A hash algorithm is a mathematical operation that converts an input string (numbers, letters, or media files) of arbitrary size into a defined length [45]. The result of a hash is always a discrete integer that is typically 32-, 64-, 128- or 256-bits long, regardless of the length of the input data. The output with a fixed length is known as a hash. The hash function should be collision-resistant in order to be cryptographically strong, which implies that finding two inputs that yield identical result is almost unattainable. Despite being a cryptographic operation, hashing is not encryption. Contrary to encryption, a hashing algorithm acts as a one-way process; the original data cannot be deciphered using only the hash. The identical hash will be generated by anyone with access to the original message and hashing algorithm. Here, a compression function (a component of a hashing algorithm) offers the characteristic of maintaining a constant length of hashing function output. A hashing function is characterized by the determinism principle, which states that the identical message hashed with the same algorithm will always yield the same hashing result. This is analogous to the blockchain architecture, where new blockchain nodes should first synchronize themselves with the rest of the operating full nodes.

Hash functions fall into a number of distinct types. Some of the popular classes are RACE Integrity Primitives Evaluation Message Digest (RIPEMD) [46], BLAKE2 [47] and Secure Hashing Algorithm (SHA) [48]. The Bitcoin blockchain employs the SHA-256 [49] hashing algorithm for a variety of tasks i.e. Markle tree makes use of hashing algorithm to ensure that it is impossible to locate two Markle trees that have the same root hash. The root hash is kept inside the block header, which contributes to protecting the block header integrity.

Furthermore, hash functions are essential components of digital signatures that protect data integrity and are utilized for blockchain transaction authentication. Moreover, every block in the blockchain carries the hash of the preceding block. This guarantees that it is not possible to change any blockchain block without being detected. In this context, it can be stated that hash functions are an essential component of blockchain technology, that is employed to guarantee the accuracy and immutability of data stored on the network.

How does the SHA algorithm work?

The SHA family consists of the SHA-0, SHA-1, SHA-2 and SHA-3 algorithms. Presently, SHA-2 is among the most popular algorithms in the field of cryptography. Four versions of SHA-2 have been released since it was first released in 2001, and these versions have undergone significant improvements over time. There are four main versions, with SHA-256 being the most commonly used, along with SHA-224, SHA-512 and SHA-384. The most recent developments in secure hashing algorithms, such as SHA-3 [50], demonstrate the significance of hashing in the blockchain. In 2015, SHA-3 was developed and is capable of substituting for SHA-2 and provides equivalent variations and hash lengths.

The input data is sorted via a loop into 256 or 512-bit (depending upon the SHA-2 variant being used) big blocks of data, one at a time until the file is expanded. A message will only be hashed once if it is precisely one block in length. This implies that the hashing function's output will only be updated once. More loops are needed if the message is larger since each loop introduces a new block of data to the hashing operation. Then, looping data is compressed using a compression function. This data, together with a portion of the message, is used by the compression function to create another set of "n" values that are repeated throughout the whole message. A Merkel down guard structure updates the internal state using a compression mechanism. Padding must be utilized when a message is too short to fit exactly one 512-bit block. When a block is padded, the remaining space is filled with a binary notation that indicates the block's message length. The padding scheme makes sure that messages with the same length and messages with identical or very similar endings, don't share the same padding and consequently, the final hash.

2.1.3 Permissioned vs Permissionless Blockchain

Blockchain technology falls into three categories: public, private and consortium blockchain. The architecture of blockchain is established based on the characteristics of its participants and how rights are assigned to them. Participants' privileges can be divided into three categories of reading, writing and validation. Users who have the right to read can look at transaction history, while those who have the right to write can write transactions to the ledger and the right of validation allows users to verify transactions [51]. Public (permissionless) blockchains are accessible to the general public, which implies that everyone has the ability to read, write and validate in public settings [52]. The creation of a digital currency is the principal use case for public blockchains. It is worth mentioning that in a public blockchain, a participant would not have any motivation to mine in the absence of a valuable underlying cryptocurrency. Take Bitcoin as an illustration. When the Bitcoin value increases, the motivation for each blockchain user to mine honestly increases. Which in turn, increases the security and difficulty of manipulation of the bitcoin blockchain. Although permissionless networks have a number of drawbacks. While networks enable any participant to join a network, the employed consensus algorithms (PoW, PoS) are inefficient in terms of both energy and time. One of the major problems with public networks is their privacy. These networks ensure pseudo-anonymity by assigning a key pair to each user. Recognizing an entity's public key entails knowing all of the transactions that the user has conducted since joining the network.

Unlike public blockchains, private (permissioned) networks are solely managed by authorized users who have received an invitation to participate and only they have the authority to read transactions [53]. Depending on the dynamics of an organization, only a subset of participants can have the right to write and validate transactions. The private architecture can be referred to as a selective network with a hierarchical permission structure, whereas a public network can be described as a fair network without a centralized power but instead employing a consensus process. Participation and operations are restricted in permissioned networks. As a result, there are fewer privacy risks and the network can handle more transactions. It is possible to use more efficient algorithms (PoC, BPFT) for

transaction validations since the basic trust in these networks is larger than in a permissionless network. Each user has an assigned role with corresponding permissions, however, this opens the door of compromising the blockchain's legitimacy if the network is incorrectly configured, i.e. by granting an entity override rights.

A consortium blockchain structure is made up of several organizations. Operations are set up and managed by the preliminary designated users in a consortium. Only individuals who have been chosen in advance are accepted, unlike a public blockchain. Therefore, the nature of blockchain is semi-private, rather than public. One distinguishing feature is that authority is dispersed evenly among all participants. Although a consortium blockchain often outperforms a public blockchain, it is less decentralized, which gives participants more control. Hyperledger [54] is an example of a consortium blockchain. Table 2.1 provided a thorough comparison of these three blockchain architectures:

Table 2.1: Comparison among blockchain systems.

Property	Public blockchain	Consortium blockchain	Private blockchain
Permission	Public	Public or restricted	Public or restricted
Consensus decision	All miners	Selected set of nodes	Within one organization
Consensus process	Permissionless	Needs permission	Needs permission
Efficiency	Low	High	High
Centralization	No	Partial	Yes
Immutability level	High	Low	Low

2.2 Blockchain for Supply Chain, Access Control, Data Sharing and Scalability

The following sections examine how blockchain technology can be used to provide scalability, access control and data sharing in Supply Chain Management (SCM). Section

2.2.1 begins by outlining the blockchain system for supply chains. Section 2.2.2 discusses the existing access control system based on blockchain. Next, Section 2.2.3 discusses blockchain-based data-sharing trust models. The discussion of the current consensus algorithms and blockchain-based solutions for scalability then follows in Section 2.2.4.

2.2.1 Blockchain for Supply Chains

Blockchain has received a lot of attention for providing security, scalability, resilience and anonymity to supply chains [55]. Blockchain is a decentralized ledger, that stores and verifies shared transaction data across numerous participants in the network. It could be classified as an immutable, auditable ledger of time-stamped blocks that are used for distributed data storage and sharing and that cannot be manipulated. The recordings of information on a blockchain are referred to as transactions, and collections of transactions are packed into blocks. The cryptographic hash value of each block is used to link the blocks together. Each block has a unique hash as well as the hash of the block before it. Therefore, the hash is a security measure that demonstrates that the block's content has not been altered.

The network nodes known as miners add newly created transactions to a pool of waiting transactions [56]. Each miner organizes the gathered transactions into a block after the total number of collected transactions reaches a specified threshold, known as the block size. A new block must adhere to a consensus mechanism in order to be added to the blockchain. The consensus mechanism makes miners unpredictable and maintains blockchain security against malicious nodes. Each participant can verify the history of a transaction and identify flaws since participants have copies of the blockchain. It's not possible to alter a transaction, as doing so would require changing all previous blocks on all shared copies of the blockchain. For supply chains, the objective is to facilitate high performance while enabling organizations to operate their business operations in a secure environment. In this regard, blockchain can significantly improve supply chains by permitting efficient delivery processes, reducing disruptions, enhancing coordination and

facilitating communication among organizations [57].

Supply networks demand permissioned blockchains, instead of public blockchains with anonymous users [58]. It is important that each piece of inventory must be connected with the identification of its specific owner at every stage of the process so that participants can determine the origin and quality of their product. Therefore, only trusted entities can be permitted to take part in such a blockchain, which means that organizations need the authorization to access the network. Furthermore, authorization must be given only to certain organizations, because data privacy is at risk due to blockchain's open and decentralized nature [24]. Any participant can view the data that businesses record on a blockchain when they broadcast transactions. As data volume increases, there is a chance that it can be utilized wrongly to monitor competing firms, trade stocks, or forecast market moves. Therefore, it is essential to ascertain and authorize the blockchain participants for security purposes.

There are various obstacles to be overcome in order to create a reliable network of partners who can share data on a blockchain. One of these is the requirement for a governance framework to establish the rules of the system, necessitating an access control framework, a data security strategy and instructions on how to preserve user and data privacy, which will be further discussed [59]. Even though a blockchain is secure, it is still possible for a fake transaction, or a block, to be approved and recorded into the supply chain, either accidentally or by a malicious node [60]. Mistakes in data entry can also lead to erroneous inventory data, which is a threat. Here, a consensus method ensures that all network transactions are genuine, legitimate and approved by the majority of users (also called miners). A resource-intensive algorithm can however affect the efficiency and volume of transactions in supply chains. Therefore, supply chains can be improved in terms of coordination, better consensus algorithm, data privacy, scalability and security.

2.2.2 Blockchain and Access Control Management

Numerous research works on blockchain technology focus on providing an access control system, either in the context of a specialized area, like healthcare or Internet of things (IoT), or as a general access control system that may be used for a wide range of applications. The next sections explore access control approaches, blockchain-based data access and recent research that aims to address the issue of managing data access in distributed systems built on blockchain technology.

Access Control Models

An efficient access control system must address critical security concerns and be focused on scalability, flexibility and consistency factors. To solve data security concerns in distributed networks, numerous access control approaches with distinct objectives have been developed. Classical access control systems, such as Role Based Access Control (RBAC), Attribute Based Access Control (ABAC) and discretionary access control (DAC) have been proposed as solutions to the problem of access control in large networks. It is worth mentioning that in both the DAC and RBAC schemes, validating subjects' access permissions is often done by a centralized authority, which can lead to a single point of failure. To overcome this shortcoming, ABAC is used to limit the number of rules, an ABAC model is made up of a set of rules that define requirements for a set of properties related to the subject, object or environment. The rules are integrated and they must be satisfied in order for access permission to be given. ABAC is gaining popularity since it has the potential to combine the demonstrated benefits of DAC and RBAC, while also overcoming several of their flaws. There have been several proposals for the ABAC model, such as the Usage Control (UCON) [61] model. UCON is attribute-based, but instead of focusing on core ABAC principles, it concentrates on advanced access control capabilities, including modifiable attributes, continuous enforcement, liabilities and restrictions.

Capability-based access control (CapBAC) is a potential option for distributed networks [62]. CapBAC-based schemes assign access rights to subjects, based on the concept of

capability. An access right is a transferable token of authorization that defines a set of access permissions for every subject [62]. Access Control List (ACL) and Capability are often used in access control management [63] because it is a centralized solution to enable administrative activities with improved traceability. The link between capabilities and access control lists (ACLs) is quite symmetrical: in the capabilities model, authorities are bound to objects requesting access; in the access control list model, authorities are bound to objects being guarded. Therefore if the objects were arranged in a table with the access-seeking items across the top and the security-seeking objects down the side, the columns would stand for sets of capabilities and the rows would stand for individual access control lists. Each object in the ACL model has an access control list that saves subjects and their object access privileges. However, ACL cannot handle complexity and is prone to system failure due to its centralized management feature. Similarly, each subject in the capability model has a capability list that specifies its access privileges to all objects. Skinner et al. [64] presented a CapAC model for implementing access control policies for an IoT network. However, the CapAC approach relied on a centralized authority and failed to consider lightweight requirements for smart devices. Furthermore, numerous models were presented to address these challenges (e.g. capability propagation and revocation, such as Secure Identity-Based Capability (SICAP), Capability-based Context-Aware Access Control (CCAAC) and Distributed Capability-based Access Control (DCapAC)). Existing access control approaches have some drawbacks, since they are user-centric and ignore an organization's relationships. To address these points, the access control system should be distributed to avoid single points of failure, adaptable and scalable to handle a large number of users, dynamic, trustworthy and must be capable of protecting the privacy, integrity and anonymity of members of the network [65].

Blockchain-based Data Access

With the evolution of blockchain technology, services were created with the goal of facilitating and strengthening supply chains [66]. Blockchain is a distributed, transparent, traceable and immutable ledger in which blocks are added in chronological sequence [67]. However, due to the decentralized nature of blockchain, it is critical to ensure reliable

access control of sensitive information. Therefore, access control is a vital mechanism for ensuring data access is not manipulated or compromised. Given the security concerns surrounding access control in SCM networks, blockchain technology, which is decentralized and tamper-proof, can be utilized to effectively store access control policies. The idea of using blockchain to store access control policies has also recently attracted a lot of interest.

Maesa et al. [68] proposed a framework for distributed auditability, which prevents a third party from refusing privileges granted by an enforceable policy and employed blockchain technology to create and manage access tokens and allows distributed transfer access across network users. However, their approach continues to rely on an external centralized policy database to manage access rights based on blockchain linkages and experimental results were not presented. Moreover, they mainly offer an implementation strategy and do not implement it in a specific case.

FairAccess [69, 70] is a blockchain-based access management framework for IoT networks. All interacting entities are identified by addresses that resemble bitcoins in order to ensure pseudonymity and access control measures are specified in smart contracts before being recorded on the blockchain. In addition, endorsement tokens are a form of unique identity that are utilized by blockchain to show connection authorization for access to a certain resource. To prevent token fraud and reuse, transaction integrity checks and double spending detection mechanisms are used. The proposed scheme alleviates the difficulty of managing a tremendous amount of admission control data from restricted IoT devices. The resource owner can set access policies and create access tokens for any peer. Additionally, by attempting to transfer a token, a token owner can delegate access to a new owner. The sender incorporates access control restrictions in the transaction output's locking scripts while transmitting a token. The receiver must first unlock the locking script to verify the provenance of the token. Although using locking scripts for access control is a good option, the computational capabilities of locking scripts are restricted. Other drawbacks with FairAccess include the fact that if a token expires or is revoked, the subject must contact the owner to obtain a new token. Moreover, for this access framework, at least two blocks must be mined to the blockchain for a new token to be effective, making it

expensive and time-consuming to gain access.

Zhang et al. [71] presented an access control framework based on smart contracts to automate access control. Several Access Control Contracts (ACCs), one Judge Contract (JC) and one Register Contract (RC) are used in a smart contract-based architecture to construct distributed, dependable IoT access control systems. Each ACC provides a single access control mechanism for a subject-object pair, including both predefined access right validation and dynamic access right validation based on the subject's activity. However, the implementation specifics of access control are not covered.

Ding et al. [72] proposed an attribute-based access control system for IoT applications by leveraging elliptic curve cryptography (ECC) to generate public and private key pairs for IoT devices in accordance with their identity or abilities and then encrypting the corresponding characteristics onto the blockchain. For access policies, access is shared using a symmetric key technique and the properties provided on the chain serve to verify the access authority. But creating a one-to-one link unavoidably drives up the cost of communication.

Liu et al. [73] provided a decentralized, fine-grained and dynamic access control framework to allow effective attribute management for large-scale IoT systems enabled by blockchain. The 'Fabric-IoT' solution is built on the Hyperledger Fabric blockchain platform and makes use of an ABAC mechanism. In this paradigm, access control restrictions are managed and controlled by smart contracts, which apply to both administrators and end users. Device Contract (DC), Policy Contract (PC) and Access Contract (AC) are the three smart contacts. DC offers a mechanism to assist in storing the distinctive URLs of chosen resources produced by certain IoT devices. PC is in charge of overseeing and implementing ABAC regulations for admin users and AC includes the primary access control measures utilized by end users. The processing time was calculated by simulating simultaneous access to three smart contracts using clients with multiple threads. However, Fabric-IoT processing time is significantly impacted by network disturbances at the concurrent node's location.

Sun et al. [74] proposed an attribute management system for IoT access control that is supported by ABAC. This proposal divides the IoT system into many functional units. Then, for each of the units, a local blockchain ledger is created. Unit entities, such as characteristics and access choices, are recorded in the local blockchain ledger. This will make it possible for more IoT devices to act as blockchain nodes. Each serves cross-domain access requests made by authorized users from each IoT unit using an identity-based signature. However, modelling the access across several units causes laborious decision-making and communication. Moreover, there is an issue with unauthorized data access since the judgment mechanism for the repeated membership of the same node in a unit is not taken into account throughout the access request procedure.

Algarni et al. [75] presented a blockchain-based access control mechanism for IoT. This offers a compact and decentralized secure access control system for implementing access control privileges with smart contracts. The main goal of the given architecture is to enable trustworthy implementation and safe communication between edge IoT devices, enabled by the underlying blockchain's scalability, auditability and transparency features. To provide more fine-grained authorization enforcement at various levels of access (such as at the user level and blockchain level), a private hierarchical blockchain structure is taken into account. Cryptographic procedures are employed at the user level. Lightweight consensus algorithms are employed at the blockchain level to impose permissions based on different IoT needs for access control. However, the authors solely presented a research analysis without any implementation or evaluation to establish the solution's applicability and efficacy. Li et al. [76] combines the benefits of the ABAC and RBAC models to provide flexibility in the granular and dynamic administration of privileges and streamline permission management across the board. However, the model provided in [76] has not been tested for system scalability; as the data in the blockchain keeps expanding, the strain on the blockchain system will gradually increase.

Recently, the processing capabilities of blockchain have been used for access control in [77]. The authors proposed a framework for identity management and access control that combines some benefits of both conventional banking and blockchain technology.

The architecture creates a trustworthy personal information transaction security control platform using smart contracts and a stateless authentication mechanism. The prototype implemented the self-sovereign identification concept in the open banking domain.

Summary

The summary of the above-discussed approaches is presented in Table 2.2. Based on the literature, it can be stated that blockchain has been investigated as a back-end design for a distributed access control framework in a number of research works. However, the majority of research that combines blockchain technology and Access Control is focused on one of three fields: IoT, health care or cloud storage. The state-of-the-art retains the following research gaps: explicit access criteria for supply chain participants in terms of accessing blockchain data, and a supply chainable blockchain architecture that can support higher transaction volumes. In Chapter 3, *AccesChain* presents a solution that takes into account the aforementioned challenges and conducts in-depth evaluations of network cost, throughput, latency and scalability, in addition to security and privacy.

Table 2.2: Comparison of access control approaches in blockchain-enabled applications.

Article Identifier	Use-case	Mechanism	Platform	Access control	Scalability	Data privacy
Maesa et al. [68]	General	Script language implements ABAC policies	Bitcoin blockchain	ABAC	✗	✓
Ouaddah et al. [70]	IoT devices	Judgment mechanism for passing access token	Bitcoin blockchain	N/A	✗	✓
Zhang et al. [71]	IoT devices	Smart contract based implementation	Ethereum	N/A	✓	✗
Ding et al. [72]	IoT devices	Smart contracts obtain the on-chain attributes	Hyperledger Fabric	ABAC	✓	✗
Liu et al. [73]	IoT devices	Smart contracts implement ABAC policies	Hyperledger Fabric	ABAC	✓	✗
Sun et al. [74]	IoT devices	ABAC and an identity-based signature for a cross-domain AC	Hyperledger Fabric	ABAC	✗	✓
Algarni et al. [75]	IoT devices	Enforcing access control using smart contracts	N/A	N/A	✗	✓
Liao et al. [77]	Open banking	Digital identity integration and data sharing	Ethereum	N/A	✗	✓
Li et al. [76]	Medical equipment supply chain	Combining the RBAC and ABAC to manage AC	Hyperledger Fabric	RBAC & ABAC	✗	✓

2.2.3 Blockchain-based Data Sharing Trust Models

This section examines blockchain-based data-sharing methods, with a focus on reducing the bullwhip effect. Blockchain technology has the potential to minimize BWE and boost partner trust, in order to facilitate the sharing of data. A considerable amount of work has been undertaken to measure the effects of BWE.

Costantino et al. [78] assumed that the root cause of BWE is lack of information sharing; in addition, the conditions that generate BWE are erroneous forecasts and elevated inventory. However, as a modelling assumption, this analysis limits the allowance of negative customer demand and replenishment orders. Another major factor is lead time; it can be split into two categories: physical lead time and information lead time [79]. It is common for SCs to encounter delays in the transmission of information and resources due to order handling because when an order is made by one business entity, it enters an upstream provider after an information lead time. As a result, there is a production time involved, usually known as physical lead time, as the component is made and the order is shipped. Problems arise in the efficient management of an SC when the demand for goods shifts from the time the order is placed to the time the material is delivered. The literature indicates that another major contributor to BWE is batching [79]. Batching has been shown to have a significant impact on the performance of SC, not only amplifying the bullwhip effect but also making these systems less efficient when satisfying consumer demand.

Modern SCs are decentralized in nature, and corporations are often unwilling to share sensitive data with other partners. Therefore, partial, or only a restricted amount of information, is shared. According to the literature, partial information sharing in SCs occurs when information is disseminated asymmetrically among SC members or exclusively among some members of the SC [80]. The effect of information sharing on the performance of an SC was evaluated by Tan [81]. That study analyzed how different information-sharing strategies work within different mechanisms of an SC and demand patterns. One of the results of the study was that, under volatile demand, a hybrid information-sharing policy increases SC efficiency. A number of studies in the literature [82] suggested that partial

information sharing may perform much better than full information sharing due to the investment and technical limitations (trust) of full information sharing.

Cachon and Fisher [83] analyzed the importance of sharing information with different retailers in an SC. In this case, to better distribute inventory among retailers, a supplier may take advantage of full information sharing. They believed that it is more important to incorporate information technology to accelerate and smooth the physical movement of goods through an SC than to use information technology to expand information flow. They assume that information is always shared truthfully, although their model is reflective of several specific SCs, hence the conclusion was restricted to this context. In particular their model has predictable demand, identical retailers, one source for inventory, no capacity restrictions, no incentive conflicts and rational ordering practices by businesses.

Dominguez et al. [84] examined the influence of adopting different strategies to implement partial information sharing on SC performance. Although they have examined several information sharing (IS) situations in a four-echelon supply model based on various operational aspects in an agent-based framework, they have not yet taken into consideration agents' logical autonomy in making cooperative choices. The authors concluded that cross-retailer information exchange had a significant impact on SC performance, particularly in stores with a wide variety of products. However, the scope of the research was restricted to the deployment of information sharing at retailers.

Moghadam and Zarandi [85] focused on managing BWE in a four-tier agent-based supply system by focusing on information sharing among parties, by providing a proper context to facilitate automated negotiation; the solution to this system is given by providing a modern agent-based architecture for BWE management by conducting an automated negotiation between retailer and manufacturer agents, in order to decide on the supply system's ordering policy. However, a lack of trust might lead to less productivity in negotiations.

Jiang [86] considered the various attributes of supply chains from the perspective of supply chain vulnerability and evaluates the existing bullwhip effect problem and information

collaboration model in supply chains based on existing theory. This study combines the benefits of Big data (BD) and IoT to investigate the key factors that affect the Bullwhip Effect in SCM. Additionally, they looked at cloud computing as a potential answer to the difficulty of IoT data collection. In this study, a simulation model is created, based on the bullwhip effect's mathematical model. Although coordination has a beneficial effect on information sharing, as is necessary to minimize BWE in SC, trust is still necessary to lower the risk of opportunistic behavior and shared understanding between suppliers and consumers, which is lacking in this study. van Engelenburg et al. [15] examined the feasibility of BC technologies in terms of minimizing BWE and discussed the key requirements in establishing an architecture for information sharing. The study also raises concerns about data privacy. In this work, the authors simulated that a retailer embraces blockchain technology and adds significant demand forecasting data to a blockchain network. Because the data is encrypted by design, the provider must first decode the data before they can access it, ensuring the privacy of the retailer's data. When the retailer wishes to communicate private demand information with the supplier, they can do so by first sharing the key to the necessary data with the supplier, after which the supplier can access the data. Due to the high cost of tampering, the retailer will not arbitrarily alter the blockchain record; instead, the supplier may confirm the validity of the private demand information supplied by the retailer. However the study lacked a prototype implementation, which would have provided more information.

Ghode et al. [87] developed a shared ledger for a particular SC where all relevant information was shared and it included the following four stakeholders: manufacturer, distributor, wholesaler and retailer. Only ordered quantity data was communicated in this BT-based SC. However, the model is quite simple; it is important to take into account that there are numerous stakeholders present at each echelon in order to validate the performance of the blockchain on the BWE. The research only considers the flow of ordered data, but in a real-world setting, there are also other transactions such as transfer, produce and financial transactions.

Hrušovskỳ and Taudes [88] suggested using homomorphic encryption (HE) or secure multi-

party computing (SMC) to calculate and exchange average order/inventory levels without exposing the sensitive information of SC participants. To demonstrate that the bullwhip effect is also lessened in the context of limited information sharing, they integrate data into a beer game supply chain model. Each actor receives a demand/order signal that is enhanced with knowledge about the previous orders of SC actors. In order for each SC actor to get the right signal on the average order quantity, the approach makes the assumption that all parties are operating honestly, in accordance with the model's principles. However, the evaluation does not specify how the system provides security or privacy in the event of malicious actors.

Generally, to address SC challenges, there have been many trust and reputation models (see Section 2.2.4). Most of these models are user-driven trust models that are based on the feedback of others via user ratings. BC's underlying philosophy is the consensus algorithm and considering data storage as a network. For instance, Malik et al. [89] has developed a BC-based approach to increase trust among participants in an SC. The authors indicated that the overall trust score of an entity is calculated based on an overall reputation score and consumer feedback. However, their hypothesis of low-trust settings and the corresponding necessity for advanced privacy security are contradicted by their expectation of trustworthy participants. Similarly, [90] suggested a blockchain-based trust and reputation model for IoT. It uses a dynamic evaluation mechanism that employs two algorithms, of dynamic evaluation windows and reputation hierarchical decay, to improve a network's security. Putra et al. [91] offered a decentralized framework for blockchain-based SCM that aims to address challenges with participant trust in data and behavior. The author designed a tiered architecture, consisting of the physical, data, blockchain and application layers. In the physical layer, trust is derived through sensor observations, while in the blockchain layer, trust is derived from adherence to trading agreements. In [92], a blockchain-based decentralized and modular trust management system is proposed for assessing trust in extremely large P2P networks. To quantify and analyze the trustworthiness of peers and identify malicious peers, a multi-dimensional trust and reputation model is used to represent trust and reputation scores in a single value obtained from many parameters with suitable weightings.

Summary

Table 2.3 summarizes the above-discussed approaches with respect to data-sharing trust models. The studies presented above highlight the significance of sharing information. However, one main drawback of the preceding research is that they consider partial information exchange in linear SC scenarios. It is critical to have a framework for incorporating new technologies into the SC network so that all stakeholders effectively share demand information with upstream tiers. It is apparent that building trust in SCM is difficult since stakeholders are often rivals. The trust mechanism is not addressed in any of the discussed approaches.

Our proposed data-sharing solutions (see Chapter 4) address the above-mentioned challenges. Thus our work is distinguished from previous studies as follows: (i) For both traditional and BC-based information-sharing configurations, a four-echelon multistage SC has been studied, in which all echelons are subject to demand variation and either BC is or is not applied. (ii) Each stakeholder can share demand data with other partners, who can then use it in their inventory control policies. (iii) Furthermore, since SC operations are frequently subjected to uncertainty, stochastic demand and lead time approaches have been utilized to provide more realistic findings. (iv) Nevertheless, another significant limitation of the studies mentioned above is that they neglected to consider establishing trust relationships among stakeholders, which is a critical component of a comprehensive information-sharing framework. Therefore, the suggested trust-based consensus mechanism, by reusing a PoA algorithm, is a significant component of our framework, as it assures that participants are motivated to share sensitive information without reservations. In the proposed solution, extensive performance evaluations are carried out and a detailed security and privacy analysis is also provided.

Table 2.3: Comparison of data sharing and trust based approaches in blockchain-enabled supply chain.

Article Name	Mechanism	Data sharing	BWE	Trust	Security Analysis	Digital Technology
Dominguez et al. [84]	Information sharing practice on heterogeneous retailers	✓(partial)	✓	✗	✗	Conventional method
van Engelenburg et al. [15]	Addressing the key requirement in developing an architecture for IS	✓	✓	✗	✗	Blockchain
Jiang [86]	Combination of BD and IoT to investigate the key factors that affect the BWE	✓	✓	✗	✗	IoT and big data
Ghode et al. [87]	A shared ledger for a particular SC	✓	✓	✗	✗	Blockchain
Moghadam and Zarandi [85]	A negotiation situation is designed between manufacturer and retailer	✓	✓	✗	✗	Conventional method
Hrušovskỳ and Taudes [88]	To avoid exchanging direct data and instead determine average values for inventory/orders using SMPC/HE approaches	✓(partial)	✓	✗	✗	Blockchain

2.2.4 Blockchain for Scalability

The value proposition of blockchain technology for enhancing supply chain scalability is thoroughly examined in this section. More precisely, a number of consensus algorithms that are widely used in the existing blockchain solutions are examined, along with many scalability solutions for blockchains.

As discussed in Chapter 1, scalability is a critical requirement in SCM. Despite all of the potential applications for blockchain, there are still significant drawbacks, one of which is scalability. Scalability continues to be a significant barrier to blockchain's adoption in supply chain environments [93]. In fact, the volume of transactions in a supply chain tends to increase as a network's nodes increase, resulting in low throughput. This is due to the fact, that all nodes must process every transaction on the network. Blockchains are regarded as scalable when their communication costs per transaction are O(N), where N is the number of network nodes [94]. Consequently, their throughput can fluctuate in response to an increase in nodes or transactions.

The scalability bottleneck has been addressed in the literature by introducing different strategies, such as using the block size, handling off-chain transactions and on-chain methods (i.e. sharding) [95, 96]. Sharding is viewed as the most promising way for enhancing the scalability of blockchain networks. The main idea behind sharding is to divide the network evenly and randomly into small chunks called shards. Instead of having the entire network process the same set of transactions, each shard will only process its own transactions. This enables the network to grow along with the number of shards, resulting in improved throughput and storage efficiency.

Wan et al. [97] uses a similar hierarchical sharding technique to increase throughput performance. Using Blockchain technology and a decentralized, hierarchical identity-based signature mechanism, HIBEChain (Hierarchical Identity-based Signature Scheme) offers an alternative to current IoT identification techniques. To create a hierarchical design, the system combines a number of private blockchains; each private blockchain serves as a node in the overall tree structure and will control a specific group of devices. As a result

of decentralizing and sharing the management of device identities among various nodes, its hierarchical layered design appears to provide a serious risk for scalability concerns. However, the authors do not specify what kind of blockchain would be appropriate or what would be the validator nodes, thus the idea still remains mostly theoretical and requires careful implementation. Additionally, the environment in which the devices are expected to operate is not taken into account by the approach.

OmniLedger [98] provides statistically representative shards for permissionless transaction processing and delivers great throughput and resistance against corruption by up to 1/4 of all participants. In order to choose validators at random for every shard, it employs a verified random function and a public-randomness protocol. OmniLedger is based on two layers of epoch-based Byzantine agreement procedures, with the shard level being in charge of intra-committee consensus and the network level being in charge of epoch randomness generation. A global identity blockchain is adopted at the network level, and only the leaders of the network can expand it. A Sybil-proof identity setup process must be used by each node that wishes to join a committee in order to register to this global blockchain. All nodes with verified credentials are obligated to carry out an interactive consistency procedure at the start of each epoch by exchanging a "ticket" based on a gossip protocol. The header of the identity blockchain and the node's address is hashed to create the ticket. The network-level leader will be chosen by the node that produces the lowest-value valid ticket. The leader must execute a verifiable random function and provide a global random string with credible proof.

Another two-level BFT system, called "RapidChain", is suggested in [99] for complete sharding. In order to start the creation of shard-level committees, RapidChain needs a reference BFT committee to execute a distributed randomness generation protocol and produce a public random string. To maintain operability throughout the committee transition, RapidChain's shard-level committee reconfiguration only rearranges a portion of the committee's participants at each epoch. The established identity of a node is transferred to a random location in the range [0, 1] during the initialization phase in a network of n nodes by using a hash function. The range is then divided into n=k regions with a fixed

committee size of k, and the shard-level committees are then constructed based on this region division. RapidChain refers to the set of the first half-shard level committees with the most active members as the "active committee set" during the reconfiguration stage. New nodes are consistently and randomly assigned to the active shard-level committees by the network-level committee. After then, it randomly reassigns a certain number of members from each active committee to other committees. RapidChain mandates that each BFT committee member runs the distributed randomness generation protocol and creates a local random string at the shard level. When the usual PoW problem based on the local random string is solved, the committee members conduct a fight for the leader's election. By communicating their votes with signatures to one another, the members choose the node with the shortest PoW solution. That node will then take the lead in guiding the BFT protocol to the intra-shard consensus for transaction commitment.

Consensus algorithms are critical for improving and automating business and vendor customer logistics between various stakeholders in SCM. It is critical in accelerating the delivery of a manufactured product with greater performance while reducing costs and time. According to the literature, most consensus algorithms are created specifically for cryptocurrency [100]. However the trend is shifting and SCM is embracing blockchain for a variety of reasons, including traceability, efficiency, security and trust.

Proof of Work (PoW) [19] is the most widely used consensus algorithm in the blockchain [101]. In PoW, miners do many calculations in order to solve a mathematical puzzle, and the puzzle is solved using a Hash algorithm such as SHA-256 [102]. A typical PoW block consists of a hash of the previous and current blocks, a transaction record and a nonce. To reach consensus, miners seek hash values that are equal to or lower than a "Target hash". Whenever a miner finds a solution, it announces the block to the entire network, where all other miners check the hash value to verify it. If the block is validated, the miner gets the reward for mining and every other node in the network will add this block to their chain. Although the PoW algorithm has a high level of decentralization and security, it confronts challenges with the mining process, as well as resource and time consumption. In addition, the speed and success rate of the hash function are largely dependent on the processing

capabilities of the hardware of the miners. Because of the aforementioned limitations, it is unsuitable for big networks like SCM, which require great efficiency.

Delegated proof-of-stake (DPoS) [103] is an improved and optimized version of Proof of Stake (PoS), where nodes are allowed to choose validators to validate blocks by voting. The following is how DPoS works: Miners are referred to as delegates and selected delegates are rotated through the network from time to time, and they deliver blocks in a predetermined sequence. However, there's a chance that delegated clients will be fraudulent and there's no way to punish malicious nodes in the system. When there are fewer delegates in the network, it is easier for them to organize themselves according to the designated time slot. When delegates publish invalid transactions, the rest of the token holders vote them out and new delegates are chosen. Users may delegate their voting power to other users they trust to vote for them. Since the number of validators is small, the network can be easily organized and validators can decide when is the best time to publish blocks. However, restricting the number of validators would result in a centralized network. Despite its scalability, energy conservation and low-cost transactions, its application in SCM is limited because of its semi-centralized nature.

According to the PoS paradigm, a node's ability to mine block transactions is proportional to the number of coins they own [104]. This strategy incentives miners to save their coins rather than spend them, while making the rich richer. Another concern with this is that nodes would accumulate as many coins as possible to reap the benefits of block formation, so this behaviour concentrates capital and reduces transaction activity. Proof of Importance (PoI) [105] is an advanced consensus mechanism that eliminates the disadvantage of the wealthy being even wealthier. With PoI, each node is assigned an importance value and nodes are chosen for mining based on that value. So each node's 'importance scale' decides which nodes are qualified to add a block to its blockchain. This approach maintains blockchain's decentralization while also striking a balance between channelling funds in wallets and circulating them out. PoI enables certain nodes to mine blocks that support the chain's infrastructure, rather than focusing solely on computational and value aspects. Since no mining is needed, the PoI algorithm is fast, energy-efficient and secure.

In Proof-of-Capacity (PoC) [106], a miner's storage is prioritized over hashing power. The PoC technique allows mining devices to validate transactions using their available hard drive space, instead of by energy consummation. PoC works by saving a list of solutions on a hard drive before it starts mining. It can produce several large datasets (known as plots) on a hard disk during work. The more plots a node has, the more chances it has to match the required hash value, resulting in a higher possibility of winning the mining reward. If a hard drive solves the previous block's problem the fastest, it wins the block. PoC is scalable, efficient and cost-effective, however, with the rise of cloud providers and large corporations, the mining process is becoming increasingly centralized and monopolized [107].

Practical Byzantine Fault Tolerance (PBFT) [108] is a scalable multi-layer PBFT consensus mechanism that hierarchically arranges nodes into different levels and restricts the amount of communication within a group. However, ensuring data consistency among nodes requires a significant amount of communication resources. The PBFT algorithm is fast at processing transaction requests, but the overhead of communications limits its scalability [99]. In order to keep the system secure, PBFT requires 3f+1 nodes in its system, where f is the maximum number of faulty nodes that the system can tolerate. Therefore for the group of nodes to make any decision, approval from 2f+1 nodes is required. The whole process can be divided into three phases: pre-prepared, prepared and commit. A primary will be chosen according to certain rules in each round and in each round, a new block is decided. In each round, a node advances to the next phase if it receives votes from more than two-thirds of all nodes. As a result, when running the PBFT algorithm, the nodes in the entire network must be specified [109]. However, a constant number of nodes in an SCM application is not guaranteed. The PBFT algorithm cannot be a perfect alternative for SCM due to the unknown number of network nodes. Unlike other consensus algorithms, this approach does not require any asset to be staked and thus allows for faster and more cost-effective consensus. Its advantages include energy efficiency and high throughput, while its drawbacks include a lack of scalability and storage latency due to the network's need to wait for all nodes' votes. Following it, a great deal of work was done into improving BFT [110, 111]. Another Byzantine-based consensus mechanism is

HoneyBadgerBFT [112], the first asynchronous BFT consensus system created specifically for a blockchain. However it leads to a significant increase in communication complexity and some financial scenarios are vulnerable to latency and scalability, demanding more in-depth analysis to resolve the situation. Moreover, HoneyBadgerBFT has a larger cryptographic overhead than PBFT. Hyperledger Fabric [54] and Zilliqa [113] are two projects that presently employ PBFT. Yin et al. [114] introduced HotStuff, which uses a three-phase commit mechanism to allow the protocol to establish agreement at the speed of actual network latency. Nevertheless, such techniques are difficult to scale up and suffer from trust difficulties created by botnets [115].

Ripple [116] divides network nodes into two types: servers and clients. Servers are responsible for the consensus process and clients can only move funds. Each server contains its own list of nodes, which are called Unique Node Lists (UNLs). The importance of UNL to the server is large. When deciding whether to pack a transaction into the ledger, the server queries the UNL nodes and if the received agreements exceed 80%, the transaction is packed into the ledger. The ledger for a node will thereby remain correct as long as the number of defective nodes in a UNL is less than 20%.

Proof of Trust (PoT) [117] calculates a node's trust based only on the total number of transactions they've completed, the number of times they've participated in validation processes and the number of times they've received complaints from other nodes during those operations. Giving service coins to reward honest behavior and assigning a low trust value to dishonest activity are among its incentive and penalty mechanisms. However such rewards and punishments are often excessively biased. Additionally, the procedure of PoT consensus algorithms is similar to that of classical techniques, besides the selection of trustworthy nodes. Work has been done to improve PoT [118] by classifying nodes as accounting, validating, or propagating, based on their trust values. However because nodes are treated differently based on their trust scores, the system cannot lead to total decentralization.

Integrating a reputation system with blockchain has received a lot of attention in the last few years, and is still being investigated. The reputation mechanism is primarily

used to facilitate delegated consensus, which reduces message complexity and resource usage by reducing the number of consensus participants [119]. Gai et al. [120] presented Proof-of-Reputation, a reputation-based consensus method for permissioned blockchain that only relies on reputation incentives and relies on trustworthy registries for quick bootstrapping. During each round, the node with the best reputation is allowed to compile and publish transactions into a block. There are three key steps to this process, which are as follows: (i) Broadcasting transactions: At the completion of each transaction, the service requester generates feedback and broadcasts it. (ii) Building block: As soon as the volume of transactions hits a certain point, the nodes begin compiling a ranking list based on the ratings of each service provider. To publish a block, a node must have the greatest reputation on this list. (3) Block verification: nodes that receive a block verify the ranking list. Aside from that, if a miner's unethical behavior is identified by the network, the releases of their reputation transactions are restricted by it. Unfortunately, only reputation revocation is able to stop a miner from exploiting the blockchain network. Some nodes can also work together to perform tasks for the requestors and build up their high reputation. Since the authors make no assumptions about the conduct of the nodes, fraudulent behavior can undermine the consensus.

Yu et al. [121] presented RepuCoin, a reputation-based weighted voting consensus method that incorporates reputation into a PoW consensus process. In RepuCoin, the consensus is carried out by a group of miners with a high reputation score. Each vote cast by a member of that group carries a certain weight. The proportion of a member's reputation determines how much weight that member's vote has. The member's vote is weighted based on both qualitative (honest behavior) and quantitative (the total quantity of that minor's contributions to the blockchain) grounds. However, RepuCoin is still classified as PoW, which means it has all of the same shortcomings as PoW, such as power inefficiency, probabilistic consensus and low throughput [122].

Zhuang et al. [123] proposed a reputation-based consensus process that is both leaders- and vote-based, where the node with the highest reputation gathers transactions from transaction blocks and broadcasts them to the top 20% of nodes with the highest reputation in

each round. These nodes check the received transaction block's signature and cast votes for it. Each vote cast by an individual from that group carries a certain value and the proportion of a member's reputation contributes towards a vote's weight. Three factors—node age, social interaction and consensus participation—are used to determine each node's reputation. This approach enhances the security of the blockchain by allowing nodes to lose their reputation in the event of misconduct. Since the voting consensus of a reputed node increases the security of the protocol, the competence of the proposed system depends heavily on the leader selection. Additionally, as it is a hybrid reputation/leader-based consensus algorithm, it is prone to all of the flaws of leader-based consensus algorithms, such as access fairness and denial of service attacks.

A permissionless hybrid reputation/proof-of-reputation-X consensus mechanism was proposed by Bou Abdo et al. [122]. This technique substitutes a new registration process for the trusted identity database in proof-of-reputation-X in order to make it compatible with permissionless blockchain, while maintaining a reputation mechanism and named it PL-PoRX. New miners initially obtain a walletID created by other miners, after which they may finish the registration process. The authors showed that the proposed technique reduces the number of blocks produced by malicious miners by contrasting its performance with proof-of-reputation-X. However, the algorithm is centralized as user registration is handled by a third party.

Fortino et al. [124] presented a two-phase group creation approach to support their reputation model in an IoT setting. In this study, the concept of modelling each agent's reputation was enhanced, such that rather than using a constant value, the features of IoT devices were used to determine the number of groups and the reputation threshold for merging with a group. As a result, in the first phase, the k-means clustering algorithm selects the number of groups based on the reputation scores of the agents. Each agent switches from one group to another during the second phase, which is repeated on a regular basis.

Mohsenzadeh et al. [125] proposed a consensus model, known as the fair reputation-based consensus model (FRCM), for the equitable selection of community members' transactions

and the only factor used to choose the target trustee is the history of the trustees' behavior. Pre-consensus and consensus are the two primary stages of FRCM, which execute both of these phases for each time period. Each trustee signs the transactions they receive from the community during the pre-consensus phases and distributes them to x additional randomly chosen trustees. Then these trustees merge the transactions data they obtained with the transactions they acquired from other trustees, and transmit it to x more trustees who were chosen at random. This procedure continues until the time period ends. The transactions that each trustee may execute and record in the current block are announced during the consensus phase, when all trustees have an equal chance to do so, regardless of the transaction context. Following this, the reputation of each trustee is determined.

Summary

Table 2.4 summarizes the discussion of the state-of-the-art consensus algorithm in the blockchain. Nevertheless, existing algorithms rely on resource-based or voting-based mechanisms, which increase communication costs by requiring several interactions. Moreover, some reputation-based algorithms achieve security and scalability while reducing fault tolerance or being semi-centralized.

The proposed PRoC consensus algorithm addresses the aforementioned issues (see Chapter 5). Despite the fact that its terminology is built on a reputation mechanism, it has significant distinctions. Firstly, a consensus process that maintains peer trust is proposed, where instead of selecting a few nodes for mining, all network nodes participate and collaborate in the consensus mechanism, making it scalable, efficient and decentralized. Second, a high reputation score is not only a metric for consensus nodes; in order to be chosen for mining, they must put their identity at stake. Moreover, to minimize the communication overhead, rather than relying on a voting mechanism to verify new blocks, the proposed work relies on signature verification.

Table 2.4: Comparison of state-of-the-art consensus algorithm in blockchain.

Article Identifier	Mechanism	Applicable blockchain type	Platform	Security	Scalability	Fault tolerance
PoW [19]	Resource-based	Permissionless	Bitcoin	High	High	<=25%
DPoS [103]	Voting	Permissioned	BitShares	High	Moderate	<51%
PoI [105]	Vested coins	Permissionless	NEM	Moderate	Moderate	<25%
PoC [106]	Hard disk space	Permissionless	Burst	Low	High	<25%
PBFT [108]	Voting	Permissioned	Hypereldger Fabric	High	Low	<33.3%
Ripple [116]	Voting	Permissionless	XRP	High	Low	<20%
Bitcoin-NG [126]	Resource	Permissioned	Bitcoin core	Moderate	High	50%
PoR [120]	Reputation	Permissioned	Python	Moderate	High	Not specified
Trust [118]	Trust	Consortium Blockchain	Ganache Ethereum	Moderate	High	40%
PoR [123]	Hybrid reputation/leader	Permissionless/ permissioned	Not specified	Moderate	Low	Not specified
Repucoin [121]	Hybrid reputation/resource	Permissionless	BFT-SMaRt	High	Low	<33.3%
PL-PoRX [122]	Hybrid reputation/proof-of-reputation-X	Permissionless	Not specified	Moderate	High	Low
FRCM [125]	Reputation	Permissionless/ permissioned	Python	Low	Moderate	Not specified

2.3 Blockchain for E-auction

Several user privacy solutions have been proposed in the literature for modern online marketplaces. Most of these solutions address user privacy by hiding the identity of the bidder on the blockchain, while others focus on bid security. The identity of bidders should be secured in an English auction; hence bidder privacy is important. The subsequent subsection explore numerous approaches in the literature that use BC technology to establish a secure online bidding system.

Online Bidding System

Electronic auction research started in the mid-1990s and has become an e-commerce hot spot. Stubblebine and Syverson [127] suggested an open-bid auction scheme that is based on a hash chain technique. They presented an online English auction in which bids are assessed equally and the auction closes reasonably without specialist trusted parties. However bidder anonymity was not satisfied by this scheme. This scheme is often questioned [128] because the scheme has a single auctioneer (which could be corrupted) that ensures against the selective blocking of bids on the grounds of their volume and the early selective termination of an auction.

Nguyen and Traoré [129] proposed an open-bid auction, which holds a bidder's privacy using a slightly changed group signature scheme. This protocol thus suffers from the following limitations in group signature systems: (i) a group manager (GM) operates as an auction Manager (AM) in their scheme, and a group member corresponds to a bidder. Anonymity is the second challenge; (ii) as GM has a special authority, the party signature does not fulfill GM's confidentiality; (iii) it is very challenging to withdraw a bidder because each bidder is distributed with a membership credential.

A similar study can be found in the work of Omote and Miyaji [130], in which they presented a bulletin board for reducing the computational load for both bidding and verifying a bid. However, for security reasons, their method did not publish any bidder information because of potential security breaches to privacy issues. As the system would not disclose

any bidder information for security purposes, it would violate the purposes of anonymity, fairness and unlinkability among various auction rounds, and other characteristics that are required in an English auction protocol [131]. Unlinkability refers to the inability to find a relationship between two observable system entities, so it should not be possible to tell if multiple transactions are from a single user, since it is simple to infer other details about the user once all the transactions related to a user can be connected. Anonymity relates to the state of being unidentified and anonymous. Although the Bitcoin BC guarantees pseudonymity by providing pseudo-identity as a support for the anonymity of a user's identity, it doesn't provide users with non-linkability security for their transactions [132].

Some auction schemes are built with the assumption that trusted third parties (TTP) exist [133]. Such schemes have considerable advantages when it comes to computation and communication loads, but the shortcoming of this type of scheme is that it is difficult to create a fully reliable TTP. Huang et al. [134] proposed a privacy protection scheme, using Paillier encryption to encrypt bids and calculate cipher text by a third-party agent. Another secure auction system was proposed by Chen et al. [135] that uses a secret sharing scheme, resulting in secure auctions that do not disclose other information than auction results. All the above schemes incorporated a third-party agent to assist the auctioneer to complete each auction. But it is unknown whether the third party is credible. The auctioneer and an agent may collude to make an unfair profit from an auction. The schemes have risks to security and cannot guarantee the successful execution of their auctions.

Blockchain for bidding systems

BC has been facing numerous challenges in large database structures and industry data due to its linear data structure. Its sequential nature makes it a single-user operation which affects its performance [136]. The existing chain structure has been widely used in almost every BC, except for a few industry projects, for example, the directed acyclic graph (DAG) that was proposed by DagCoin and IoTA [137]. According to a survey [138], DAG and tree structures are different. The main difference is that in DAG a node can have more than one parent node. Additionally, the presentation of data in a tree structure gives a representation of the overall system in a manner that is easier to

understand and communicate. As there are challenges with using a system representation of graphs, especially if the graph is complex, presenting information in a data tree reduces this problem. Groß et al. [139] is a scalable local grid system for smart communities that makes use of a blockchain with tangled data structures to get around problems like high transaction fees and constrained throughput. However, since it differs from the linear data structure of the conventional blockchain, there are still many unanswered questions about this blockchain.

A new protocol of BC is suggested in [136], where the authors argue that merely tweaking BC parameters is not sufficient to significantly improve the performance and scalability of BC. Nonetheless, due to the advantages of BC's decentralized nature and accountability properties, several bidding protocols have recently been deployed on top of BC, as with the aid of a BC framework it is easier to eliminate vulnerabilities introduced by a third party and the data on the ledger can be reviewed and validated by everyone.

Blass and Kerschbaum [140] implemented a bidding auction on BC, where they developed an algorithm for two-party comparisons, which was conducted between any pair of bids to determine the outcome of the auction in cipher text. However the computation overhead is very large for individual users.

There is a lack of transparency in traditional bid management systems [141]. Usually, a third party publishes the bidding rules, bidding time periods and the time to announce the winner. All the interested parties submit their bid price within a certain time. After a period, a third party announces the winner. It is difficult for bidders to fully trust the third-party intermediary, as the third party may leak sensitive information or misconduct the procedure for their self-interest [135].

In recent times, online auctioning has taken on a considerable interest in the community of security researchers. Franklin and Reiter [142] were amongst the first to discuss the security of an electronic auction. They covered many fundamental topics and merged cryptographic primitives, such as secret sharing, digital cash and multi-casts and implemented their own primitive verifiable signature-sharing scheme. This primitive allows a

signed message holder to share their signature with a group of users. The signature can be restored only by group members, even when some of the group members are unreliable. The only downside to this strategy was that at the end of the bidding cycle, all bids were open.

Later Harkavy et al. [143] proposed a solution to this issue. Bids included in their protocol were not released but were contrasted with the highest bid by using distributed multi-party computation. However their work only considered the sealed-bid auction scenario. That may be due to the belief that sealed-bid auctions face more technological challenges to protect the privacy of bids than open-bid auctions. In addition, the existence of online commercial auctions appears to blur the lines between different auction types. For example, eBay auctions are primarily English-style auctions. The involvement of proxy bidding and a fixed auction termination time reflects the Vickrey style [144]. However the latest auctioning model of eBay now conceals the identity of each bidder, which further detracts from what people consider a typical auction [145]. Nonetheless, little work has been done on the security of open-bid auctions and thus open bid systems and their security components, have been a challenging and new opportunity for study [146] [147].

Recently Opensea [148] provided an auction system based on BC. However Opensea suffers from the following drawbacks: all the bids are public, bidders may bid on any sum, not necessarily higher than the maximum bid and sellers can terminate the auction at any time. This system does not guarantee non-cancellation of bids, neither that the highest price always wins.

Braghin et al. [149] demonstrated how various types of auctions can be built on top of Ethereum BC and analyzed the implementations in terms of cost and time efficiency. They did not apply any formal testing methods to identify possible bugs and flaws in their proposed framework. Moreover, security and privacy issues were not addressed.

Later Lafourcade et al. [141] proposed an open-bid auction and illustrated that a bidder's privacy in a blockchain-based e-auction protocol is a significant concern, as every transaction inside the blockchain system is visible to everyone and may be examined and

analyzed to link actual identities. The Ethereum-based English auction system known as Auctionity uses ECDSA and non-fungible tokens to enhance security and anonymity.

Tso et al. [150] designed a distributed e-bidding system with smart contracts. The study highlights that BC can be used to improve the transparency of user information and transaction data. Moreover they explored different encryption schemes to satisfy security requirements. They used Ethereum smart contract and utilized Proverif to evaluate Auctionity security. However this protocol does not provide privacy for bidders.

According to Martins et al. [151], customer bargaining and electronic procurement can be offered through a decentralized market. They suggest a decentralized e-commerce platform powered by blockchain, where users can place orders in supply chains. They developed a smart contract architecture that includes listing, aggregation and auctioning processes, as well as contracts that employ tracking tokens for listing interaction. Reverse auction bids are used as the pricing method, which is inappropriate for the context of food and agricultural output from underdeveloped nations.

Omar et al. [152] suggested a solution based on the Ethereum blockchain that makes use of Ethereum smart contracts, decentralized storage systems and a trusted Oracle to record interactions between auctioneers and bidders in order to guarantee data transparency and integrity and to eliminate financial intermediaries.

Nodehi et al. [153] proposed an extensive Enterprise Blockchain Design Framework (EBDF), together with a design for an ecosystem, for a group of European e-Procurement platform providers headed by the business Vortal. The proposed blockchain ecosystem, which is still in the design phase, would support the users of the current platforms by enabling communication between buyers and sellers on various platforms. They demonstrated the advantages of the proposed framework with a proof of concept implementation using Hyperledger Fabric.

Summary

The literature on open-bid auctions using BC is small. Although there are several works

that describe the use of BC bidding systems, there is a common limitation with many academic works on enterprise blockchains, because many projects are still in the very early stages or at the proof of concept implementation. As a result, an appropriate formulation for fulfilling these gaps is required to build a transparent, non-repudiative e-bidding system with high throughput and an efficient searching mechanism.

The aforementioned challenges are addressed by the proposed privacy-preserving solution (see Chapter 6). The concepts of a tree data structure, Elliptic Curve Cryptosystem (ECC) and dynamic accumulator have been proposed for an open-bid auction system in order to achieve high performance within auctions. The tree data structure makes the system more efficient, ECC provides strong security and the dynamic accumulator will secure the privacy of bidders. A thorough security and privacy analysis are also provided for the proposed solution, along with extensive performance evaluations.

2.4 Summary

This chapter offers a comprehensive introduction to blockchain technology, encompassing the block structure, transaction structure, mining, cryptographic primitives and blockchain types for in-depth technical discussions in the subsequent chapters. Afterwards, the literature on blockchain solutions for access control, data sharing and scalability problems is reviewed. The requirement for fined-grained access control solutions for SCM is highlighted by the fact that the existing solutions mostly suffer from data privacy and scalability issues while sharing demand/order data. In the following section, the key consensus algorithms deployed in current blockchain systems are discussed, with an emphasis on SCM requirements. Solutions for blockchain-based data-sharing models in supply chain management have also been covered. Following that, the blockchain's potential for e-auction, with a focus on open bidding systems, is examined. At the end of each intermediate section, there is a comparative analysis of the existing literature utilizing tables. Finally, the challenges and limitations in the reviewed literature are compared and covered in detail in chapters 3 through 6.

Chapter 3

AccessChain

In this chapter, *AccessChain* is developed, which aims to address the data privacy challenge as outlined in Sections 1.2.1 and 2.2.4. *AccessChain* is a supply chain management (SCM) access control framework, that is based on an attribute-based access control (ABAC) model that restricts access to competing parties while allowing for network scalability. Despite the fact that some blockchains can restrict participants to read and write data, the blockchain's transparency makes protecting sensitive data challenging. This proposed *AccessChain* model has two types of ledgers in the system: local and global. Local ledgers are used to store business contracts between stakeholders and the ABAC model management, whereas the *global ledger* is used to record transaction data. *AccessChain* can enable decentralized, fine-grained and dynamic access control management in SCM, when combined with the ABAC model and blockchain technology (BCT). This chapter's experimental results illustrate that high throughput can be achieved in a large-scale request environment while maintaining data privacy and sustaining a scalable network.

3.1 Introduction

A blockchain is defined as a series of blocks that hold tamper-proof data transactions. Nakamoto [19] first proposed blockchain as a way to store and share Bitcoin transactions. Each blockchain offers a decentralized information exchange without the need for a mediator. Apart from digital currencies, blockchain can be utilized in a variety of SCM applications [154], and in doing so, it has established a new paradigm for supply chain data integrity and transparency. The decentralized framework of blockchain can be used to provide reliable data transmission for SCM. Because supply chain operations typically involve multiple stakeholders (e.g., suppliers, manufacturers, third-party vendors and retailers), having a transparent, immutable operational ledger may be advantageous. By introducing BCT into an SCM, stakeholders can gain competitive advantages through enhanced data visibility, automated purchasing and payment processes, lower risk of errors and protection of a supply chain against counterfeiting.

The data on a blockchain can be divided into two categories: user identification and transaction records. However transactional data (i.e., manufacturing records, supplier information and consumer demand data) are valuable assets for any SCM; thus their encryption or privacy is vital [155]. It is established that on a public ledger, transaction data is accessible to all participants, whereas on a private ledger, read and write permission is determined by a permissioned blockchain. Participants who may share resources and have reading and writing privileges can be restricted by using a consortium blockchain. Despite the fact that these features have increased the technical acceptability of blockchains in the supply chain sector, data privacy remains a concern. In some cases, regardless of the fact that access to the ledger is restricted, anyone that matches the accessibility criteria can still access data from the ledger. Thus businesses' objectives may be jeopardised if data is shared with a huge number of participants without data privacy [156]. Limiting the network to members that are part of a specific production structure is an option, but this limits the supply chain's flexibility. Thus this work emphasises protecting data privacy by employing a fine-grained access control framework when transferring business data, such as locations, manufacturing materials and demand data. To enhance data sharing

in a supply chain, a balance between data accessibility and data privacy is essential [157], as the possibility of data breaches makes participants more reluctant to share personal information. In addition, inappropriate data sharing can also significantly cost a business, in the form of fines for privacy invasion.

However since data is exchanged across several stakeholders, finding this balance is extremely challenging. Such businesses are diversified, which means they operate in various ways and employ various data models. To make things work, they must first agree on who has access to their data and how they can trust one another before publishing it. There has been a significant amount of research on supply chain data privacy. For example, Ferdousi et al. [158] proposed a distributed ledger that provides pseudonymity. However due to persistent user IDs and a one-to-one mapping of business operations to publicly accessible transactions, their approach is vulnerable to correlation. Data ownership and the privacy of sensitive data are issues that must be addressed; therefore data access is a crucial concern. The subject of access control is important in SCM, along with other security concerns. Access control is a critical resource restriction tool that has been widely used in a variety of applications, such as in the internet of things (IoT) [155], healthcare data [159] and cloud Computing [160]. Access control can be considered a kind of security to ensure that only permitted businesses with access control policies can access the required information. A robust access control framework typically covers three major security concerns: Accountability, Authorization and Authentication [161]. Access can be enforced through many types of access control models: discretionary access control (DAC), mandatory access control (MAC) and Role Based Access Control (RBAC), which are examples of traditional access control models. These models can give fine-grained access control over resources. However their nature is extremely centralized, with the drawbacks of single-point failure, difficulty in scaling and low throughput for large-scale dynamic frameworks. It is challenging to fulfil AC needs in an SCM framework with centralized AC. Section 3.2 further explores these models. To overcome these challenges, attribute-based access control (ABAC) is used to enforce access restrictions based on the attributes of the subject, resource, action and environment involved in an access event. ABAC, also known as policy-based access control, first separates the user, resource, permission and

environment attribute, then fuses their relationships and eventually converts permission management into attribute management, resulting in a fine-grained and dynamic access control framework. Despite blockchains having several advantages, scalability remains a major bottleneck when it comes to implementing one in a supply chain setting. Yet it seems to be critical for blockchain acceptance in large-scale networks with large numbers of participants, such as SCM. As a matter of fact, as the number of nodes in a network grows, the transaction volume in its corresponding supply chain also tends to grow. Several research works have addressed the scalability problem by using various scaling approaches, such as Sharding, Directed Acyclic Graphs and Lightning Network [162]. The concept of sharding is one of the most promising approaches to the scalability challenge. Sharding [163] is a method that divides a blockchain into numerous shards and allows participating nodes to execute and store transactions from only a few of the shards. To maximize throughput, several shards can process transactions in parallel. High flexibility, high throughput and high scalability are just a few benefits that come with this division. The Scalability Trilemma [164] is the widely held notion that, at any given time, decentralized networks can only provide two of three benefits: decentralization, security and scalability. Achieving these three features simultaneously in current blockchains is quite challenging.

3.1.1 Chapter Contributions

The following are the major contributions of this work:

- A multi-blockchain data privacy-preserving framework called *AccessChain* is proposed to control read and write access to the blockchain using a tiered architecture.
- Addressing the challenge of data privacy, the local access ledger has been abstracted into a business contract to provide fine-grained data access control.
- To maximize scalability, two distinct global and local ledgers are used to store business contracts and business operations, respectively.

- The architecture's performance is evaluated in terms of time cost, latency, throughput and scalability. The experimental results show that the proposed architecture can effectively be used to ensure a secure, scalable and efficient data exchange among supply chain entities.

3.1.2 Chapter Organization

The rest of this chapter is organized as follows. Section 3.2 details relevant background knowledge. In Section 3.3, the proposed *AccessChain* model is presented, and details of experimental evaluation and the threat model of the proposal are in 3.4. Finally, Section 3.5 gives the conclusion.

3.2 Preliminaries

In this section, a few widely used access control models are briefly discussed. Additionally, it provides an overview of the ABAC model, which forms the basis of the proposed data access control.

3.2.1 Access Control Models

Access control systems can be used in a range of areas and at various levels in software and hardware. This section presents a literature-based argument for choosing ABAC as a framework over RBAC and DAC. There are a variety of access control models to choose from, however all types of access control can be traced back to one of the three basic models: DAC, RBAC and ABAC. To regulate how users access resources, each model employs a different set of techniques.

- **DAC:** is a type of access control that allows or limits user access depending on an access policy set by the resource's owner. A DAC framework is established using

user credentials, such as login information. DACs are discretionary since each user can provide other users access to authenticated resources or data. To put it in other words, the user determines their own resource access privileges.

- **RBAC:** grants access to users depending on their responsibilities or roles within a network. Users are only given access to critical data. Numerous characteristics, such as authority, responsibility and job expertise, can determine access. Furthermore, access to resources might be restricted to certain operations, such as reading, writing, or updating.

- **ABAC:** offers access to users based on a collection of attributes. Permissions can be based on the user's type, location, department and other attributes allowing for a more straightforward control structure that reflects the physical aspects of the network [165]. ABAC simplifies the expression of a comprehensive, sophisticated access control policy, by examining a user's attributes information that is already known and frequently kept in a system.

DAC vs RBAC vs ABAC: Although DAC is easy to implement, it has significant drawbacks that make it unsuitable for use in a complex SCM environment. The main difficulty is that due to the enormous number of generated log entries, monitoring is challenging. Despite the fact that RBAC is a popular choice for organizations, the RBAC technique has a number of limitations, including the inability to construct rules using parameters that are unknown to the framework [166]. Furthermore, because the RBAC paradigm is primarily focused on static organizational roles, RBAC designs provide issues with dynamic demand access control frameworks. No multi-factor decisions are supported by RBAC. The ABAC approach, on the other hand, offers significant advantages that are tailored to the approach in this chapter, such as ABAC may automatically modify authorization and once everything is set up, there is less overall management required [165]. When properly configured, it is also reliable. Most importantly, ABAC facilitates access control actions without the user's prior comprehension of each resource [166] and has been widely used in the literature [73, 167, 168]. RBAC and ABAC differ significantly in terms of their static versus dynamic nature. RBAC is more static and employs role-based access con-

trol, whereas ABAC is more dynamic and uses relationship-based access control. ABAC is based on attributes, which can change frequently, but RBAC is based on roles, which are usually quite static inside a network. RBAC allows one to define access controls in broad strokes, whereas ABAC allows for greater refinement. An RBAC system grants access to all employees, whereas an ABAC policy only grants access to administrators in some specific department or region. Table 3.1 provides a comparison that takes relevant parameters into account, although it is not a detailed list of all of the required characteristics for configuring the models, but rather highlights the key criteria that come from each model. The models are contrasted in terms of scalability, performance, granularity, flexibility, security and custom permissions, which are the primary needs of any access control mechanism. A detailed description of the comparison criteria can be found at [169, 170]. This table can be used to draw the conclusion that ABAC is the appropriate access control technique for the supply chain application since it prioritizes scalability and flexibility. Access control for supply chain systems should be scalable, flexible, efficient and trustworthy and must be sufficient to secure the supply chain and its components' privacy and integrity. Usually, a supply chain's applications involve multiple stakeholders, like manufacturers, producers, transporters, retailers and customers, and it is critical to developing trust between these entities. When it comes to maintaining participants' identities, the ABAC approach offers a lot more freedom and is useful for supply chain access control because it allows stakeholders to remain anonymous. Furthermore, ABAC allows supply chain managers to apply access control policies to an unlimited number of participants without having any prior knowledge of them. One of the biggest benefits of ABAC is that it makes it simple to add new users.

3.2.2 ABAC Model

ABAC is a type of logical access control that includes access control lists, role-based access control and its own method for granting access based on attribute analysis. ABAC regulates system resource access by comparing policies to user properties, such as subject (user), object (resource) and environment. When making ABAC decisions, both subjects

Table 3.1: Comparison of access control models.

Characteristic	DAC	RBAC	ABAC
Scalability	Yes	No	Yes
Performance	Low	High	High
Granularity	High	Low	High
Flexibility	Yes	Yes	Yes
Security level	Low	High	High
Custom permissions	Yes	No	Yes

and objects have attributes and the conditions of the environment may be taken into account. In essence, this means that it may utilize key-value combinations, like Role = Production Manager and Category = Manufacturer, to define rules in eXtensible Access Control Markup Language (XACML).

ABAC is defined as follows:

- **Attributes** are traits of a subject, an object, or the conditions in which they exist. A name-value pair provides information for attributes.
- **Subject** is a user, it can be a human or a device, who makes requests for access to execute actions on objects. One or more qualities are given to subjects. Assume that subject and user are synonymous for the purpose of this article.
- **Object** is a system resource. It can be any requested resource as well as anything that a subject can use to complete an operation; including data, services and devices.
- **Environment** is the context in which an access request is made. The time, date, location and currency risk are examples of environmental features that are independent of the subject or object.

3.3 AccessChain Framework

AccessChain, is a blockchain-based access control framework for SCM. As shown in Figure 3.1, it consists of two key components: *access point ledger* and *global ledger*. Each key component is explained below.

3.3.1 Architecture

A blockchain-enabled supply chain model consists of suppliers, manufacturers, distributors, retailers and end users, which records supply chain transactions like production and sales on a blockchain. A retail supply chain has been considered to demonstrate our access control framework. As a result of large supply chain networks and the massive amount of big data they produce, scalability has been a major issue in the integration of blockchain. To boost scalability the suggested underlying network is built on shards that run in parallel and are responsible for scaling the blockchain while keeping security guarantees. The two main components of the framework's sharding are depicted in Figure 3.1, as follows: (1) The global chain; (2) the local *access point ledger*. Our framework organizes shards based on geographic zones. Each of these local chains, called *Access ledger*, is a public blockchain network with administrators in charge of registering the business contracts of participants. A supply chain entity must first register with its region's certification authority (CA) and get an identity, which verifies its digital profile on the ledger. In access point ledgers, each participant's access privileges are defined using the ABAC model. After they have established their identities and business contracts, the participants send a join request to the *global ledger* administrator. The stakeholder must authenticate their trade identities without disclosing them. The *global ledger* administrator validates and approves the stakeholder's registration request in *global ledger*. The stakeholder can then log business transactions on *global ledger*. For the purpose of evaluating *AccessChain*, the retail business network being developed includes:

- Participants: Include primary supplier, manufacturer, distributors and retailers.

- Business Contract: A business contract is a legally binding written agreement between two or more stakeholders in a business. These contracts specify terms, such as when a task will be accomplished, what products must be supplied and when payment for goods or services is due. For this proof of concept, two business contracts were utilized.
- Contract Owner: A contract owner is the one who owns the contract and can utilize it to establish business relationships. In the prototype, each stakeholder can own a contract; for example, a retailer can have a short-term contract with other retailers.
- Validator/administrator: A Validator captures all business contracts and access rules for read and write access from businesses, authenticates them and authorizes them to be added to the local blockchain.
- Certificate Authority: When a stakeholder first registers, the certificate authority issues digital certificates for a private key-public key pair established by the stakeholder. The certificates ensure that the key belongs to the stakeholder.

For each subject, a subject $ID(ID_{\text{user}})$ is retained, as well as a set of its attributes and values and the same for Object attributes. The Attribute *ID* field is essential for storing attributes in a business contract and retrieving them later for authorization decisions. For any business contract access policy, having subject (user) attributes and object (resource) attributes, the subject must have a valid access right to the resource, have a valid Identifier (ID), be in an active status and the subject location must match with the resource location. The access permission will be declined if one of the subject's attributes does not match the policy requirements.

3.3.1.1 Access Point Local Ledger

Access point ledger is committed to providing business contracts and access control policies to all participants. The IDs on *access point ledger* can be verified by a centralized authority. In blockchain-based frameworks, public-key cryptography is frequently used to

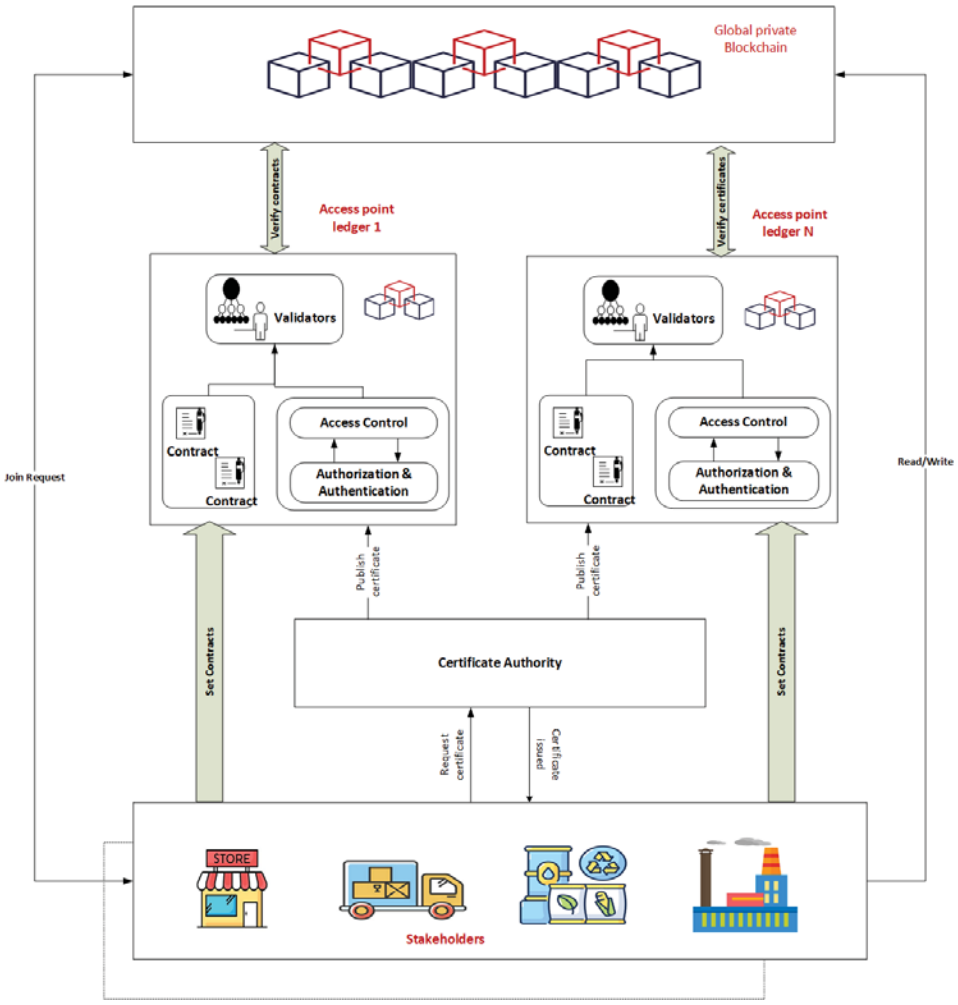


Figure 3.1: AccessChain framework

authenticate users in a network. As part of this design concept, cryptographic algorithms such as SHA256, digital signatures and state of the art elliptic-curve cryptography (ECC) [171] are used to digitally sign transactions. In *access point ledger*, business contracts are created using an approach that is similar to that utilized in the real world. Consider a retail supply chain, which is made up of several businesses. As discussed in Section 3.3.1, there is no need to communicate every piece of information among several businesses. For example, it is not necessary for supplier 1 and supplier 2 to be on the same business contract. To protect data privacy and security, only the relevant entities required for a

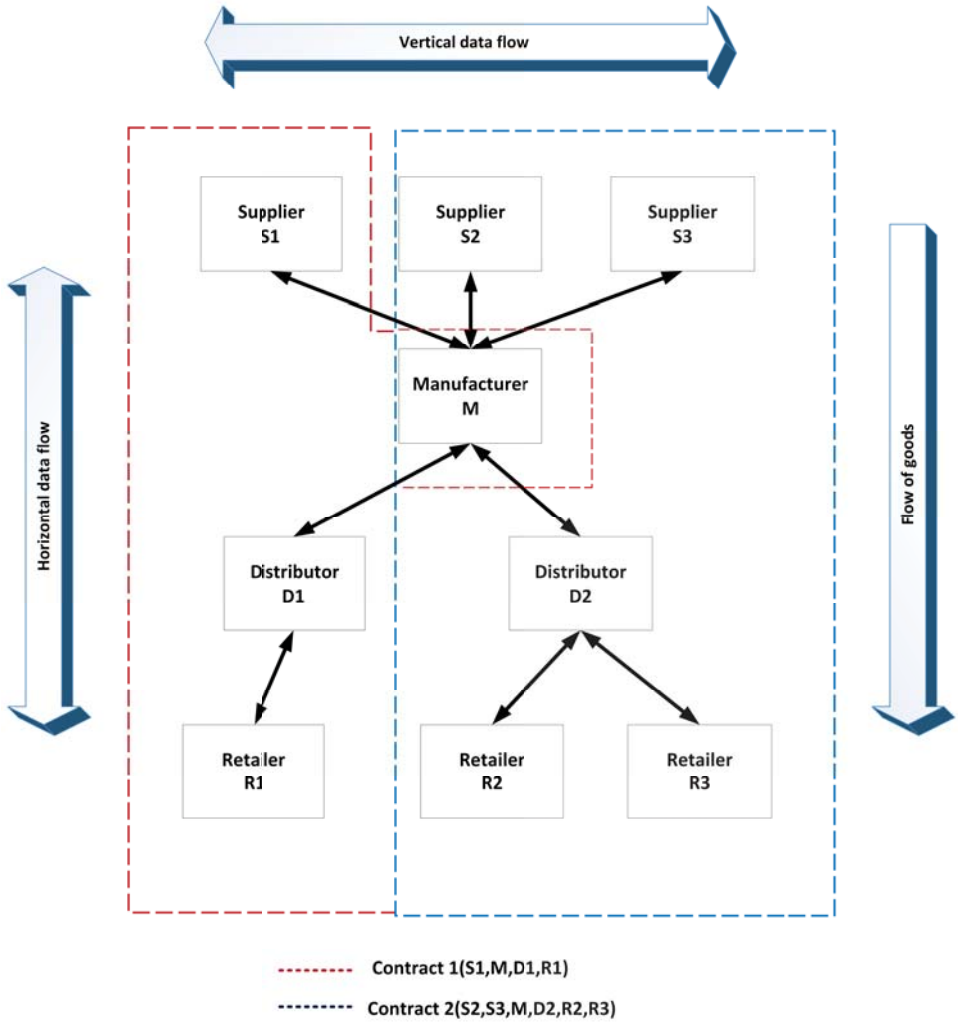


Figure 3.2: SC-contract: Stakeholders in a blockchain-based data access contract.

swift business transaction are included in the contract. Furthermore, every contract must contain an expiration date.

$$Contract \rightarrow [Policy|ID_{\text{user}}|ExpDate|Sig_{\text{Owner}}] \tag{3.1}$$

where *policy* is the user access model, ID_{user} is the identifier of the participants, Sig_{Owner} is the signature of the contract owner and *ExpDate* represents the contract's expiration date. Only related businesses are included in a contract and have read/write, update/delete

permissions. Figure 3.2 depicts a number of businesses in a supply chain that are linked via contracts. There are two ongoing contracts: in *contract 1*, the manufacturer has supplier s1 along with distributor D1 and retailer R1 (S1,M, D1,R1). The Manufacturer has another ongoing *Contract 2* with other businesses (S2, S3, M, D2, R2, R3). There is no contract path from S1 to D2, this indicates that S1 has no access to data from D2 and vice versa. In summary, *access point ledger* stores credentials, business contracts and access control rules that are not accessible to the general public and can only be accessed with the contract owner's, i.e. the manufacturer's, approval.

3.3.1.2 Global Ledger

A *global ledger* is where the actual business transactions are recorded and it contains a full history of transactions. When a request to read/write or update/delete is made, validation is carried out by ensuring that the participant has the required permits and that the contract is still active before granting permission. The *ID* of the *access point ledger* where the contract for the access is held, must be kept as supplementary data with each transaction to the *global ledger*. The global ledger's transaction structure is defined as follows:

$$Transaction \rightarrow [ID_{\text{user}}|ID_{\text{ledger}}|Sig|data] \tag{3.2}$$

where ID_{user} and *Sig* is the signature and public key of the participant and $\text{ID}_{\text{ledger}}$ is the location of *access point ledger*. The global ledger utilizes the *RequestAccess()* function to verify the permissions from the *access point ledger* before logging a transaction and the contract's validity is double-checked using a function known as *ContractValidation()* (explained in algorithm 5). Permission is given if both functions produce a positive return. Otherwise, access is denied through *RevokeAccess()*. The main function of access control management is Algorithm 4. To begin, it obtains the attribute set specified by *GetAttribute()*. It will provide an error message if the returned result is empty, indicating that there is no contract to support the request. If the returned result is not empty, it means that at least one contract will be obtained. Finally, it calls the *policy()* function to verify that the desired user *ID* and permission are in the contract and returns access to

the user, failing which an exception is returned, where *policy()* contains the ABAC access privileges for the specific contract.

Algorithm 4 RequestAccess(): Check user's privileges.

```
1: procedure RequestAccess(user, permission)
2:     sub, obj, env, opr ← GetAttribute(ABAC)
3:     contract ← C_1, C_2, ..., C_n
4:     for all con ∈ contract do
5:         if con = null then return error
6:         else
7:             policy ← getPolicy()
8:             if policy ∈ (UID_user, permission) then
9:                 Access ← getAccess(UID)
10:                grant ← TRUE
11:            else
12:                grant ← FALSE
13:                Access ← RevokeAccess(UID)
14:            end if
15:        end if
16:    end forreturn grant
17: end procedure
```

Additionally, the *global ledger* must validate the contract's validity, as stated in Algorithm 5. To meet the requirements, a legitimate contract must not be expired and not be cancelled by the administrator.

3.3.1.3 Mining

Mining is the operation of adding new transactions and contracts to the ledger. To authenticate identities and authorize nodes to join/rejoin the network, BFT protocols for permissionless blockchains like Proof-of-Work (PoW) [19] and Proof-of-Stake (PoS) [23]

Algorithm 5 ContractValidation(): Check contract validity.

```
1: procedure ContractValidation(C_1, ..., C_n)
2:     contract ← C_1, C_2, ..., C_n
3:     valid ← TRUE
4:     for all c ∈ contract do
5:         if c ∉ (ExpDate, status) then
6:             valid ← FALSE
7:         end if
8:     end forreturn valid
9: end procedure
```

are widely applied. It is crucial to avoid shard hijacking in order to ensure the security of a sharded blockchain system. In order to prevent malicious nodes from controlling a shard, a node cannot be permitted to select the shard it wants to join. Therefore, the concept of regional chains is provided and joining the network requires utilizing the location property. As soon as shards are constructed, PoW consensus algorithms are utilized to achieve consensus and validate transactions within regional chains. PoW systems have received a lot of criticism, primarily because of their high energy consumption [172], however they are used as they help safeguard a blockchain from potential attacks (outside the scope of the model). It is worth noting that mining in the *global ledger* is based on the concept of proof-of-authority (PoA) [173], which relies on trustworthy and recognized validators to generate blocks, rather than their own computing power. Through mining, the network creates a tamper-resistant state on the blockchain, which is essential for secure access control for the SCM network.

3.3.1.4 Storage Consumption

It is not efficient to employ permanent data storage in blockchain as a data storage system because it is quite expensive [174]. For instance, Ethereum uses optional data storage and only retains the information needed for essential operations. Traditionally the cost of

any computer program is determined by its spatial (the amount of data storage required) and computational complexity (the amount of computing power required). In order to reduce the storage consumption in the *AccessChain*, local chains have been used, since local chains can help to reduce the cost of storage on the global blockchain. Thus *AccessChain* is a viable approach that uses local chains to provide a cost-effective blockchain framework.

The prototype solution is not built on an enterprise platform (i.e. Ethereum or hyperledger Fabric), making it impossible to illustrate the costs associated with contracts' storage and accesses. In future work, it is intended to do additional real-time testing by deploying the proposed solution on a test hyperledger network. This offers a more realistic setting to test the performance of the *AccessChain*. However, *AccessChain's* effectiveness is evaluated in terms of transaction throughput/latency (write/read performance), time cost and scalability.

3.3.1.5 Cross-shard Transactions

System performance and communication time are both impacted negatively by cross-shard transactions, which are significant contributors to the overhead of data transmission among shards. There are two main methods for handling cross-shard transactions in sharded blockchains. The first solution is to build a full-mesh connection between nodes (for example, OmniLedger [98] and RapidChain [99]). While this method eliminates a computing bottleneck in the main chain and splits out storage, it can also increase communication costs and bring security risks. Due to the fact that the transactions are broken up into different chains, if a shard is taken over by an adversary other shards will no longer be able to authenticate transactions that rely on the targeted shard. The other approach is to store a global chain, such as Elastico [175] and Ethereum 2.0 [21]. The final block is recorded in a *global ledger* after the nodes in each shard finalize and agree upon local transactions. In Ethereum 2.0, the global chain ensures that all transactions in all shards are in sync and eliminates any transaction that has been double-spent. While the

global chain manages cross-shard transactions, the shards boost parallelism. The system retains consistency even when one shard is under the control of an adversary and this strategy minimizes the overhead of data migration when processing cross-shard transactions in comparison to the first approach. Therefore, the proposed approach employs the Ethereum [21] technique, where each local chain independently manages and processes separate subsets of transactions, while the global chain preserves the overall sequence of transactions.

3.3.2 Access Control Model Design

When integrated with the ABAC model and the specifications of data provided by the contract owners, the ABAC access control policy model is as follows:

$$policy = \{S, O, P, E\} \tag{3.3}$$

$$S = \{ID_{\text{user}}, role, level\} \tag{3.4}$$

$$O = \{data, ID_{\text{ledger}}\} \tag{3.5}$$

$$P = \begin{cases} 0, & \text{Denied} \\ 1, & \text{Granted} \end{cases} \tag{3.6}$$

$$E = \{time_{\text{start}}, time_{\text{end}}, location\} \tag{3.7}$$

Policy: The attribute-based access control policy can be expressed as $S \wedge O \wedge P \wedge E$, thereby indicating the subject's access control rules for accessing the object. It thus specifies the set of attributes that must be present in order to access the sensitive data.

Subject Attribute: It denotes the subject's attributes, i.e., the subject's identification and characteristics that allow them to perform an access request, such as userID (unique identifying user), role (user role) and level.

Object Attribute: It represents the object's attributes, i.e., information about the requested resource, such as data and ledger ID.

Permission Attribute: It determines whether or not a user has access to resources. The values 0 and 1 represent "Denied" and "Granted," respectively.

Environment Attribute: It specifies the characteristics of the environment that are required for access control. $time_{\text{start}}, time_{\text{end}}$ and location are the three types of environment attributes. The attribute $time_{\text{start}}$ refers to the period during which a policy is established. The attribute $time_{\text{end}}$ refers to the policy's expiration date. When the current time is later than the end time, the policy will be expired. The purpose of the location is to prohibit users from accessing the system from outside the specified region.

3.3.3 Business Contract Design

The fundamental component of the access control implementation is the business contracts that provide the methods to operate the ABAC policy. The methods provided are as follows.

- Acred(): A business contract owner defines ABAC for the participating stakeholders and sends the request for adding ABAC to the validator. Validator encrypts the data with the public key of the business contract owner and then signs the request with the private key. The authenticity of ABAC must be verified by validators.
- Addpolicy(): After the validator has confirmed that ABAC is valid, the validator calls *AddPolicy()* to add ABAC to the business contract.
- Updatepolicy(): The business contract owner occasionally needs to change ABAC. The interface for updating the policy is implemented by the method *Updatepolicy()*

and the operation record for updating will also be recorded on the blockchain. Similar to *Addpolicy()*, the validator calls the *Updatepolicy()* function to replace the previous value.

- Deletepolicy(): Each ABAC policy has an expiry date and can be ended by the administrator. There are two scenarios in which deletion happens. One happens when an administrator uses this function to knowingly remove a policy. The second instance takes place when "endTime" is expired.

- Auth(): This checks to see if the user's access request and ABAC policy are valid. The user's private key is used to sign the request data, and then the validator verifies the signature to confirm the user's identity by using the user's public key.

- Attrib(): After the signature has been validated, *Attrib()* analyses the attribute data field. A data request includes only subject (S) and object (O) attributes.

- GrantAccess(); In order to request the appropriate ABAC policy in accordance with S and O, it first obtains the attribute set by *Attrib()*. If the returned result is null, indicating that no policy exists to support the request, it will immediately return an error message. If the returned result does not contain a null value, it proceeds to evaluate the request to see whether the E attribute matches the E attribute of the ABAC policy and whether the value of the P attribute is 1. The verification is successful if each attribute complies with the policy and access is granted to the user.

Multiple functions have been provided by the framework to help with access control. These functions primarily consist of registering, updating and deleting a business contract, as well as adding, updating and deleting a business contract's access control policy. The following is a breakdown of how the above-listed functions work.

i **Registering a new business contract** A number of businesses might agree on a new business deal, which is then registered by the contract's owner (i.e., the manufacturer) via the steps below.

- Create a contract for the new supply network (defining access privileges for each user).
- Register the contract on their local *access point ledger* and pay a fee to deploy the newly created contract onto the blockchain.

Notice that in our framework, any stakeholder can create a business contract, as long as they have a legitimate business deal, for example, a group of retailers may want to enter into a short-term contract with local stores to reduce shortages.

ii **Updating an existing business contract** A number of businesses might agree on updating an existing contract, which is done by the contract's owner using the steps below.

- Create a new contract to replace the existing one.
- Register the contract to the local *access point ledger*. They should not have to pay the full cost this time, however there will be a small fee to replace the previous contract on the blockchain with the newly established one.

iii **Deleting an existing business contract** Businesses can agree to terminate an existing contract that is due to expire but that they do not want to renew. The contract's owner can send a transaction to the local ledger to delete the existing contract's details.

iv **Adding and updating an access control policy** Businesses can agree to add an access control policy to a newly deployed contract, which is done by the contract's owner by sending a request to the contact's *policyAdd* service. Similarly the owner can submit a request to call the contract's *policyUpdate* function to update an existing access control policy for a specific contract. However our framework does not allow stakeholders to delete a contract's access control policy. The owner can only add or update user access privileges; if the owner tries to delete the access policy, the contract will immediately be nullified.

3.3.4 Workflow

This section outlines the workflow of *AccessChain*. The entire framework's workflow is represented in Figure 3.3. The framework's foundation step is the blockchain network's regis-

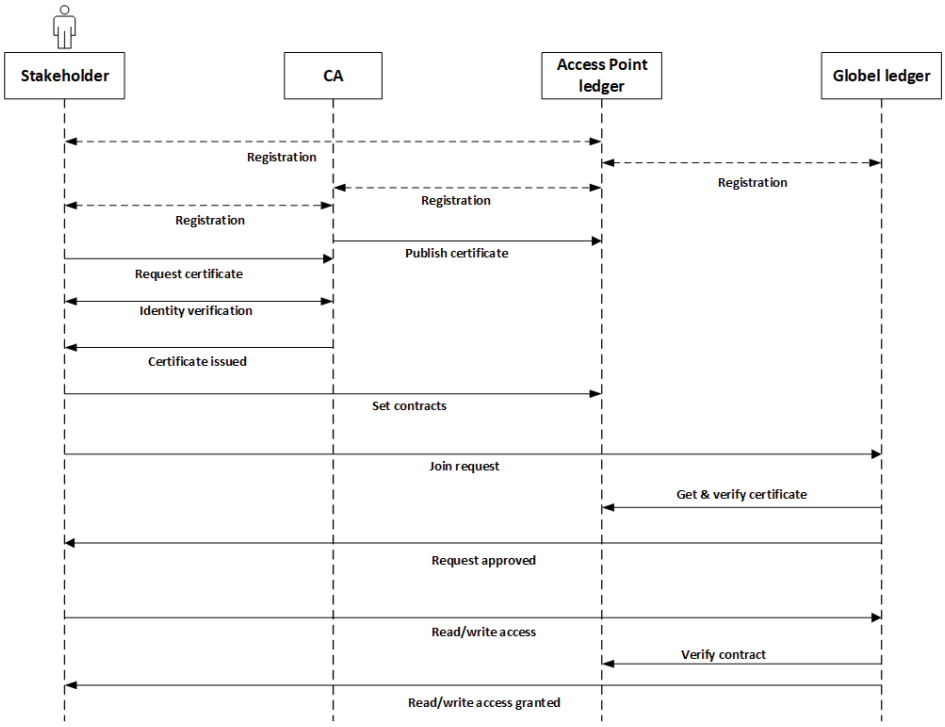


Figure 3.3: Workflow of *AccessChain*.

tration. Credentials for all participants should be created first before joining a blockchain network. Here, CA is the one that generates all certificates.

$$CA \rightarrow Credential_{\text{user}} \tag{3.8}$$

Once the stakeholders have their credentials, and they wish to write a business contract on *access point ledger*. This is done prior to logging into the *global ledger*. They create a business contract as well as an access control policy. This procedure involves the stakeholder determining and designing the access policy beforehand and then publishing it to the *access point ledger*, which is defined based on the subject (user), object (resource), op-

eration and environment parameters (discussed in Section 3.2.2). Validators are in charge of saving and managing contracts in the ledger.

$$Determine(Subject, Object, Operation, Environment) \rightarrow ABAC \tag{3.9}$$

Once the access policy is defined in the contract, the validator publishes it to the network.

$$Publish(ABAC) \rightarrow Set_{\text{Contract}} \tag{3.10}$$

To gain read/write access to the *global ledger*, a user initiates an access request.

$$ID_{\text{User}} \rightarrow AccessRequest_{\text{read/write}} \tag{3.11}$$

The *global ledger* invokes the *access point ledger* for that particular contract after receiving the request.

$$AccessRequest = \begin{cases} 0, & \text{Denied} \\ 1, & \text{Granted} \end{cases} \tag{3.12}$$

If the request is authorized, the user has access to the *global ledger* and can read and write to it. If it fails, the user will receive an error status message.

3.4 Evaluation and Results

This section analyses performance statistics to demonstrate the feasibility of the proposed approach. To be more specific, it begins by defining the testing environment. Following that, performance evaluation is explained and finally, the suggested work's security and privacy are investigated. The settings used for the AccessChain prototype implementation are listed in Table 3.2. In experiments, one to four local ledgers are used (evaluation results are below). The solution was tested on a Dell (Intel Core i7 CPU operating at 2.21 GHz with 16 GB of RAM) and developed using JDK 15 with multithreading in the Visual Studio Code experimental setting. Several hashing and encryption operations were performed utilizing the Java Cryptography Extension (JCE) toolkit. By using the Java.net package, connections between nodes were managed.

Table 3.2: Experimental environment.

Number of global ledgers	1
Number of local ledgers	4
Consensus mechanism for global ledger	PoA
Consensus mechanism for local ledger	PoW
Number of simulated nodes	200
Block time (average time needed to mine a new block)	1 s

3.4.1 Performance Evaluation

The following three performance metrics were considered to validate the effectiveness of the proposed approach:

1. Time cost trade-off
2. Latency and Throughput
3. Scalability

The proposed approach is evaluated and contrasted with a traditional blockchain (bitcoin) called "Uniform ledger". However the baseline system does not comprise local ledgers for policies and contracts; rather everything is accomplished in a single uniform ledger. The framework configuration is as follows:

- Read (get) the ledger and write the ledgers(add, update).
- Various network configurations with node counts ranging from 10 to 200.
- The workload varies between 50 and 1000 access requests in each network configuration.

3.4.1.1 Time Cost Trade-off

Time cost indicates the amount of time it takes to evaluate an access request and process it. The access response time $T(_{\mathrm{AR}})$ is calculated by using the following equation:

$$T(_{\mathrm{AR}}) = \sum_{i=0}^{n} (ARP_{\mathrm{t}} - AR_{\mathrm{t}}) \tag{3.13}$$

where n is the number of access requests, $\mathrm{ARP_t}$ is the time INTERVAL, $\mathrm{AR_t}$ is time INSTANT. Thus, $\mathrm{T}(_{\mathrm{AR}})$ calculates differences between INTERVALs and INSTANTs. The implementation is compared with an alternative uniform ledger implementation that does not involve the overhead of business contracts and access controls in the BC that are related to permission validation. Figure 3.4 depicts the average response time for the total number of access requests for *AccessChain* and uniform ledger depending on different numbers of access requests. The requests for access might be for either reading or writing. Figure 3.4 shows that our implementation has an overhead compared to the uniform ledger. This ought to be deemed acceptable given the significant additional actions required, such as the required validation and verification of access control policy and business contracts to validate stakeholder permissions across various local ledgers.

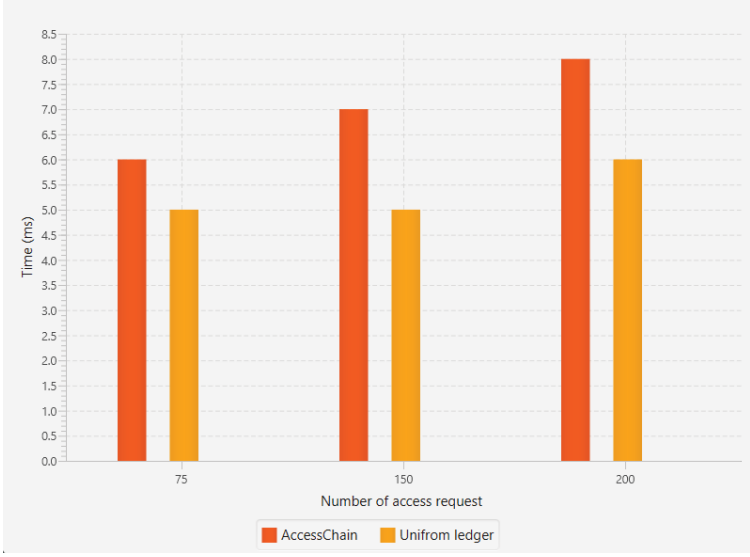


Figure 3.4: Comparison of time cost of access response for different numbers of requests.

To examine the framework's accessibility further, the time required to complete various read and write requests were calculated, as shown in Figure 3.5. To begin, ten concurrent requests were initially sent to the *AccessChain*, then a further ten concurrent requests were simultaneously added and this was continued until 100 concurrent requests were sent. The graph illustrates that write requests take longer than read requests.

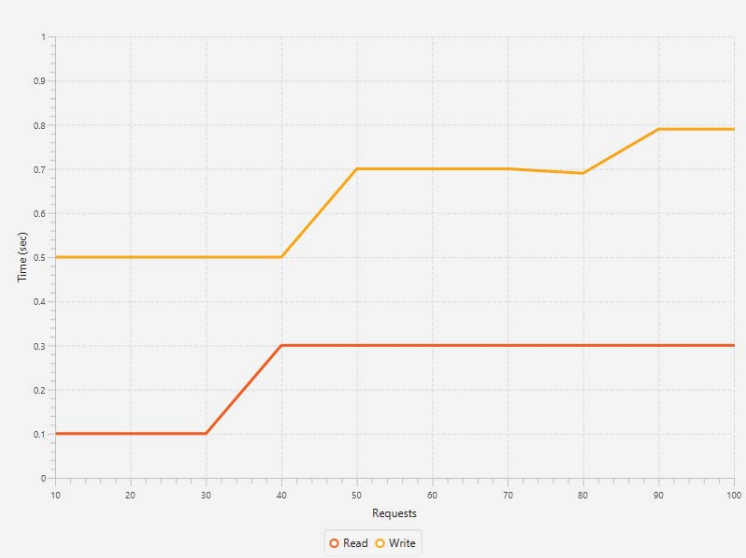


Figure 3.5: The trend of average cost time of *AccessChain* read/write operation at different numbers of requests.

The standard deviations for the read and write transactions are shown in Table 3.3. It can be observed that in the read transaction, each ledger's response time is quite similar. Therefore the standard deviation is modest. This indicates that the read transactions are more consistent across the ledgers. The RT of each ledger in the write transaction has significant variability, indicating that the standard deviation is high. This implies that the write transaction is less consistent during the course of the experiment.

3.4.1.2 Latency and Throughput

Throughput: Figure 3.6 depicts the throughput comparison between *AccessChain* and uniform ledger. For throughput computations, the number of access requests was set

Table 3.3: The average and standard deviation of read/write requests.

Transactions	L-01	L-02	L-03	L-04	Average RT	Std deviation.
Read requests	100	200	400	1000	0.425	0.349
Write requests	50	190	300	1000	0.385	0.365

between 100 to 1000. The throughput of the uniform ledger is only 5 transactions higher at 800tps, due to the Accesschain's additional overhead, which is in the range of a few seconds. Figure 3.6 further demonstrates that throughput increases linearly, as anticipated, up to a maximum throughput of about 40 tps, before it begins to decline. This denotes the threshold at which a validating peer can no longer handle the rising transaction rate. The baseline system exhibits a similar trend, with throughput beginning to decline by about 40 tps.

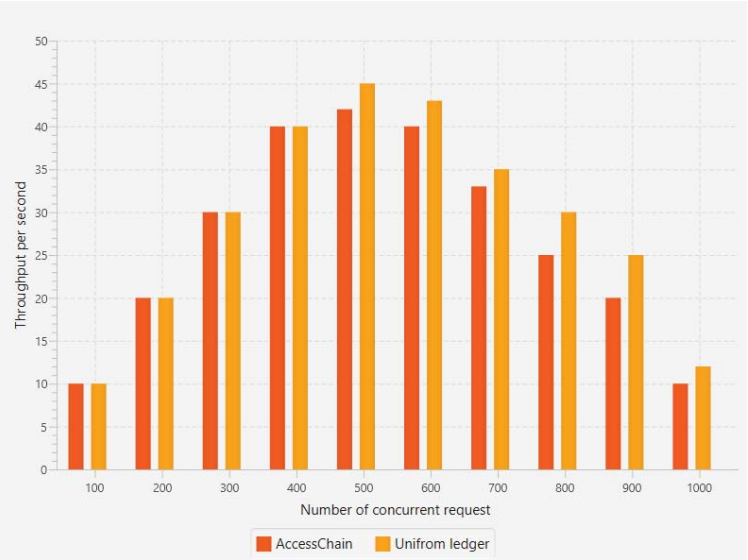


Figure 3.6: Throughput comparison of *AccessChain* with uniform ledger.

Latency: Figure 3.7 depicts the average latency comparison between *AccessChain* and uniform ledger. As the number of access requests grows, the time it takes to complete the

request grows linearly until it reaches 29 milliseconds for 650 requests. This reveals that the delay for processing 1000 requests is around 36 milliseconds, which is quite acceptable. The graph also displays the latency of a uniform ledger with a similar amount of access requests. The uniform ledger has a delay of about 50 milliseconds.

Figures 3.7 and 3.6 show that given 1000 concurrent requests, throughput grows linearly to around 42 tps and afterwards the framework goes beyond this threshold, resulting in lower throughput and higher latency. The results show that *AccessChain* can sustain excellent throughput for large-scale request settings.

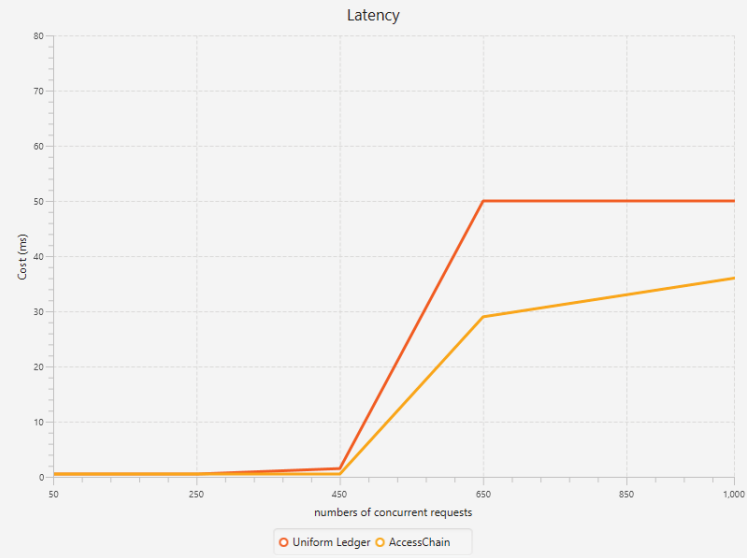


Figure 3.7: Latency comparison of *AccessChain* with uniform ledger.

3.4.1.3 Scalability

The objective of this set of experiments is to see how scalable the network is in terms of managing nodes. As the number of nodes grows in a network, the communication cost increases between ledgers. It also adds to the time it takes to verify and authenticate requests before allowing access to the *global ledger*. Figure 3.8 compares *AccessChain* to a uniform ledger when the network scales up. The average throughput for both frameworks declines as the number of nodes increases. However the average throughput of *Access-*

Chain with 200 nodes rises to 624 transactions per second. On the other hand, a uniform ledger fails to provide scalability, since throughput decreases significantly as the network grows up. To give more insight, Figure 3.9 provides the standard deviation (STDev) of the response value. In the context of performance analysis, the STDev of a transaction indicates whether or not it is stable throughout a sample. Where a smaller STDev indicates that all iterations of the same transaction have similar response times (RT). So when the transaction amount decreases, the transaction RT becomes closer and the transaction becomes more consistent. Hence the STDev of the response times can be used as a metric to see how response times fluctuate.

$$Std(x_{\mathrm{n}}) = \sqrt{\frac{n\sum x^2 - (\sum x^2)}{n(n-1)}} \tag{3.14}$$

where time is represented by x and n represents the response magnitude.

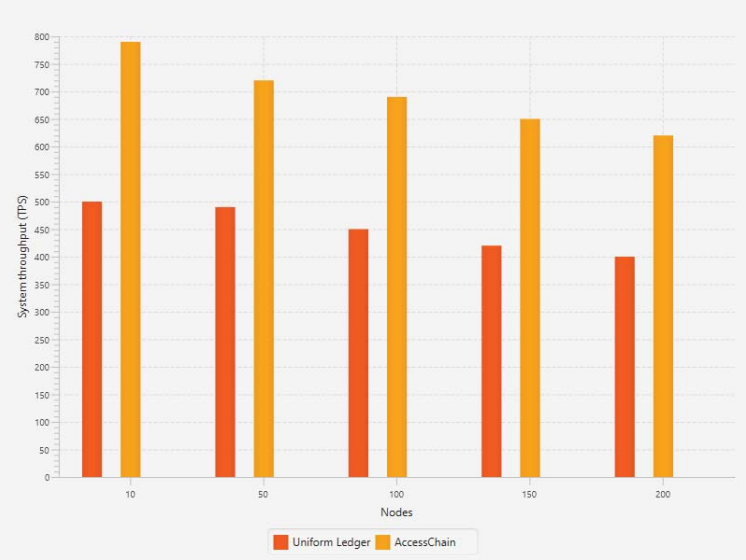


Figure 3.8: Comparison of framework throughput according to number of nodes.

Figure 3.9 presents the response time with the four concurrent transaction amounts of 200, 500, 800 and 1000. The columns show the RT of four local ledgers. It can be observed that when the number of local ledgers grows, RT grows linearly. However when switching from one to two local ledgers, the rate of growth is faster than when switching from two to three

and three to four local ledgers. In the case of having one local ledger, the average RT is 145 and the standard deviation is 7.8. RT was in the range of 141 to 149, 67% of the time. This is fairly consistent when compared to three to four numbers of local ledgers. However it becomes evident that RT has a broader range when there are more local ledgers. It's

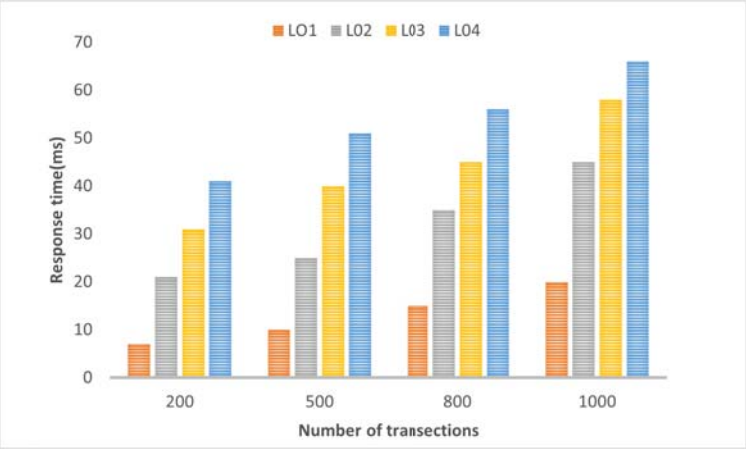


Figure 3.9: Response time vs number of transactions.

worth noting that RT is just 65.73 milliseconds, even with 1000 transactions and four local ledgers. Although sharding ensures scalability [176], the existence of many local chains increases RT. This is because although different sharding arrangements are possible, the proposed *AccessChain* shards are divided into geographical zones. As a result, a global ledger may need to traverse through one or more of the scattered locations of the shards, which can result in additional RT. Thus it can be concluded from these early findings, that our framework scales efficiently, indicating that the addition of numerous local chains shouldn't greatly raise RT. Table 3.4 represents the statics from 15 runs of this experiment.

3.4.2 Security and Privacy Analysis

Our threat model is based on the following assumptions:
Access Control and Key Management: A CA certifies each SC participant to confirm their identity. For key management, the current public key infrastructure (Elliptic Curve Cryptography) is utilized. The CA can provide security support for enrollment and transaction

Table 3.4: The average and standard deviation.

Number of local ledgers	Average	Std deviation
Local L-01	145	7.83
Local L-02	157	31.63
Local L-03	240	45.31
Local L-04	182	65.73

certificates.

System Security: cutting-edge secure encryption and intrusion detection technologies are used in the proposed framework. All data on the blockchain is encrypted with private-public key pairs. The data can only be decrypted by the user who has the corresponding private key and is granted access permissions.

The considered attacks include:

1. Creating multiple Contracts: To put more network strain on the local ledger, an adversary can create a number of contracts.

   Response: There are two controls to mitigate this, all contracts include a deposit, similar to how most public ledgers demand a minimum fee amount before a transaction is allowed. Secondly, for local ledgers, each trader has a contract creation threshold for a defined time period. Even if an adversary colludes to create multiple contracts, this can be identified.

2. Violating access token: An adversary or dishonest participant may request a token for certain data access and after it has been issued, they may try to modify the token by adding more data privileges.

   Response: The access token issued by the local validator ensures that the original access token is not modified by signing the hash of the token. Therefore modifying a token is impossible without the contract's owner's authorization. When an adversary is identified as attempting to access unauthorized data, a local validator

has the authority to revoke authorization and remove any privileges assigned to the adversary.

3. Local ledger Administration: A local ledger validator performs most of the blockchain-based operations, such as access control and setting contracts, as mentioned in Section 3.3.1. It is possible to argue that the system's over-reliance on an administrative body increases its security risks.

   Response: Local ledgers are public ledgers, therefore pairing public blockchain with the PoW consensus mechanism makes them extremely secure. Since POW makes it very challenging to alter any aspect of the blockchain because it necessitates re-mining all succeeding blocks. Additionally, the high cost of the hardware and the amount of power needed to perform the hash functions, makes it challenging for a validator to control the network's processing power. Furthermore, the blockchain's underlying cryptographic primitives, like digital signatures, hash functions and the data saved in transactions, can demonstrate the integrity of information.

3.5 Chapter Summary

In this chapter, *AccessChain* is presented. It is a blockchain-based framework with an ABAC model to assure data privacy by adopting a distributed framework to enable fine-grained, dynamic access control management for SCM. In order to solve the scalability issue, the framework helps by offering a two-tiered network design. A *global ledger* is used to record transactions, while access policies and business contracts are kept in multiple local ledgers. The framework enables a systematic approach that advantages the supply chain, and the experiments yield convincing results. Furthermore, the threat model depicts how resilient our framework is against a wide range of threats. The results of the performance monitoring also show that *AccessChain's* response time with four local ledgers is acceptable, and therefore it provides significantly greater scalability. This work implies multiple ledgers can support more complex business processes and interactions between participants in the supply chain. Different ledgers can be used to manage various sorts

of transactions, which can help to simplify procedures, lower mistake rates, and increase overall supply chain efficiency. However, stakeholders should be aware of the challenges and costs associated with using multiple ledgers, and they ought to carefully evaluate their objectives and goals before deploying a multiple ledger solution.

Chapter 4

Blockchain-coordination for SCM

In this chapter a blockchain framework is presented to address the data sharing, trust and privacy challenges of blockchain, as outlined in Sections 1.2.1 and 2.2.3. In this chapter two frameworks are developed for supply chain application. First, a supply chain bullwhip effect (BWE) base model is developed. Second, a blockchain-enabled robust information-sharing framework is simulated. Based on the preliminary results, the proposed model is extended to find trust among stakeholders, and consequently, an improved version of information sharing is developed. Information sharing is challenging and SC stakeholders may not really trust each other and hence be reluctant to share sensitive information. Considering that, this chapter proposes an improved Proof-of-Authority (PoA) consensus algorithm that will increase trust in a decentralized SC model. Extensive experiments are carried out to demonstrate the effectiveness of the approach and the simulation results clearly demonstrate the effectiveness of information sharing in a supply chain via blockchain (BC), as well as that trust between partners tends to increase overall SC efficiency and reduce BWE.

4.1 Introduction

A supply chain (SC) is a dynamic structure made up of certain entities, such as manufacturers, distributors, retailers and customers, that participate in the production and selling of a commodity. Supply chain management (SCM) is the preparation, planning, execution, supervision and tracking of SC operations, to generate profit, develop strategic infrastructure, optimize global logistics, synchronizing supply with demand and/or improve global efficiency [177]. While the SC industry has tremendous growth potential, it suffers from a wide variety of SCM problems [178], such as the need for trust among stakeholders (related to their credibility) [179], the lack of information and traceability that is increasingly demanded by end users[180] and the difficulty of handling uncertainties [181], delays or disruptions [182]. The SCM revolution is dependent on accurate and effective data management so that data gathered from SCs can be processed, incorporated and recovered, with high quality and reliability.

Trust and information sharing are two key factors for almost every SC [183] [184]. Trust has been shown to be a strong measure of an SC's success and to foster revenue growth and increased efficiency. In assessing trust and information sharing in an SC, studies typically focus on demand and inventory data [185]. The Bullwhip Effect (BWE) is one of the most well-known SC shortcomings. BWE refers to the potential of replenishment orders to increase variability, as its implications are processed through a SC. The information in such a chain is distorted as smooth final consumer demand trends are turned into highly chaotic demand patterns for suppliers. BWE is distinguished by order oscillations at each SC stage and the acceleration of these oscillations away from the marketplace to higher down the chain. Among the first researchers to identify this effect, then called "Demand Amplification", was Forrester [186]. MIT's Beer Game[187] is a very popular and widely used scenario for bullwhip effect analysis [188]. Lee et al. [79] identified five key reasons for BWE: demand signalling, order batching, fluctuation in prices, lead times and game rationing due to supply shortage.

Blockchain technology is one of the most effective approaches to address the aforementioned issues [189]. Blockchain is a ledger that is distributed and decentralized, consisting

of an increasingly long list of records, called blocks, chained in a sequential sequence and operated by a peer-to-peer network [190]. To validate a new block, each node within a network has a private copy of its ledger and must agree to a protocol. Therefore it is not necessary that the participating nodes know the identity of each other. In essence, the addition of a new block is globally managed by a shared protocol to maintain consistency between different copies. Because of its features, blockchain was originally used for bitcoin [191]. A blockchain can be public or private, a public blockchain is one in which anyone can participate and all transactions are recorded publicly, whereas a private blockchain functions in a closed network. In the context of SCs, many of the intrinsic features of public blockchain systems are irrelevant. SC stakeholders are authenticated and known to each other in the majority of effective SCM applications. Therefore in SCM, a private permissioned blockchain is best for sharing sensitive data [192]. It's worthwhile mentioning that a private blockchain can also be a permissioned network, with different access levels allowing for different levels of participation, such as Corda Network [193].

A few of the recent technical innovations, such as cloud computing [194], Internet of Things (IoT) [195], cyber-physical systems (CPS) [196], artificial intelligence (AI) [197] and blockchain [198] have paved the way for substantial advancements in the SC sector (see [199, 200, 201]). Cloud computing has lately emerged as a strong alternative for information sharing and digitizing real-time data in SC [202]. Implementing a cloud-based platform may assist organizations in better coordinating all SC partners and obtaining more reliable data for demand forecasts [203]. In comparison to a cloud, blockchain clearly has the higher ground [204]. A cloud is a virtual space where a user can access data. While the cloud cannot ensure 100 % integrity and tamper-free data, blockchain provides strong security by employing a range of encryption and hashing algorithms, blockchain does not rely on any central third party and data integrity and tamper-free data are guaranteed [205]. Another factor is cost, cloud computing solutions are less expensive when utilized on an ad-hoc basis, but when used on a regular basis and with predictable workloads, they are more expensive [206]. Blockchain has a number of advantages that make it a better choice for SC businesses [207, 208].

Furthermore, despite the benefits of blockchain, there are still a number of obstacles to overcome when it comes to implementing it in SC. Stakeholders are concerned about sharing demand and inventory data on a blockchain with other businesses [15]. As demand data is confidential, disclosing it could damage a business' financial and competitive position. Businesses are believed to be unwilling to give the data needed to reduce BWE via a blockchain, without first establishing trust. Therefore one way to improve trust and data accuracy is by integrating a trust management system into a supply chain blockchain.

One of the key issues of blockchain distributed computing is the problem of consensus [209]. Each transaction is distributed and verified across a network of nodes running the blockchain protocol. In the literature, the issue of consensus has been widely explored, but its application in the blockchain area has given the motivation to develop innovative ideas for blockchain systems architecture [210]. Owing to its decentralized existence, the proof-of-work (PoW) [211] consensus algorithm is commonly used in blockchain frameworks. Conventional consensus algorithms, such as PoW, takes more time and energy, resulting in excessive latency and low throughput. An alternate consensus approach that is gaining momentum, is called "Proof of Authority" (PoA) [173], where transactions are validated by validators and participants can gain the right to be such validators. As a result, without the need to spend large amounts of computing power and energy, PoA achieves the same mining effect as the usual PoW approach.

The existing PoA algorithm may not be a suitable choice for a larger network because of its static nature. PoA encourages network monopolization and centralization, making it unsuitable for SCM applications. As there is no competition for the few authority nodes who obtain the privilege to validate blocks, this results in a mining monopoly [25]. To prevent this monopolizing a decentralized consensus mechanism, based on a trust score is proposed, which inherits the advantages of PoA in terms of energy and transaction efficiency, but can better select validators. It is challenging to apply since the uncertainty that leads to inaccurate selection is implicit: the algorithm could overlook competent miners without knowledge of the mining skill of unknown candidates. And when facing applicants who were previously chosen due to their unpredictable actions, the algorithm can nevertheless make incorrect choices, such as being inaccessible due to energy shortages or

malicious attacks. To overcome these challenges, a consensus algorithm based on random selection is proposed, along with a scoring mechanism. The selection process is intended to randomly select miners with high trust scores and availability.
Despite the potential, the study of establishing a full information-sharing SC model based on BCT and trust has not been addressed earlier. Therefore understanding how to build trust among partners is critical to ensure that all information is communicated in real-time and thereby reduce demand and inventory amplification. From a strategic perspective, it is evident that sharing full information with more businesses in a more explicit manner may result in greater advantage, as well as a greater risk. This research attempts to investigate certain fundamental concerns in order for businesses involved in an SC to strike the optimal balance between minimizing BWE and inventory costs and sharing information.

4.1.1 Chapter Contributions

To achieve that, this chapter makes the following contributions:

- An information-sharing model is implemented based on a private permissioned blockchain for a complex SC scenario. The implications of full information sharing on minimizing BWE, inventory variance and costs are investigated.
- A trust system is proposed, which takes into consideration the authenticity of data. To eliminate monopolies, authority is kept decentralized while identities remain anonymous.
- The correlations between the proposed blockchain-based framework and its managerial implications are discussed.

4.1.2 Chapter Organization

The rest of this chapter is organized as follows. Section 4.2 introduces an SC data-sharing model and its components. In Section 4.3, the proposed blockchain-based SCM model is

presented with the improved trust consensus protocol. Section 4.4 describes the details of the experimental evaluation of the proposal. Finally, Section 4.5 gives the conclusion.

4.2 Data Sharing in Blockchain-based Supply Chains

A supply chain analysis framework has been created to carry out a "what-if" case assessment to quantitatively analyze the advantages that businesses would achieve in a blockchain-enabled supply chain. This captures the network of the manufacturer, supplier, distributor and retailer, the distribution of products and information between them and all the major processes of an organization, production and distribution.

Let us consider a conventional four-echelon SC consisting of a supplier, a manufacturer, a distributor and a retailer, as shown in Figure 4.1. In each period t, each echelon I accepts orders from its downstream partner $I-1$, fulfils these orders from its own stock and then issues an order O to echelon $I+1$. The observed demand at the end of period t is denoted by d_{t}. The retailer has met the customer's demand $\mathrm{d_t}$ at time period t. The retailer then assesses its inventory and places an order with the distributor to replenish it. The replenishing lead time from the distributor to the retailer is denoted by l4. This order will be delivered to the retailer at the start of the time period $t+1+l_{\mathrm{r}}$. The distributor perceives the retailer's order as a demand, and it keeps an eye on its inventory levels. It will place an order from its upstream echelon if there isn't enough stock. The replenishment lead time from the manufacturer to the distributor is denoted by l3. Thus the distributor will get the order at $t+1+l_{\mathrm{d}}$. When the manufacturer receives the order from the distributor, it examines its inventory level as well. If there is not enough stock, the manufacturer must place an order with their supplier to restock their inventory. Similarly, the lead time from the supplier to the manufacturer is denoted by l2. Hence, the manufacturer will receive the order at time $t+1+l_{\mathrm{m}}$. An echelon's lead time L is defined as the period between placing an order and receiving it, and it is considered to be stochastic, independent and uniformly distributed. As a result, each time an order is generated, a random integer L is assigned to it, which corresponds to the number of periods required for it to arrive. After

L time, the order is accomplished and returned to the previous echelon at the beginning of period $t + L$. When placing an order, it is assumed that no fixed order costs would be incurred and that unit inventory holding costs and shortage costs will continue to remain constant over time.[212].

Let d_t, where t = 1, 2,..., be the demand process at time t.

$$D_t = d + pD_{t-1} + q \tag{4.1}$$

where $d > 0$ is a prior estimation of average demand at period 1; $-1 \leq p \leq 1$ and $-1 \leq q \leq 1$ are constant coefficients representing the correlation between current demand and previous demand and retailer action, respectively; Following is a description of the retailer's net inventory level:

$$I_t = I_{t-1} + q_{t-L} - d_{t-1} \tag{4.2}$$

where I_t is the net inventory at the beginning of time t, which is on-hand inventory, whereas a negative value of net inventory indicates a back-order condition on customer demand.

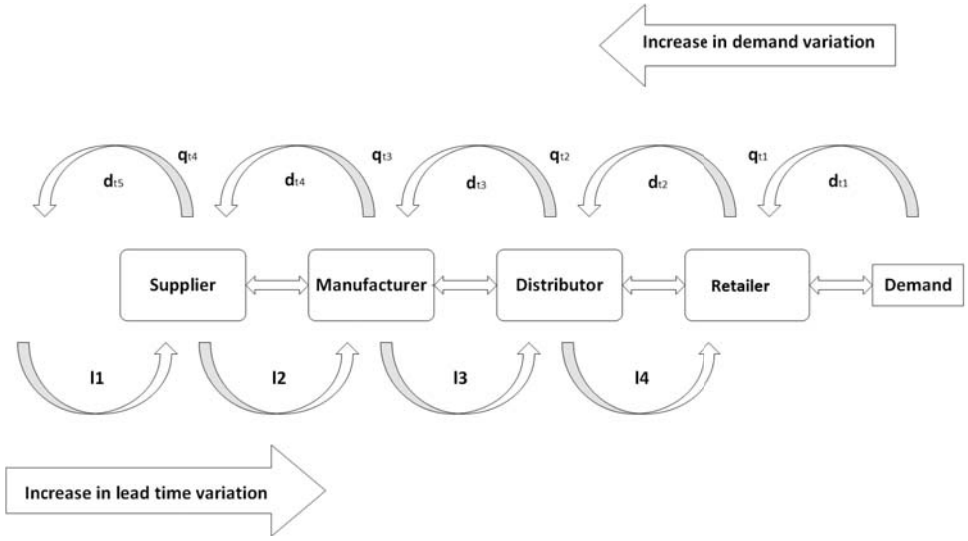


Figure 4.1: A multi-echelon SC with demand and lead time variations.

4.2.1 Assumptions

The following assumptions are provided in the SC model.

- The retailer follows the end customer's demand and uses the (Q, R) inventory policy (explained in the next subsection) to establish orders at the upper tier.
- The manufacturer has an infinite capacity for any amount requested by the retailer to be produced.
- Unfulfilled orders are not lost at any stage due to being out of stock, rather they become backlogs, to be completed as soon as the inventory is restored, as the backlog is more cost-effective than lost sales [213].
- Orders are either positive or zero, cancellations are allowed.
- The actual demand is not known in advance.
- Only a single product is considered.
- Information delay is simulated as a shift in the time dimension of information concerning demand.
- The cost of transportation, production and transportation lead time are not considered.

4.2.2 Inventory Policy

An inventory system is a set of rules that are used to manage inventory levels under control. The two most common replenishment policies are the Continuous Review Policy and the Periodic Review Policy. The Continuous Review Policy is a policy that checks inventory on a regular basis and orders Q quantities when the inventory level reaches the reorder point R [214]. On the other hand, the Periodic Review Policy monitors the physical supply of inventories at regular intervals and places orders for a maximum quantity of

inventory [215]. The periodic review policy involves calculating and recording inventory over a specified period of time, but the continuous review policy necessitates calculating and recording each item from the time it was removed from the inventory.
For a variety of reasons, the Continuous Review Inventory Policy is chosen over the Periodic Review Policy for the proposed framework. A continuous review strategy has the advantage of allowing for real-time inventory counts, making it easier to decide whether or not to restock items. Furthermore it facilitates accurate accounting calculations. A periodic review could save time when reviewing inventory levels, but there could be errors in determining the amount of inventory at times of high sales volume, which has a significant impact on accounting inaccuracies. [216].

4.2.2.1 Continuous Review Inventory Policy

Each echelon is presumed to follow a continuous review (Q, R) inventory policy to handle their inventory. The two parameters to be determined under the policy are replenishment quantity, Q and reorder amount, R. As the on-hand stock level crosses level R, Q is ordered from the next upstream echelon. If there is an adequate inventory on hand for the upper echelon to satisfy the requirement, the order lead time would be limited, as it only requires transport time. Otherwise there would be more delay as the lead time requires the production time plus the transportation time. If the lead time is constant, the reorder point (ROP) is determined as the amount of demand during the lead time plus the safety inventory [217]:

$$ROP = \bar{d} \times Lt + {\sigma^2}_{\text{Demand}} \times \sqrt{Lt} \tag{4.3}$$

where

$\bar{d}$ – average demand

Lt – lead time (assumed to always be the same)

${\sigma^2}_{\text{Demand}}$ – standard deviation of demand

Here it is assumed that demand is a random variable with a normal distribution and that the average and standard deviation are the same every day.
The simplest method to determine safety stock (SS) when dealing with uncertainties and

multiple variables, is to utilize standard deviation to determine supply and demand variances.

$$SS = \sigma_{\mathrm{d}} \times \sqrt{Lt} \times Z \tag{4.4}$$

where ${\sigma^2}_{\mathrm{d}}$ is the standard deviation of demand, $\sqrt{Lt}$ is the deviation of lead time and Z is the desired service level. The service level factor refers to determining the appropriate service level for a given product by balancing inventory costs against stock-out costs [218]. The higher the desired service level, the more safety stock is required. The objective is to maintain a standard service level of 90% to 95% in the suggested framework.

4.2.3 Performance Measures

Three of the most often employed metrics are used to evaluate performance in the literature, due to their practical importance [219]. The goal of this analysis is to examine the effect of the sharing of blockchain-enabled information in a multi-echelon supply chain, in terms of the amplification of demand fluctuations and the subsequent inventory and cost performance across the supply chain. Three efficiency metrics are considered for this purpose: BWE ratio, inventory variance ratio and system cost.

4.2.3.1 Quantifying the BWE

The BWE ratio reflects the expansion of demand variability in the supply chain [220]. The coefficient of variation, variance, or standard deviation will calculate this variability. The order rate variance ratio (OrVr) and stock variance ratio, are the most common measures to calculate BWE. The variance ratio of the order rate is given by the variance of the order size, divided by the demand variance.

$$OrVr = \frac{{\sigma^2}_{\mathrm{Orders}}}{{\sigma^2}_{\mathrm{Demand}}} \tag{4.5}$$

The larger OrVr is, the stronger is BWE. It is also called BWE variance amplification (BwVA) [221]. The purpose of simulating a traditional SC model is to examine the fluctuations in demand variability due to lead time at each tier. The uncertainty in the SC

can be calculated by taking into account the demand placed on the next echelon from the previous echelon and the order. As the demand is variable and varies during each cycle, it is possible to measure the mean and standard deviation. The standard deviation of demand is calculated at each echelon and then this ratio is used to measure BWE between each echelon.

4.2.3.2 Inventory Variance Ratio

The second metric is called the ratio of inventory variation, which was introduced to measure the degree of inventory consistency [222]. This quantifies the net inventory variations ${\sigma^2}_{\mathrm{NI}}$ relative to the fluctuations in the variability of demand ${\sigma^2}_{\mathrm{D}}$. The spike in inventory instability can also be calculated as the supply chain steps up. An increased inventory variation ratio will result in a higher cost of keeping and backlog, reduced quality of operation and increased total cost per cycle of inventory.

$$InVr = \frac{{\sigma^2}_{\mathrm{NI}}}{{\sigma^2}_{\mathrm{D}}} \tag{4.6}$$

4.2.3.3 System Cost

In order to quantify the system's efficiency, system cost also needs to be measured. Motivated from the work of [223], the overall cost over time t is:

$$Cost_{\mathrm{t}} = B.I_{\mathrm{b}}(t) + H_{\mathrm{s}}.I_{\mathrm{s}}(t) + H_{\mathrm{r}}.I_{\mathrm{r}}(t) \tag{4.7}$$

where B is the cost of back-order as demand arises but is not fulfilled, $\mathrm{H_s}$ is the cost of keeping the serviceable inventory, $\mathrm{H_r}$ is the cost of holding the recoverable inventory. $\mathrm{I_b}$, $\mathrm{I_s}$ and $\mathrm{I_r}$ are the back-orders, serviceable inventory and recoverable inventory during time t, respectively. Serviceable inventory includes finished products that are ready for sale, while recoverable inventory includes no longer needed used products returned to the manufacturer, which is considered for re-manufacturing (see e.g., [224, 225] among others). Often items are returned to recoverable inventory after a consumer has used an item and is unsatisfied. Customer demand is met by a serviceable inventory that can be replenished

through manufacturing or re-manufacturing. Transportation costs and lost sales are not considered in the proposed framework.

4.3 Trust in Blockchain-based Supply Chains

The Byzantine fault tolerant (BFT) [226] algorithm plays a crucial role in the blockchain ecosystem and is a powerful candidate in permissioned environments. BFT-like algorithms have been studied extensively with the objective of replacing PoW while maintaining sufficient fault tolerance.

PoA [173] is a new family of BFT algorithms that has recently gained interest because of the efficiency and fault tolerance offered. The core PoA impression is that it waives decentralization and allows more powerful centralized systems. Although this makes PoA an appealing option for broad logistical businesses, it does carry some concern [227]. PoA systems do have high throughput but as elements of immutability come to the forefront, such as censorship and blacklisting, these can become a problem. Another limitation is that PoA validators' identities are transparent to all and this could possibly lead to exploitation by third parties. For example, if an opponent tries to compromise a PoA-based network, they can attempt to influence publicly known validators to behave dishonestly. Although PoA is less energy-intensive, when compared with PoW and will validate blocks in a shorter period, the following limitations make it unsuitable in the corporate sector.

1. Lack of decentralization.
2. Censorship and blacklisting
3. Visible Identities of validators.

4.3.1 Improved PoA Consensus Algorithm

Thus during the consensus process, the biggest drawback of this consensus algorithm is the possibility of centralization and monopoly. However a centralized aspect can be

avoided to restrict the conduct of autocracy in a network and to resolve single-point-of-failure problems [228]. The algorithm to do this is presented in this section, it has been optimized to improve upon the problems above.

4.3.1.1 System Initialization

To join a network, all network nodes should have their identities verified by the CA. Each participating node has its own public and private keys, as well as encryption and decryption certificates. The identities of each node have been kept private to maintain anonymity within the framework.

4.3.1.2 Trust Contact

The major purpose of the suggested algorithm is to evaluate the SC participants' credibility. To give a quantitative metric of trust, each node in the SC network is given a trust score. The trust contract is used to compute the trust of the node for a trade event happening between two nodes that occurs at time t. At the time (t_0), when a node joins the network for the first time, it has no previous trust score ($TNode_0$). Therefore the initial trust score is assigned to the node ($TNode_0 = TrustMin$), which is the least trust score that each node must retain in order to continue participating in the network. The validator's trust is based on their previous encounters with each other. Each transaction steadily builds the node's trust score. The trust values of nodes fluctuate depending on their actions. Their trust values fall if they conduct themselves badly; otherwise, they grow. Stakeholders can provide a rating to a transaction based on their individual experiences. This can be accomplished by expressing satisfaction with a transaction (with 0 denoting dissatisfaction and 1 denoting complete satisfaction). The equation is:

$$T_{\text{Score}} = E_{\text{suc}} \setminus E_{\text{tot}} \tag{4.8}$$

where T_{Score} denotes the score for the transaction experience; and E_{tot} is the total number of interactions; E_{suc} is the number of successful interactions. Trust scores are calculated

on a continuous basis and stored in the profiles of stakeholders. For the framework, Zou et al. [117] the proposed trust management is adopted. To generate the overall trust score $Trust(\Delta_{\mathrm{n}})$ for a stakeholder at time Δ_{t}, the trust scores are applied to the current and past SC events as follows:

$$Trust(\Delta_{\mathrm{t}}) = \sum_{t=0}^{t=i} Trust_{\mathrm{Node}}(t) \tag{4.9}$$

where $Trust_{Node}(t_0)$ is the initial trust score of the stakeholder and $Trust_{Node}(t_i)$ is the current trust score of the stakeholder, happening at $\Delta_t t$.

The consensus mechanism is built on a fair balance of trust and accountability across all network participants. The most significant characteristics of any network are transaction and business activity, which are represented by trust scores.

4.3.1.3 Validator Selection

When it comes to selecting authority nodes, the algorithm selects a fixed number of authority nodes based on their highest trust scores (*TrustMax= 2000*). In order to decrease the predictability of the authority nodes and ensure that the chosen node has more credibility (trust score), randomization in the selection of the authority nodes is proposed. By default, three nodes are picked as authority nodes each round; however the number of authority nodes can be changed as the network grows. Often network congestion and node synchronization issues may cause transactions to be lost or delayed [229]. However PoA takes less time to synchronize since only authority nodes need to validate transactions. Additionally validation is completed in a single round [230]. Therefore in the event of network congestion, the number of authority nodes can be increased to a substantial percentage of validators without jeopardizing transactions. The selection and consensus processes of validators in the enhanced PoA algorithm, are unaffected by network expansion because of the inherited characteristics of PoA. Furthermore, the proposed approach is considered to be a better alternative, since it provides trust, as opposed to PoA which promotes network monopolization and centralization. The workflow for the improved PoA

algorithm is given in Figure 4.2. Whenever an authority node conducts an attack by making fraudulent verification and/or packaging false transactions into a block, their trust scores decline as a result of their negative conduct. This study uses a trust-based selection mechanism, where authority nodes are chosen based on the combination of randomness and trust scores. Therefore there is no way to predict which node will be picked as a validator. The nodes with the highest trust scores have an equal probability of being selected. How to select authority nodes correctly and effectively is key to the algorithm. According to their trust scores, consensus nodes are chosen randomly from many nodes.

4.3.1.4 Consensus Process

All the transactions are obtained and validated by the validator nodes when a block is created and distributed through the network. Finally a block is mined by the authority nodes and synchronizes data to other nodes for validation and clarification. Other nodes can obtain and validate the block by checking the authority node's trust score. Every authority node has the authority to mine 100 blocks (*Authduration = 100 blocks*). After mining 100 blocks, the algorithm replaces the current validator with another validator. Note that the authority nodes do not remain in place indefinitely, their position changes according to their reputation score. This will strengthen the decentralization of the system and eliminate monopolization of validation. If an authority node fabricates a block, it will get a lower trust score and will be downgraded by losing the right to be an authority node for further rounds. Through dynamic selection, the algorithm regulates the network nodes such that the number of nodes involved in the consensus remains constant and each node has an equal opportunity to become an authority node.

4.3.1.5 Trust Score Update

Note from section 4.3.1.2, that in order to participate in the network, each node must maintain a certain degree of trust score. However when a seller's reputation improves or deteriorates, this trust value must be revised. Therefore when a round of consensus is

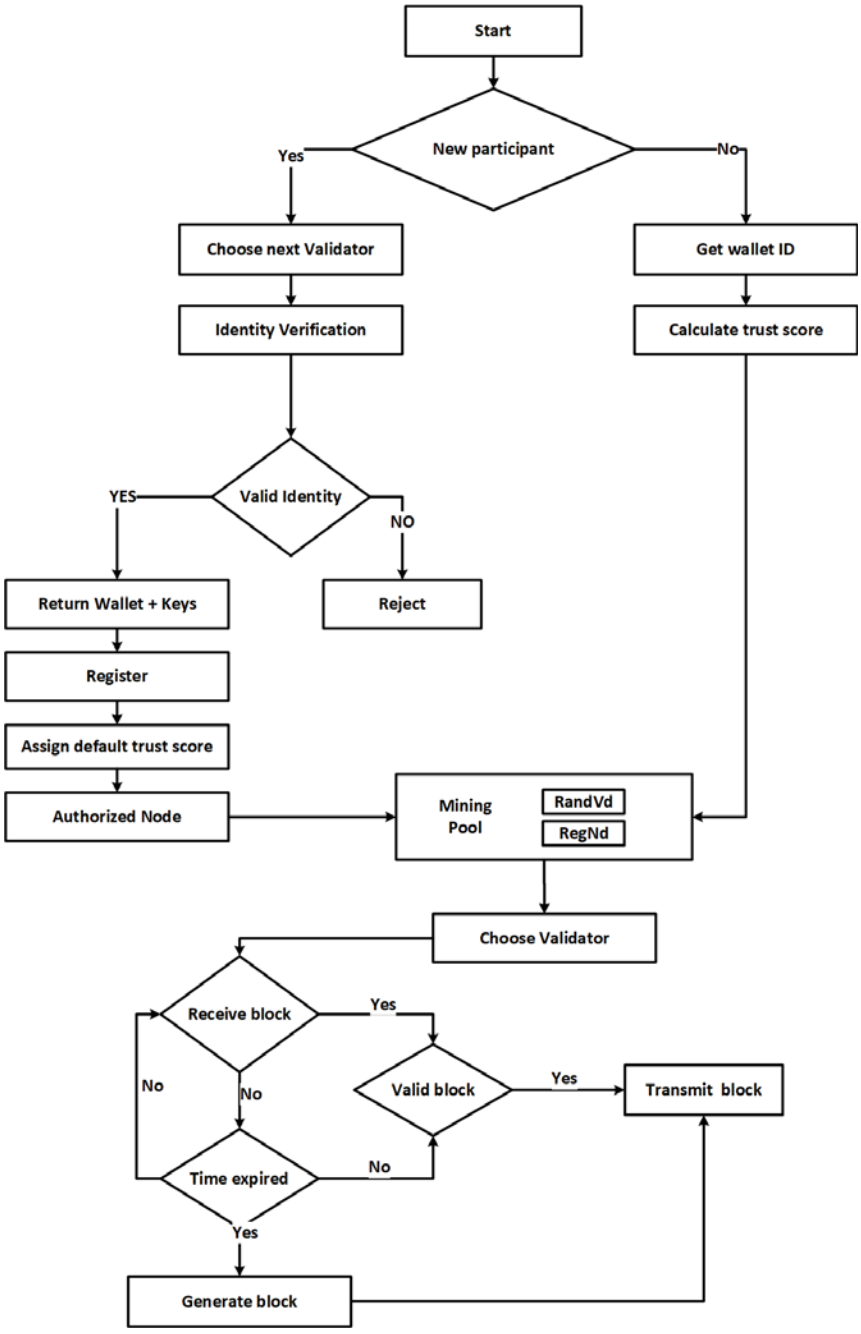


Figure 4.2: Working of enhanced PoA algorithm.

completed, each node's new trust value is updated and recorded in their profile on the blockchain, where it can be viewed and validated by other nodes. A trust score can be

positive or negative, solely based on their transactions and the responses from other nodes. Therefore a node's trust score influences its chances of becoming a validator node during the validator selection phase. The trust score has a range of [0, 1].
In addition to the trust score, various application-specific variables, such as customer experience, could have an influence on a node's trust score. Thus the trust score of a node can be calculated as:

$$Trust_{\text{Node}} = Trust(\Delta_{\text{t}}) + Q_{\text{n}} \tag{4.10}$$

Where $Trust(\Delta_{\text{t}})$ is the overall trust score of a stakeholder and f_{n} is the quality score from a customer. The enhanced algorithm also preserves the benefits of the BFT algorithm, such as low latency and high throughput. Since blockchain is a decentralized environment without overarching control, to derive their current trust scores, the nodes are constantly reassessed. All transactions are given the trust score of the initiator of the transaction (trust score of the sender node).

4.4 Evaluation and Results

In this section, a detailed description of the proposed model is explained, along with the system components and critical considerations.

4.4.1 Experimental Setup

The proposed framework was tested on an Intel Core i7 64-bit, 3.4 GHz machine with an 8 MB cache and 16 GB of RAM running Windows, and simulations were performed using Python 2.7. For analysis, modelling and simulation, a variety of python libraries were utilized, e.g. the NumPy package made the simulation run faster by assisting with various types of scientific calculations. Two scenarios are considered for this research:

- In the first scenario, referred to as Scenario 1 (traditional SC without blockchain), due to low trust, no information about demand data and lead times is exchanged

between tiers.

- In the second scenario, known as Scenario 2 (blockchain-enabled Information Sharing), information about demand data and lead times are exchanged by each stakeholder in real-time and are accessible to all.

4.4.2 Certificate Authority (CA)

The proposed framework assumes that each SC entity is a trusted SC participant who has been issued a public/private key pair by a trustworthy CA. It makes use of existing public key infrastructure (PKI), in which a certificate authority (CA) acts as a trusted third party to issue public-key certificates for the verification and validation of a user's public key [231, 232]. Based on the X.509 standard [233], the PKI is a hierarchical tree structure of CA's with a CA that creates, distributes, verifies and revokes users' public-key certificates. A CA also documents the relationship between stakeholders and their individual identities. If necessary, a CA can disclose the real identities of the stakeholders.

4.4.3 Proof of Concept Implementation

The following steps and Figure 4.3 define the blockchain-coordinated SCM framework:

1. Initially, using the original identity, each stakeholder who wants to access the network is registered with a CA. A certificate for each stakeholder is created by the CA after verification. As they are registered in the blockchain, all operations undertaken by a CA are transparent.

2. The stakeholders entering the network for the first time (without a prior reputation score) will be given a minimum trust score.

3. The stock inventory is initialized when an order arrives at the retailer. The demand quantity is recorded in the blockchain, to evaluate the demand deviation. Next, depending on the existing volume of inventory, the retailer evaluates if the demand

quantity can be fulfilled. The demand quantity is then excluded from the inventory if the inventory is greater than or equal to the demand quantity. Alternatively, the demand quantity becomes backlogged and the order will be sent to the upper echelon.

4. An order is put at the next upper echelon when the relative inventory volume is lower than or equal to ROP. The information sent to the next upper echelon includes order ID and order quantity. Information on the lead time of this order is sent back from the next upper echelon and information regarding the order, including order ID, date of release, lead time and order quantity, is also recorded on the blockchain.

5. When the delivery date arrives, an order is shipped. Information about order ID and actual delivery date will also be submitted to the upper echelon. The current level of stock is changed and the order is discarded from the order receipt list.

6. As the lower echelon gets the order distribution details from the next upper tier, the replenishment quantity is added to the existing inventory. Afterwards the estimated lead time is revised in case it is different from the order lead time.

7. In order to calculate the variance of the order placed and the demand received to measure the BWE ratio, the received order quantity is also recorded on the blockchain and calculated according to Equation 4.6.

8. Inventory review is conducted to check whether the relative level of inventory is less than or equal to ROP. If relevant, the order quantity is put on the next upper echelon.

9. The retailer receives its order and gives a rating score to the supplier based on the service level. Furthermore, the trust score is stored in the blockchain with the associated profile.

10. Steps 1-9 are repeated for other stakeholders when they get a demanded quantity from their next lower echelon.

11. Based on the corresponding trust scores over time t; a stakeholder is awarded authority.

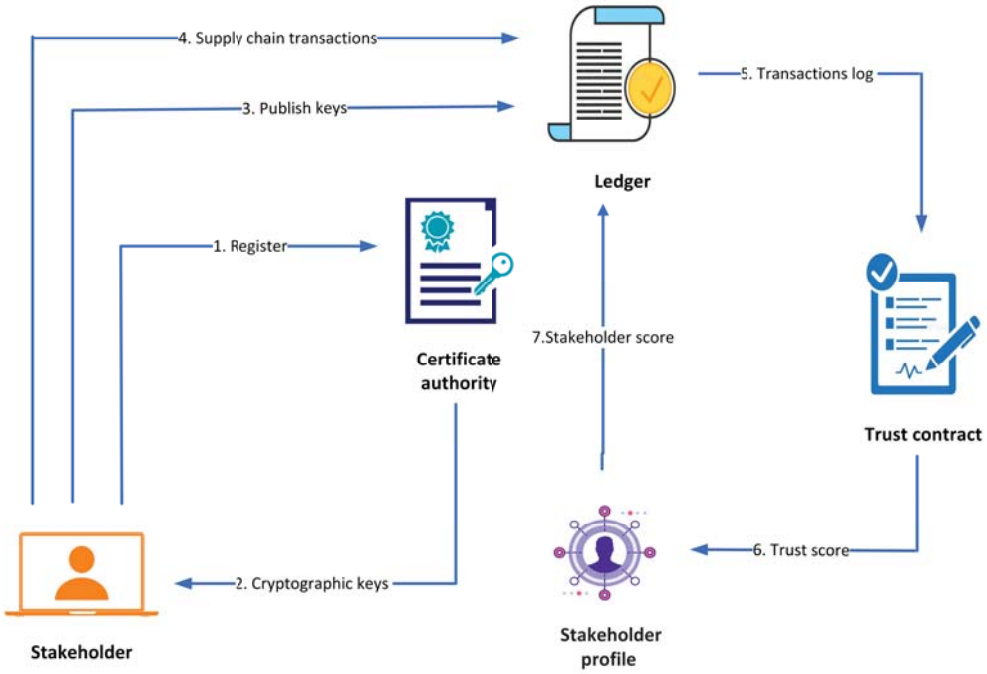


Figure 4.3: Blockchain coordinated SCM framework.

4.4.4 Security and Privacy Analysis

The proposed private permissioned blockchain for fair data sharing meets all the key security standards, including single registration, encryption and stakeholder anonymity. The model is also predicated on two main assumptions: a completely trusted permissioned network and access control. The suggested architecture employs cutting-edge secure encryption and intrusion detection technologies. In this section, the proposed architecture's security was evaluated in order to demonstrate that it satisfies security standards and is both attack-resistant and compliant.

- Whitewashing attack: A whitewashing attack is performed by a stakeholder when it

rejoins a network with a new identity and thereby acquires a new trust score. If the adversary's registration is revoked and it attempts to register again, it would fail in the proposed scheme. Since all stakeholders are registered in the approved network by a CA, who are all called reputable entities, the likelihood of a whitewashing attack is considered negligible.

- Privacy Preserving: Every stakeholder gets its unique identity from a CA after a successful registration. The events are initiated and checked by the stakeholders using their identities. CA keeps track of the identities (pseudonyms) of the participants and their original identities. Using the ECDSA signature technique [234], stakeholders' anonymity is maintained. Because of the pseudonyms, an adversary would be unable to link the data to the original identities of the stakeholders and so this is how it prevents bad-mouthing attacks. Since the identities of the participating nodes are not known to each other, identities are protected. Thus a node cannot submit negative feedback to an honest node in order to damage its reputation.

4.4.4.1 Cost Analysis

Cost is the relative metric of BWE and for businesses, this measure is more important since the goal of each business is to increase their benefits, rather than reduce BWE. The cost consequences of sharing were analyzed in both scenarios. The overall costs of the two scenarios are listed in Table 4.1. The cost is given in USD. It is obvious from the table that with blockchain-enabled information sharing, SC costs are significantly reduced.

Table 4.1 illustrates a clearer picture of total inventory cost savings in the SC after the introduction of blockchain. For both of the scenarios examined, no costs were associated with transportation or lost sales. The simulation reveals that blockchain-based data sharing delivers very good efficiency with a cumulative cost reduction of 75%. The total cost consists of an inventory-keeping cost of 97.3% and a back-order cost of 2.7%. The average performance across tiers is close, but performance enhancement is more noticeable as the

Table 4.1: Comparison of the overall system cost.

Cumulative cost in $		
	Scenario 1	Scenario 2
Retailer	2500	1160
Distributor	6066	1324
Manufacturer	6779	1308
Supplier	5075	1275
Total	20,420	5,067

upstream is spread over longer lead times than downstream. It is also important to note that the costs involved with BWE have a larger influence on upstream SC parties (e.g., manufacturers), while the costs associated with the accumulation of total stock would mostly affect the downstream tiers (e.g., retailers). It's obvious from the table statics that with complete visibility, the upstream tiers had significant cost reduction.

By providing stakeholders with access to demand data throughout the SC, it is straightforward for them to minimize costs, reduce BWE and improve inventory planning. Blockchain negates the need for an intermediary, allowing distributed data to be shared directly with entities, reducing information asymmetry, lowering costs and increasing trust, resulting in high data integrity and robustness.

4.4.5 Performance Evaluation

4.4.5.1 Scenario 1 Analysis

In scenario 1, a four-echelon linear SC is simulated: retailer, distributor, manufacturer and supplier. For each echelon, the assumption here is that the replenishment lead period is 1 week. The run length of the simulation model was set for 40 weeks. The inventory holding cost is 0.50 per week per unit of inventory, the backlog cost is 1 per week per unit of inventory and the starting inventory equals 15 units for each node. After obtaining raw data from simulations, statistical analysis was used to measure the proposed

research objectives. Simulation data can be accessed at `https://research.unsw.edu.au/projects/cross-disciplinary-optimisation-under-capability-context`.

4.4.5.2 Scenario 2 Analysis

In scenario 2, the behavior of another blockchain-enabled SC model has been studied. This experiment considers a dynamic SC with several uncertain parameters, the same as in scenario 1. Blockchain, as an encrypted digital ledger, can serve as a centralized database and provide real-time access to all records, as well as transparent and synchronized updates. All entities produce their predictions from the same information and demand data, thus further minimizing variations in production down the SC. With blockchain, entities have increased visibility across the SC, i.e., when data is added by a downstream entity, the upstream entities will immediately have the same degree of access to it, regardless of the number of other entities in between them.

The validation of the above statement was carried out by comparing the results of the simulation obtained in scenario 2 with the results of the amplification of BWE variance and order's standard deviation obtained from the base model. In the simulations with the blockchain-enabled SC, information lead time is not eliminated, but there can be a physical lead time. To address this problem, the same tier has multiple stakeholders, allowing the retailer to immediately place a new order with a different supplier in the event that an existing order is cancelled.

From Figure 4.4, it is obvious that in scenarios 1 and 2, BWE continues (i.e., BWE >1) and cannot be absolutely prevented. However with the blockchain-enabled SC (i.e., scenario 2), BWE is substantially diminished. Scenario 1 reveals the largest BWE influence with a geometrically increasing ratio as it travels from retailer to supplier. Scenario 2 reflects that the demand variance has no effect and the effect only comes from physical lead time (as we have no information on lead time with blockchain). A substantial decline in the impact of bullwhip occurs with shared demand information for the supply chain (bullwhip decreases by 99% for retailers 'R3, R2, R1', 98% for manufacturers 'M2, M1' and 97% for suppliers 'S3, S2, S1'). This is attributed to the reduced loss of supply provided by

the 'M' manufacturer. Therefore incorporating blockchain in an SC is an efficient way to manage BWE and enhance the efficiency of SCM.

After the implementation of the blockchain-enabled SC, the upstream SC members can

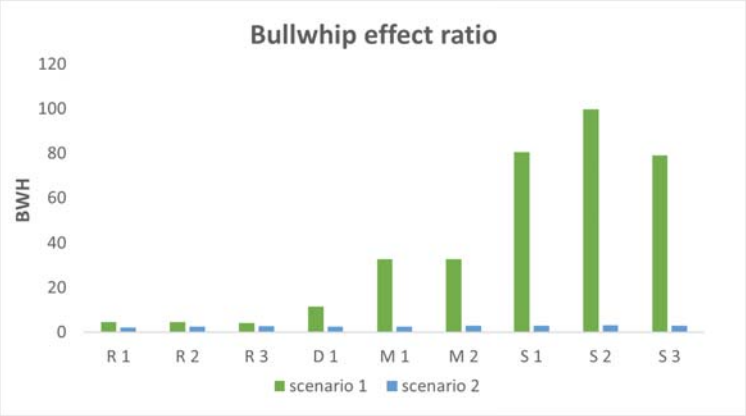


Figure 4.4: BWE ratio comparison.

immediately consider the demand data of the downstream members and can quickly respond to the orders of downstream members, the demand of downstream companies is also rapidly known. There is only a very low risk of unknown variations, so there is no need to retain a high degree of inventory in the SC to avoid out-of-stock situations. From Figure 4.5, performance in terms of the inventory variance ratio (InVr) largely parallels BWE in Figure 4.4. The lack of information sharing contributes to a significant rise in inventory variance going upstream in the SC. Blockchain collaboration has a local and global effect: the inventory variation ratio falls from 20.38 to 12.19 at the distributor (local effect) and from 7.38 to 5.19 at the supplier (global effect).

In this set of experiments, a Continuous Review Inventory Policy [235] is utilized and it is obvious from the graph that the inventory level dropped 75% for the suppliers. As the retailers always face the demands of end consumers and address them in a timely manner, the retailer should coordinate the actual demand information so that the upstream companies in the SC can have real-time downstream market information. So the inventory strategy is implemented without needing to hold unnecessary inventory to satisfy the demands of downstream companies.

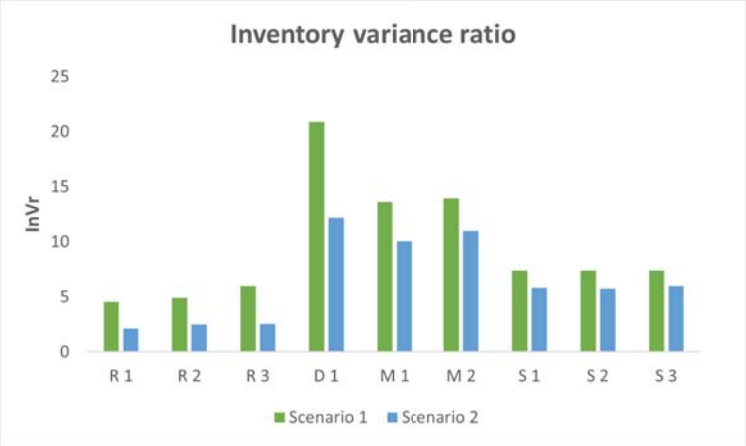


Figure 4.5: Average values of inventory variance ratio for scenario 1 and 2.

4.4.5.3 Impact of Order Lead Time

In this analysis, many stochastic variables in the dynamic SC, including consumer demand and lead times, act as random sources. Thus it is crucial to identify the behavior of the system under various settings of stochastic factors and their consequences. Previous research has looked into the effect of lead time on BWE; however, there is no blockchain-based solution in the existing literature [236, 237]. Therefore the goal of the blockchain-enabled data-sharing architecture is to explore the effect of lead time on demand variance. In addition, the trust consensus was improved and experiments with and without data sharing were conducted.

The results can be seen in Figure 4.6. The graph clearly shows that demand increases as it moves from consumer to supplier, which reflects a higher BWE. Variation is very small at the retailer, as its nearest to the demand. So it should be easier to observe and estimate demand at this level. However the instructions sent to the distributor influence the order variance of the distributor. Owing to the rise in demand variation, the upper echelon would have higher demand variance. The combined effect of the lead time variation, on the other hand, tends to strongly influence the upper echelons, especially the manufacturer. Due to the lead times of the corresponding stages, a disparity in the peak is also induced. It can be seen that the magnitude of BWE is significant relative to lead time e.g., for the

distribution when demand variance is 250 then BWE is 4.59 and without variance is 3.59. In other words, when the lead time is greater, BWE is more apparent.

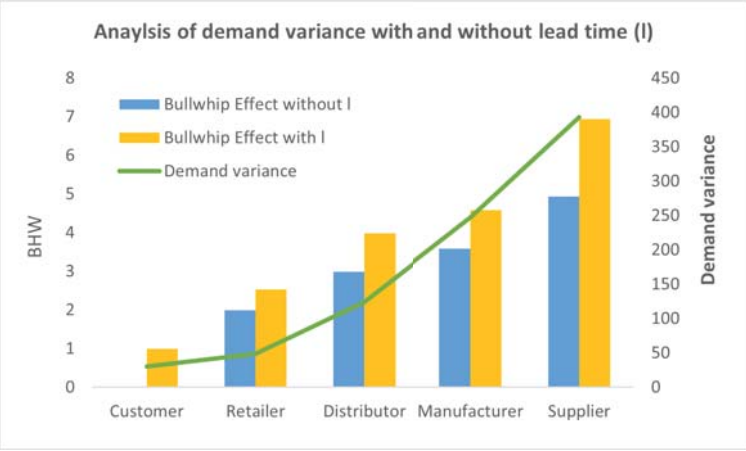


Figure 4.6: Demand variance, with and without lead time.

4.4.5.4 Impact of Order Cancellations

Uncertainty in development environments plays a significant role in evaluating expected customer order lead times. Many of these uncertainties are due to complex development processes and have large effects on efficiency. A significant step in increasing the robustness of production preparation is the integration of these considerations into the collection of expected lead times. Many areas of costs and management are influenced by lead time instability. When a retailer cancels an order because the upper echelon is unable to meet the order on schedule, then lead time increases and the holding cost of the upper echelon increases and BWE worsens. On the contrary, when a customer wishes to cancel their order because of stock out, they have probably found an alternative source for their product. Many businesses would ensure that their main products have more than one point of supply; thus buying from an alternative could be better than waiting for an order to be completed.

Tan [81] believed that bullwhip's impact is caused by the cancellation of customer orders.

They find that cancellations are a significant source of distortion of inventory information that raises overall system costs. A cancelled order can be expensive for the manufacturer, not only in missed sales but also in the procurement of raw materials or components carried out for a cancelled order of a client. Outdated, slow-moving, or unsalable products cost money, not just due to their purchase price, but also in inventory carrying costs. In Figure 4.7 it can be observed that when an order is cancelled due to increased physical lead time [238], that the system cost increases proportionally. In this set of experiments, manufacturers must deliver under the time they pledge in their contract. So if a retailer has been waiting for an order to come for a long time and it still has not been delivered after the stated time period, the retailer can cancel the order and simultaneously place the order with another manufacturer who is able to deliver their order on time.

With blockchain information sharing, the retailer can identify how many units of item inventory the manufacturers have and the volume they will need to generate for the manufacture of their goods at the time of order placement. As a result, the retailer's lead time is less than it would be if there was no sharing of information. The system's cost is further reduced by allowing retailers to make their own decisions. The experiments show that blockchain can help solve this challenge by allowing multiple stakeholders to share inventory and availability for upcoming orders, saving time and improving system performance.

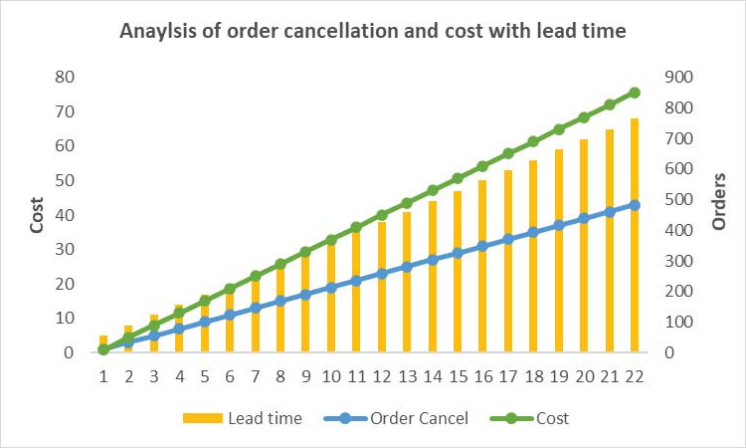


Figure 4.7: Analysis of order cancellation in terms of increasing cost with lead time.

4.4.6 Supply Chain Costs

By providing stakeholders with access to demand data throughout their SC, it is straightforward for them to minimize costs, reduce BWE and improve inventory planning. A noticeable distinction between the two scenarios can be observed in Figure 4.8, where inventory holding and shortage costs appear to be linearly linked to the total stock variance. Also BWE, total stock amplification and variability costs (that arise from unpredictable scheduling) are linearly linked to order variance. From the analysis, the lowest total costs are in scenario 2, a result that was anticipated considering the efficiencies presented in the first framework. The different tiers have lower costs of inventory, cost back order and total cost as compared to the traditional SC. In scenario 1, shortages and backlog order costs for the supplier are quite high, while in scenario 2, they are zero, which means that all orders have been met. The shortages and back orders are minimized from high to zero, ensuring that the supplier never goes out of stock and can meet all the orders issued by the retailers, as the latter provides accurate information about lead time and demand via the blockchain. This demonstrates that using the proposed framework stimulates cost savings, while also increasing profits for supply chain participants. The blockchain supply chain's holding costs have been lowered. This is due to the effects of data sharing in real-time. While scenario 2's inventory level is high to mitigate shortages at the retailer and distributor stages, and this raises the overall inventory level and thus the holding cost, this is still lower than for scenario 1. The results demonstrate that blockchain technology is a cost-effective method to solve problems of coordination and trust in an SC and to minimize the negative impact of data asymmetry on an SC's levels.

4.4.7 Comparison of PoA and Improved PoA Algorithm

The improved PoA incorporates a trust-based selection mechanism, in which authority nodes are picked using a combination of trust scores and randomization, based on their transactions and the responses from other nodes. On the other hand, PoA will consider predetermined validators for mining, without including their current behavior in the eval-

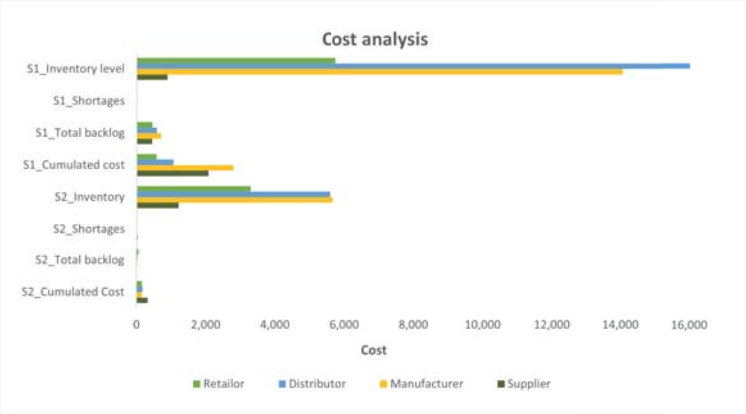


Figure 4.8: Cost analysis of both scenarios.

uation. The purpose of this experiment is to demonstrate how trust evolves when adopting the improved PoA algorithm, versus the PoA algorithm without a trust score.
In this experiment, nodes act honestly in a block cycle in order to raise their trust scores. After a round of consensus, such nodes' new trust scores are updated and recorded in their profile. Figure 4.9 shows that the improved PoA detects the validators' changing be-

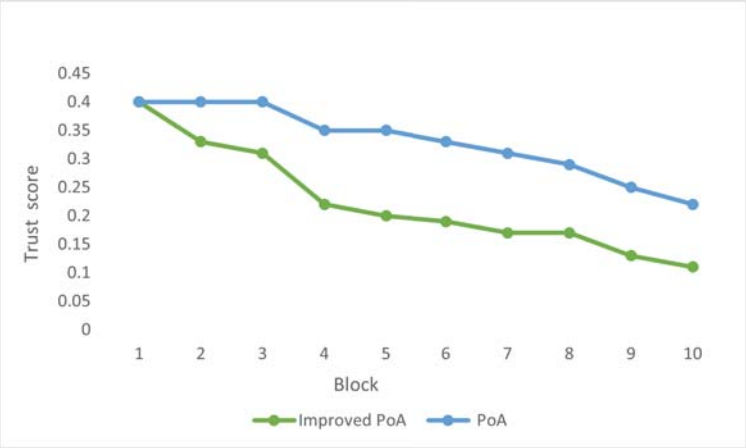


Figure 4.9: Trust score evolution.

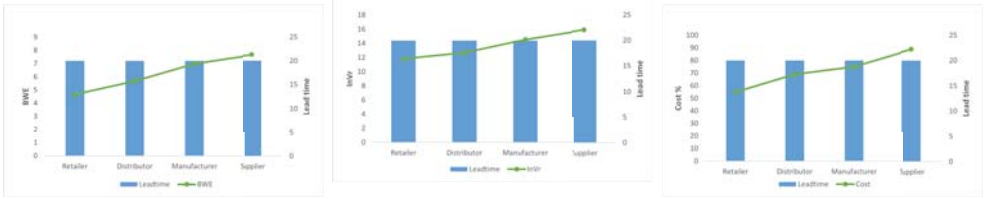
havior more quicker. Bad behavior is discovered through dynamic scoring, and malicious peers are demotivated to continue their behaviors because it results in a lower trust score (downgrade) and less acceptability in the network. Furthermore the malicious node is demotivated to behave passively in the network because it has no possibility of improving

its trust score. The improved PoA algorithm urges the nodes to stay active in all rounds. On the other hand, since the reputation rankings are known in the PoA algorithm and no rating is used, validators can act maliciously and may overlook honest transactions. Therefore the improved PoA algorithm urges the nodes to stay active in all rounds and instills trust throughout the SC, assisting the use of accessible data and data sharing.

4.4.8 Sensitivity Analysis

This section examined the model's performance under various parameters and discuss their impact by relaxing a few assumptions. In this analysis, stochastic variables in the complex supply chain's structure, including consumer demand and lead time functions, are considered as random sources. To keep the computational effort reasonable, the impact of lead-time and demand variance on BWE ratio, inventory variance ratio and cost were concentrated. Changes to the demand parameter's value are made while using a lead time range of 10 to 20 days and considering the resulting costs. Such varied lead times increase the uncertainty in an inventory environment between the demand obtained and orders placed. Analysis of the order fluctuations reveals that due to the stochastic lead period, even a slight deviation of mean demand will induce an increase in the variability of the imposed orders. This represents that order heterogeneity greatly relies on lead time. If lead time is stochastic, it is necessary to set the reorder point higher, which then involves order quantity adjustment at each echelon. Therefore it is anticipated that the longer the lead time, the stronger the effect on the inventory variance ratio and the order rate will become more oscillatory under the same demand input.
Figure 4.10 clearly indicates that lead time has a major impact on inventory variance. Since higher BWE raises inventory, this is not surprising, but also contributes to higher costs. Subsequently, lead time often plays an important role and leads to a large rise in overall costs.

Figure 4.11 demonstrates BWE, InVr and cost behavior under different values of demand variation with lead time in order to determine the impact of demand variance. BWE tends



(a) BWE. (b) Inventory variance (InVr). (c) Cost.

Figure 4.10: The impact of the lead time parameter.

(a) BWE. (b) Inventory variance (InVr). (c) Cost.

Figure 4.11: The impact of the consumer demand variance.

to increase slightly with the growth in demand variance, however, it has a huge impact on cost. It does affect the InVr of all stakeholders, however, upstream participants face larger effects. The research findings suggest that the rise in lead time leads to volatility in inventory and thus will increase system cost. When businesses exchange their inventories on a blockchain, this enables each business to make their own choices, based on shared, comprehensive data and this would improve inventory visibility across businesses and make lead times more predictable. Blockchain can significantly decrease lead times and increase the profitability of many businesses by speeding up transactions, quickly locating logistical issues and providing comprehensive data. Therefore blockchain can be a suitable candidate for eliminating information lead time (which eventually reduces the physical lead time to an extant) along with BWE.

BCT's stability lies in the distribution structure of its shared records. The "mining" process is established, such that an adversarial node must recalculate the cryptographic hash of the whole blockchain if it tries to tamper with its data. Therefore if the entire SCM uses a blockchain network, it ensures its integrity.

4.5 Chapter Summary

This chapter focuses on merging data visibility and trust while exploring the adoption of blockchain technology to reduce BWE in SCM. First, BWE is observed in SC and then a blockchain architecture design was used to minimize it. Full sharing of demand data has been shown to help improve the robustness of the overall performance in a multi-echelon SC environment, especially for BWE mitigation and cumulative cost reduction. It is observed that when it comes to providing access to data, information sharing using a blockchain has some obvious benefits in an SC. Furthermore, when data sharing is distributed, parties in the SC will have fair access to other parties' data, even though they are farther downstream. Sharing customer demand is important in an SC to enhance decision-making, reduce costs and promote the final end product. This work also explores the ability of blockchain technology as a solution in a distributed ledger approach to creating a trust-enhanced environment where trust is established so that stakeholders can share their information effectively.

The findings have managerial implications, as they can allow businesses to exchange real-time data through BCT. Firstly, the research aids in identifying the key elements for boosting BWE in supply chains, as well as how a blockchain-coordinated SC helps for BWE minimization. With the use of BCT, the cost-efficiency of a supply chain would improve in many aspects, such as reducing the need for third-party mediators, reducing processing costs and eliminating system failures. The proposed model further improves the system's overall trustworthiness, such as data authenticity, by keeping authority distributed while identities remain anonymous. Regardless of the consistency of existing coordination and communication in an SCM, managers should look more closely at BCT. The findings suggest that information lead time can be eliminated within a blockchain and thereby also decrease physical lead times.

The proposed framework provides insights into the significance of SC information collaboration in reducing the bullwhip effect. In the event of a disruption in an SC, SC coordination helps to minimize the bullwhip effect. The inventory level in the proposed

framework stays below ROP throughout the period, avoiding queues and lowering inventory costs. Furthermore the trust component gives stakeholders the confidence that they can safely share their data. Our framework does have some restrictions, though, in that we only take consumer demand and order data into account for BWE mitigation. However, when information like inventory levels and work-in-progress levels are also accessible, and all stakeholders can base forecasts on the same data, thus offering better visibility and reducing BWE.

Chapter 5

Scalable blockchain for SCM

In the previous chapter, a blockchain-coordinated framework was developed to address the data sharing, trust and privacy challenges of blockchain. The conducted research in the preceding chapter demonstrated the effectiveness of blockchain-based information sharing in supply chains, as well as how trust between partners tends to boost overall SC efficiency and lessen the bullwhip impact. Note that the consensus algorithm, which can enhance system efficiency, security and trust, is a crucial component of blockchain technology. Since the existing consensus algorithms lack the scalability and computational efficiency necessary for SCM, this chapter focuses on developing those deficiencies . In this chapter, a Scalable Blockchain consensus algorithm is proposed to address the limitations in the existing blockchain solutions, as proposed in Sections 1.2.1 and 2.2.2. The growing interest in blockchain technology has gained a lot of attention in supply chain management (SCM) and sparked the quest for decentralized, scalable, efficient and trustworthy consensus schemes. Traditional blockchains rely on computationally expensive consensus mechanisms with low throughput and high latency. This chapter conducts a performance evaluation of several existing consensus protocols to illustrate blockchain's shortcomings in terms of consensus and proposes a new consensus algorithm: Reputation-based proof-of cooperation (RPoC). The RPoC algorithm uses a layered architecture to segment the nodes that participate in the consensus phase, in order to improve scalability and efficiency

while maintaining trust among peers. The layered design addresses the issues of flexibility and scalability, as well as breaking down the extensive mining process into segments. Rather than choosing a few nodes for mining, the proposed consensus process involves all network nodes, thus making it more efficient, decentralized and scalable. Through extensive theoretical analysis and experimentation, the suitability of the proposed algorithm is shown to be well grounded in terms of scalability and efficiency.

5.1 Introduction

Blockchain has gained a lot of attention in the Supply Chain Management (SCM) domain recently, mostly due to the emergence of digitization and the growth of the industry 4.0 context across industries. The emergence of Bitcoin [190] has further fueled this recognition. Over time, blockchain technology has evolved to meet a variety of applications, resulting in three types of blockchains.

- Public blockchains: Anyone can join and participate in the blockchain network. Examples include Bitcoin [190] and Ethereum.
- Private blockchains: Only selected transactions from authorized participants are allowed on a private blockchain, and the administrator has the authority to overrule, alter or delete any entries.
- Consortium blockchains: Instead of being governed by a single organization, the platform is governed by several organizations. An example is Hyperledger Fabric [54].

Although cryptocurrencies have been the most well-known use of blockchain technology, several researchers have also identified the usage of blockchain and cryptocurrencies in different supply chain applications [184] [154] [239]. Private blockchains are ideal for supply chains [240] due to the nature of how private blockchains work. The proposed framework is based on a private blockchain solution. However integrating blockchain

technology into traditional SCM is a significant challenge, particularly with the absence of tailored consensus algorithms to tackle or embed within supply chain problems [241]. The blockchain architecture validates information through a consensus mechanism among network nodes, removing the need for intermediaries. The consensus mechanism ensures a tamper-proof environment and ensures that the information stored is reliable and valid [209]. In a blockchain, all nodes must agree on the current state of the ledger, making it difficult for adversaries to insert tampered blocks. Many challenges continue to affect blockchain technology, including insufficient transactions per second (TPS), transaction latency and decentralization [106]. The throughput of existing blockchains is relatively low due to the complex consensus process; for instance, in a public blockchain with the proof-of-work (PoW) consensus algorithm, all nodes must perform hash calculations and are only allowed to broadcast their blocks after spending a great deal of energy for their computation [100]. Consequently high consensus latency, low throughput and high energy consumption makes it difficult to use existing algorithms in complex or large supply chain systems.

Considering all those shortcomings in existing consensus algorithms, a proper and customized consensus algorithm should be designed for typical SCM problems, particularly to resolve the TPS, latency and centralization issues. Most current consensus algorithm research focuses on improving mainstream consensus algorithms, even though only a few are relevant to SCM, which are highlighted in the literature review section. While the dynamic SC sector has enormous development potential, it is challenged by several other SCM issues. The Bullwhip Effect (BWE) [242] is one of them and has been discussed in the literature in recent years. BWE occurs by order oscillations at each SC stage. Blockchain can mitigate BWE by providing real-time information and coordination among stakeholders. Sharing appropriate demand data throughout an SC is crucial because it may help the upstream echelons with resource and material scheduling. Furthermore, inventory requirements might be directly linked to inconsistencies between demand over time and actual demand fulfillment. This research utilizes BC to offset the conventional SCM phenomenon of BWE by providing total visibility and exchanging demand data across all stakeholders. This keeps business transactions tamper-proof and available to stakeholders, without the

need for a centralized control body, as long as business practices and negotiated data processing contracts between firms are followed.

To address the aforementioned challenges, first, a few proof-based (PoW, DPOS) and voting-based (PoI, PoC, Ripple, BPFT) algorithms are selected to examine how well they perform in the proposed blockchain-based SCM architecture. Based on the performance of those existing consensus algorithms, the second layer of this work proposes a reputation-based consensus mechanism by redesigning some existing approaches while complementing their strengths and eliminating some of their weaknesses. The proposed consensus algorithm is known as: reputation-based Proof-of-Coordination (RPoC). It reaches consensus by coordinating between two layers of nodes. The first layer consists of high authority nodes that are chosen based on a combination of their reputation score and verified identity. In contrast, the second layer comprises subordinate nodes selected using a random selection algorithm and grouped in clusters with a master node for each. By the performance evaluation (see section 5.3), each of the six existing consensus algorithms decreases blockchain efficiency, by limiting blockchain throughput and increasing transaction latency. Whereas RPoC is made up of layers, each with its own set of nodes operating in parallel, thus increasing efficiency, decreasing latency and eliminating the centralization issue.

5.1.1 Problem Motivation

While having numerous benefits, traditional blockchains are not immediately relevant in SCM. This is because they operate in a dynamic and unpredictable environment that creates millions of transactions per second [243], whereas traditional blockchains have low throughput.

In a blockchain, all nodes must agree on the current state of the ledger, making it difficult for adversaries to insert tampered blocks. The consensus algorithm is the most significant component of a blockchain system as its efficiency significantly influences each blockchain's overall efficiency [209]. Based on the diverse deployment types of blockchains, existing blockchain consensus algorithms may be divided into two categories: Proof-of-X (PoX)

and Byzantine Fault Tolerant (BFT) consensus algorithms. PoX consensus algorithms, such as PoW and PoS, are appropriate for public blockchains with low efficiency and high processing power requirements. BFT consensus algorithms [114, 244] necessitate significant communication resources and therefore have limited scalability. A further significant disadvantage is that the PBFT consensus algorithm's performance decreases drastically as the number of nodes in a network grows [245]. Furthermore the entire consensus process is disrupted if the principal node fails. To overcome these challenges, additional in-depth research is required. On the other hand, private blockchains are highly centralized and have fast processing speeds, making them ideal for adoption in SCM. Nevertheless the consensus algorithms suggested in the literature are mostly intended for public crypto blockchains and so cannot be deployed for private networks, particularly SCM. In SCM, businesses can construct permissioned chains among themselves, and depending on their degree of decentralization and context, they often prefer to compromise the degree of decentralization and use algorithms with higher operating speeds and scalability.

Honey- BadgerBFT [112] has a greater cryptographic overhead than PBFT. Ripple [116] requires more than 80% of nodes for transaction verification, resulting in low throughput and high latency. With the growing adoption of blockchain in the SCM domain, a number of consensus algorithms have been developed to solve these issues. For example, [118], however its shortcoming is that it compromises system decentralization by treating nodes differently depending on their trust scores.

To solve these problems to boost blockchain adoption in SCM, a new scalable, decentralized consensus method, known as RPoC, is developed for permissioned blockchains that meets both performance and security criteria. For it a two-layer consensus protocol is established using a sharding technique and nodes are assigned to different consensus layers. Expanding the consensus groups allows both TPS and scalability to be linearly boosted, while keeping the system decentralized.

5.1.2 Chapter Contributions

This chapter provides a blockchain-enabled SCM framework to provide visibility and coordination along with blockchain consensus processes. A two-layer consensus algorithm that combines reputation and a random selection algorithm is proposed. The following are the chapter's key contributions:

- An information-sharing framework is implemented based on a permissioned blockchain for a complex SC scenario. The use of BC technology in SCM is considered in terms of mitigating BWE.
- A reputation-based consensus algorithm has been proposed by combining the advantages of existing algorithms and throughput, scalability and latency were verified and validated for the improved algorithm.
- The proposed algorithm is compared to the existing consensus algorithms and significantly improves TPS and scalability for SCM applications.

5.1.3 Chapter Organization

The rest of this chapter is laid out as follows. Section 5.2 introduces the RPoC framework and the computational results and discussion along with different performance comparisons are presented in 5.3. Finally, conclusions are drawn in section 5.4.

5.2 RPoC Framework

This section presents how to employ the proposed Reputation-based Proof-of-coordination approach to build a supply chain architecture that minimizes the bullwhip effect. In addition, the selection of consensus nodes and the block confirmation mechanism in the proposed RPoC are detailed.

5.2.1 Blockchain Enabled SCM Framework

This section describes two models: network and threat models. The first is a blockchain-based information-sharing framework for a complex Supply Chain (SC) scenario that minimizes BWE and the second is a threat model that includes assumptions about the number and behavior of adversaries. The proposed system architecture is depicted in Figure 5.1, where any stakeholder who wishes to join the network must first register with a Certificate Authority (CA) while using their original identity. After verification, the CA generates a certificate for each stakeholder. Since a CA is registered in BC, all of its operations are open to the public. A minimal trust score will be assigned to stakeholders joining the network for the first time (without a prior reputation score). Upon receiving their certificates, each stakeholder can start conducting transactions on BC. The detailed BC-coordinated SCM framework can be found in [156].

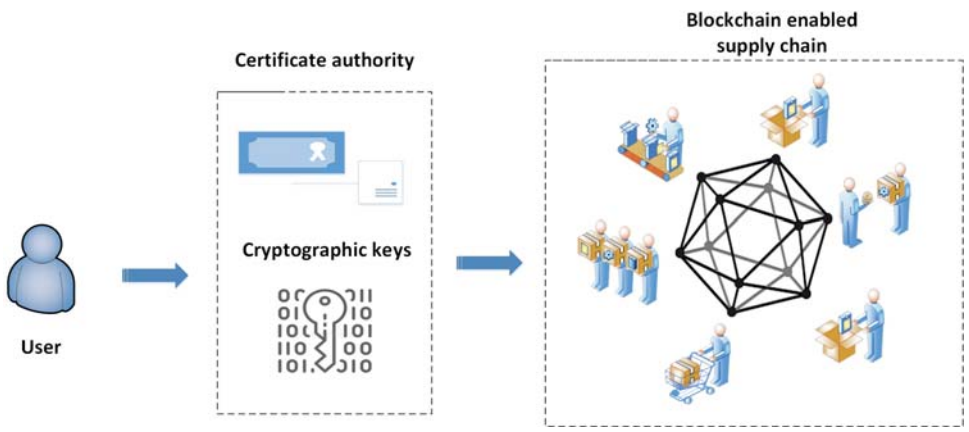


Figure 5.1: Blockchain-based SCM architecture.

In the proposed model, manufacturing and non-manufacturing stakeholders are part of a multi-tier supply network. In addition to vertical information sharing and cooperation, this method requires horizontal communication between stakeholders on the same SC tier. Suppliers and producers who use BC will collaborate by exchanging demand data and stock levels. The collaboration can be done through a permissioned BC, so only those members of the SC have access. It is easy to measure the effect of demand data because

all supply chain layers share the same demand data and inventory policy. The system model and assumptions are given below:

5.2.1.1 Assumptions

The following assumptions are provided in the SC model.

- The retailer tracks the demand of the end consumer and places orders at the top tier (e.g., distributor or manufacturer) using the (Q, R) inventory policy.
- Any number demanded by the retailer can be generated indefinitely by the producer.
- Out-of-stock orders are not lost at any point; instead they become backlogs that will be executed as soon as the inventory is replenished.
- The actual demand cannot be predicted ahead of time.
- Orders might be positive or negative and cancellations are permitted.

The credentials are obtained from a CA, consisting of a set of public and private keys and a digital signature. If a situation occurs, a CA has access to individuals' identities and may disclose the true identities of the stakeholders and their relationships. The following is an overview of the blockchain-coordinated SCM framework.

i Stakeholders will be assigned a minimum reputation score after receiving keys and joining the network.

ii When an order arrives at the retailer, the stock inventory is initialized. The demand quantity is reported in the blockchain to calculate the demand deviation. The supplier then determines whether the demand quantity can be met based on the current inventory level. If the inventory is greater than or equal to the demand quantity, the demand quantity is then removed from the inventory. Alternatively, if demand exceeds supply, the order will be sent to the upper echelon (e.g., distribution center or manufacturer).

iii When the relative inventory amount is less than or equal to the reorder point (ROP), an order is placed at the next higher echelon. Request ID and order quantity are among the details sent to the next higher echelon. The next upper echelon sends back information on this order's lead time and information about the order, including order ID, date of release, lead time and order quantity, which are also documented on the BC.

iv An order is shipped when the delivery date arrives. The upper echelon would also receive information about the order ID and actual delivery date. The stock amount is adjusted and the order is removed from the order receipt list.

v The replenishment quantity is added to the existing inventory as the lower echelon receives order delivery information from the next upper layer. If the estimated lead time differs from the order lead time, the estimated lead time is then updated.

vi The bullwhip effect (BWE) ratio is calculated by the difference between the order placed and the demand received and recorded on the blockchain.

vii After every order is received/shipped; inventory analysis is conducted to see if the relative amount of inventory is less than or equal to ROP.

viii All of the above steps are repeated when any stakeholders receive a demanded quantity from their lower echelon.

This chapter proposes a new consensus method, known as RPoC, for improving the throughput and scalability of a blockchain-based SCM architecture. A blockchain's consensus algorithm is at its core and significantly influences its security and efficiency. The essential features required for SCM applications are scalability, security and efficiency. RPoC utilizes a two-layer design that allows for quick consensus and scalability. By distributing the mining operations to all participating nodes, layering decreases the workload on individual nodes and increases consensus performance.

5.2.2 Network Model

This model Considers that the network is partially synchronous, which is the same assumption as Bitcoin makes [190]. A distributed peer-to-peer network of authorized nodes communicates via the network and maintains a shared state update. The connectivity between the honest nodes is well established, and the transmission time t between them is well-defined and minimal. Once a user broadcasts a message, the rest of the honest nodes will receive the message within a specified delay, Δ. Byzantine faults are also considered, as some network nodes may not be honest. The total number of nodes in the network is denoted by n, while the number of faulty nodes is denoted by f. The adversary's control is restricted to a maximum of f faulty nodes, where $3\mathsf{f} + 1 \leq \mathsf{n}$. [246] and [247] provide detailed proofs for interested readers.

5.2.3 Security Properties

The safety and liveness of the consensus algorithm must be demonstrated in order to prove its security. To begin with, RPoC is completely safe. Forks cannot occur as long as the number of Byzantine nodes is limited to f, even if no assumptions about network synchrony are made (i.e., there will not be a situation in which different nodes commit different blocks in the same round). The second point is that RPoC is live. RPoC achieves (eventual) liveness in a partially synchronous network, which means that new blocks are (eventually) added to the blockchain in a finite amount of time.
Given a blockchain network with a set of validators $\mathsf{V} = \{\mathsf{V}_1 \cdots \mathsf{V}_\mathrm{n}\}$, as defined pending transaction $T_\mathrm{x} \in \mathsf{T}$ and a pending block $\mathsf{B} \in \Omega$ subject to these properties are valid:

Integrity (safety): If a T_x is confirmed to the blockchain, it has already been published by a legitimate V_n and T_x is only committed to the blockchain once, so there is no duplication.

Finality (safety): If a valid B has been appended to the blockchain at time T, it becomes definitive and transactions within it cannot be reversed.

Validity (safety): If a valid B commits a transaction T_{x} in a block B, then T_{x} is committed, in the same block B, by every valid B.

Termination (liveness): For every transaction T_{x}, if a valid V_{n} commits T_{x} then all valid V eventually commit T_{x}.

5.2.3.1 Safety

Even with a slow and unstable communication network, RPoC is designed to offer safety. Once a block has been published to the blockchain by an honest node V, no other honest nodes V will ever append a different block for that round. The security of RPoC is dependent on the security of its underlying PoR protocol [120].

Claim 1 (RPoC is safe): Assume that the nodes running RPoC are $\mathsf{V} = \{\mathsf{V}_1 \cdots \mathsf{V}_{\mathrm{n}}\}$. By taking note of R_{t}, node $\{\mathsf{V}\}$'s reputation score, which gives it decision-making authority. Let B and B_0 be blocks appended to blockchain by honest nodes $I, j \in [n-1]$, respectively, in round k. Then $\mathsf{B} = \mathsf{B}_0$ in this case.

Proof: RPoC guarantees consensus safety If:

1. the adversary controls no more than f validators
2. or the validators compromised by the attacker have a total reputation score of R_{t}.

$$\mathbf{R}(t) = \frac{\sum_{i=1}^{|V|} R(\Delta_{\mathrm{T}})}{3} \tag{5.1}$$

Therefore an attacker cannot violate the safety requirement unless one of the conditions is not true.

5.2.3.2 Liveness

Liveness is a key feature of a decentralized system that ensures that the algorithm runs correctly in time and that valid and honest transactions are eventually complete. Even if

a conflict sometimes occurs, a liveness-favoring network will continue to run.

Claim 2 (RPoC is live): RPoC continues to proceed among n nodes, implying that regardless of the inner state of the nodes, some honest node will publish a new block to the blockchain within a finite time

Proof: There is a guarantee that all honest nodes' T_{x} will appear in some rounds and that all honest peers in the network will accept them. Assume that V_{n} has a high reputation and publishes T to the network, one of two things can happen: T will either be, or not be, received by peer nodes.

1. T has been received: because of the asynchronous environment, liveness is achieved for the V_{n} node.
2. There has been no notification of T: This occurs when V_{n} is malicious or shut down during the transmission.

5.2.4 Encryption Mechanism

Public key cryptography, such as elliptic curve cryptography, uses a public and a private key for each user. The mathematical operations of ECC are dispersed over an elliptic curve. A private key is a random number, whereas a public key is a point on the curve. By multiplying the private key in the curve by a generator point G, the public key is created. G is the starting point, also referred to as the generating point. The two parties that want to communicate information must first agree on using a curve and its parameters, such as the coefficients of a and b and the base point G to be used, before beginning the ECC process. The elliptic curve equation can be written as:

$$Y^2 = X^3 + AX + B \tag{5.2}$$

where $4a^3 + 27b^2 \neq 0$.

Elliptic curve encryption algorithms are preferred because they demand fewer processing resources and use smaller key sizes. ECC has a reduced growth rate and time complexity

of ($O_{\surd}X$). It also has a higher resilience to attack, reduced CPU and content utilization, lower network consumption and faster encryption[154].

5.2.5 Threat Model

The threat model describes the system's resilience to Byzantine behavior. There are two sorts of adversaries in a blockchain system developed for SCM applications. It could be external: participants may attempt to join the network or mimic an existing authorized entity. Or internal: malware or hacking can cause nodes that are correctly registered and have valid signatures to go renegade. In either case, an attacker's goal would be to get an invalid transaction approved and broadcast to the ledger [248]. Any attempt to prevent a legitimate transaction or block from being recorded in a blockchain is known as a blockchain attack. The proposed protocol is anticipated to be utilized in permissioned blockchains, where participants can communicate in a secure environment, but the reputation-based protocol is itself vulnerable to exploitation [120]. The current public key infrastructure is utilized for key management and as a state-of-the-art secure encryption technique. A variety of threats can target the blockchain network, the following attacks are considered in this framework:

Attack 1: The adversary attempts two simultaneous transactions with two different nodes in the network.

Attack 2: An attacker repeatedly engages in byzantine behavior.

Attack 3: An attacker creates numerous identities, offering network redundancy while lowering system security.

Attack 4: An attacker tries to destabilize the services of a targeted node by sending a large number of fake transactions to thence make it unavailable.

Attack 5: An attacker tries to control the network nodes to influence the consensus mechanism.

Attack 6: A malicious node pretends to be a legitimate node. It attacks the system only once its reputation score reaches a high threshold.

The attacker is presumed to be computationally prevented from exploiting cryptographic

protocols. Furthermore, the proposed system does not consider terminal attacks or key hijacking.

5.2.6 Design of the Proposed RPoC Algorithm

The DPOS method is not decentralized, as the authority continues to be concentrated in the hands of a small group of users. For scalability, DPoS foregoes decentralization. Therefore executing an attack is easier because fewer individuals are in charge of maintaining the network. Likewise, Ripple, PoC and PoI have decentralization issues. Therefore they are not viable choices for SCM. On the other hand, the PBFT algorithm's consensus model only works efficiently when the number of nodes in the distributed network is limited. PBFT does not scale efficiently because of its high communication cost that grows exponentially with each extra node in the network. The PoC protocol, from the proof-based consensus category, can be an adequate alternative for SCM because it does not require any resources or coins to invest. However malware may have the ability to disrupt mining operations.

Table 5.1: Frequently used notations.

Notion	meaning
N_{i}	Participating nodes in blockchain
Val_{a}	Higher authority validators
Val_{s}	Subordinate validators
Tn_{x}	Transaction generated by Ni
Hat_{n}	Total number of nodes in higher authority layer
Sat_{n}	Total number of nodes in subordinate layer
At_{k}	Malicious nodes

The algorithm has been designed considering the above evaluation. The acronyms used in this work are listed in Table 5.1. The proposed blockchain consensus algorithm has two steps, from the creation of a block to its confirmation: consensus node selection and trans-

action confirmation (block confirmation). A rigorous identity verification process must be completed before a node may join the network. If a node wishes to be a Val_{a}, it must confirm its true identity and agree to share it with the rest of the network. Second, the system generates the node's reputation value using a reputation algorithm and then analyses the node's credibility. Third, nodes that choose not to stake their real identification are pushed to the pool of Val_{s}.

As a result of the fair node selection method, the block addition procedure is optimized, and blocks can be added to the blockchain instantaneously after verification. Figure 5.2 depicts the algorithm's overall structure. There are two layers of N_{i}: Val_{a} and Val_{s}. To generate blocks, N_{i} are operating in parallel. Val_{s} is in charge of generating micro blocks and sending them to Val_{a}. These small blocks will be received by Val_{a}, who will verify them before combining them into a single block. The algorithm's fundamental feature is the ability to accurately and efficiently pick consensus nodes to work in parallel. Consensus nodes are chosen randomly from many nodes, based on: their reputation score, their willingness to stake their identity and the random selection algorithm that selects a subset of nodes for each cluster at random.

5.2.6.1 Consensus Node Selection

A Blockchain network is characterized as a peer-to-peer network made up of N_{i}. This algorithm divides validators into two layers: Val_{a} and Val_{s}. In order to determine node allocation into each layer, the layering setup requires the use of different methods. Therefore to establish a consensus node selection mechanism, the algorithm combines a random number-generating approach with the node's reputation score system.

5.2.6.2 Transactions Broadcasting

The stakeholder who provides a particular service during a transaction is known as the provider, whereas the stakeholder who assesses the service is known as the rater. Dur-

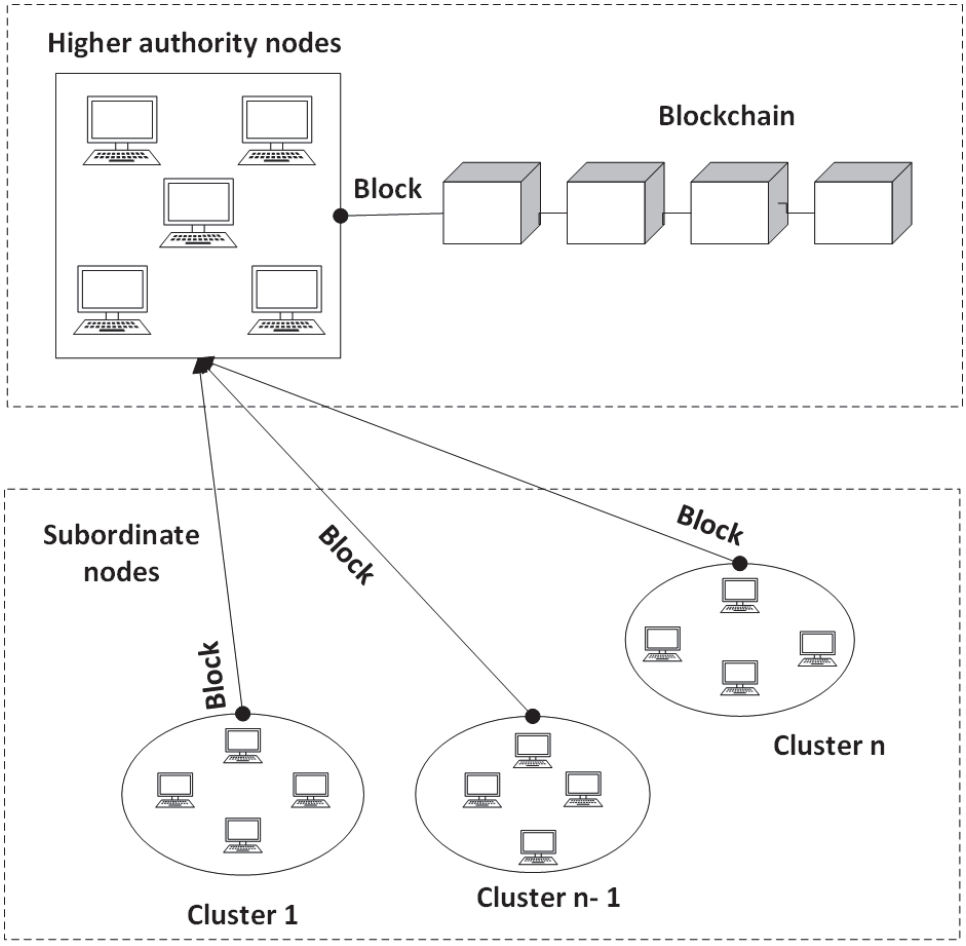


Figure 5.2: Layer structure of the proposed mechanism.

ing the transmission, the provider sends the requested service, which is signed using the provider's private key. The rater checks the data's integrity and prepares a transaction with the reputation score, which is broadcast to the rest of the network using digital signatures. The reputation score, denoted by R, can take values in the range $R_i^j \in (0, 1)$. For example, a manufacturer rates 1 to a supplier if it is satisfied with the service and 0 if it is not. In RPoC Val_{a} are chosen not only on the basis of their reputation (adopted from [120]) but also based on their proven identity. When a stakeholder joins the network for the very first time, it has no previous reputation score. So an initial reputation score $Rep_{\min}$ is assigned to the stakeholder, which is the minimal reputation score to continue operating

in the network. $H \in (0, 1)$ represents the stakeholder's honesty, which is set to "1" for each new joiner and to "0" if a stakeholder has misbehaved. When a stakeholder is selected as a validator, it is considered to be misbehaving if it sends conflicting signed messages to other consensus group members or commits mini blocks with conflicting transactions. The stakeholder's aggregate reputation $R(\Delta_{\mathrm{T}})$ at time Δ_{T} is calculated by combining the stakeholder's current and past reputation score.

$$R(\Delta_{\mathrm{T}}) = \sum_{t=0}^{t=i} Rep_{\mathrm{VAL}}(T) \tag{5.3}$$

where $Rep_{VAL}(t_0)$ is the initial reputation score of the stakeholder and $Rep_{VAL}(t_i)$ is the current reputation score of the stakeholder, happening at Δ_T. The stakeholder's reputation can be calculated regularly, with the time determined by the system's administration. A stakeholder must stay in the system long enough and conduct themselves honestly to build a high reputation score.

Nodes must be classified into different roles according to their reputation score. The node selection procedure is shown in Algorithm 6. The execution flow of Algorithm 6 is to select the nodes with the highest reputation scores (lines 2-8) for the higher authority layer. Along with reputation, the proposed algorithm also considers the value of identities in the algorithm, which implies that Val_{a} stake their real identities rather than any other resources. For Val_{a} a small number of validators are taken into consideration to create a scalable system. The Hat_{n} layer contains N_{i} with higher reputation scores and verified identity. The remaining nodes are grouped into clusters using a separate random selection procedure (lines 9-14). Each sub-layer cluster has a master node, which is chosen based on its reputation score. The master node oversees validating transactions and forwards them in a small block. In the case that the primary validator goes down or becomes unresponsive, the cluster's next highest reputation score node serves as a replacement. It is important to remember that these node roles are not fixed. A higher authority node's status changes as its reputation score changes after its tenure. For example consider a higher authority layer node, if its reputation score drops, it may become a validating or propagating node . A validating or propagating node may also become a higher authority node if its reputation score rises.

Algorithm 6 Algorithm of Selecting Consensus Nodes

procedure ConsNodeSELT($NodesN, AuthSTD$)
  $AuthNodeLst(len(n)) \leftarrow 0$
  $ClusLst(len(n)) \leftarrow 0$
  $MasterNodesLst(len(n)) \leftarrow 0$
  for all $i, n \in N$ **do**
    $AuthState \leftarrow AuthProcess(n)$
    if $(AuthState = T \wedge getRepScr(n) > Rep_{\min})$ **then**
      $AuthNodesLst(i) \leftarrow n$
    else
      $ClusLst(i) = Random(n \notin AuthNodeLst)$
    end if
  end for
  for all $i, nC \in ClusterLst$ **do**
    $MasterNodesLst(i) = Random(nC)$
  end for
end procedure

5.2.6.3 Block Confirmation

The steps for block confirmation are as follows:

- A node initiates a transaction $(Tn_{\mathrm{x}}, Sig_{\mathrm{c}}, T_{\mathrm{s}})$, where Tn_{x} is transaction, Sig_{c} is the client's signature and T_{s} indicates the timestamp.
- The cluster nodes receives and verifies Sig_{c} and T_{s}. If the verification is successful, the transaction $(Tn_{\mathrm{x}}, Sig_{\mathrm{c}}, T_{\mathrm{s}})_{\mathrm{cls}}$ is forwarded to the master node in the cluster, where $_{\mathrm{cls}}$ is the signature of the cluster node.
- The transaction must be verified by the master node. It verifies that the cluster node's signature is correct and that the transaction has not been registered in the blockchain. As soon as the verification is completed, the transaction is signed

$(Tn_{\mathrm{x}}, Sig_{\mathrm{c}}, T_{\mathrm{s}})$ $_{\mathrm{cls}}$, $_{\mathrm{m}}$, where $_{\mathrm{m}}$ is the master node signature.

- After signing, the transaction is pushed to the waiting pool. When there are a specific number of transactions in the pool, the master node packs each of them into a small block $(Smallblock_{\mathrm{TX}})$ $_{\mathrm{m}}$ and broadcasts it to the same layer.
- After receiving $(Smallblock_{TX})_m$, other cluster nodes verify the transactions included in the block. Upon successful verification, the master node receives $CONSENT, (Smallblock_{TX})_{SL}$.
- The master node can send $(CONSENT, Smallblock_{\mathrm{TX}}, \mathrm{Sig}_{\ \mathrm{SL}})_{\mathrm{m}}$ to the higher authority consensus group. Where Sig_{SL} is all the signatures from subordinate nodes.
- There may still be some Tn_{x} left in the pool after packing a small block; these Tn_{x} will be verified first in the new consensus round.
- After receiving a small block from Val_{s}, the nodes in the authority layer must validate the signatures and transactions of the small block.
- Once its verified, the higher authority nodes send an acknowledgment transaction $ACK_{\mathrm{accepted}}, Smallblock_{\mathrm{TXauth}}$ to the subordinate nodes. In some cases, if verification fails, rejection $ACK_{\mathrm{rejected}}, Smallblock_{\mathrm{TXauth}}$ is sent back to subordinates.
- The small blocks are put in chronological order after the verification is successful. A large block will be packaged and added to the blockchain after receiving a minimum of 10 small blocks.

Algorithm 7 presents the module for reaching a consensus on the verification of a block. N_{i} in the system are equally responsible for confirming Tn_{x} throughout the entire blockchain and work hand in hand to boost system throughput. By distributing the transaction verification process to every node in the network, the suggested approach enhances consensus performance while lowering the workload on miners.

Algorithm 7 Algorithm of Reaching a Consensus

procedure REACHCONS($MasterNodesML, AuthNodeAN$)
  $SmallLst(len(SL)) \leftarrow 0$
  $AuthSmallLst(len(SL)) \leftarrow 0$
  $BlockChain \leftarrow 0$
  $Block \leftarrow 0$
  for all $i, m \in SL$ **do**
    $SmallLst(i) \leftarrow generateSmallBlock(m)$
    $AuthSmallLst(i) \leftarrow SmallLst(i)$
    if then $AuthState \leftarrow AuthProces(AuthSmallLst)$
      $AuthNodeLst(i) \leftarrow m$
      $Block \leftarrow AuthState$
      $BlockChain \leftarrow Block$
    end if
  end for
end procedure

5.3 Evaluation and Results

5.3.1 Experimental Setup

The development of the proposed framework and all the results were obtained using Python 3.9 on a Windows 10 computer with an Intel Core i7 processor running at 2.21 GHz and 16 GB of memory in Visual Studio Code. Traditional PoW, DPoS, Ripple, PBFT and PoI consensus algorithms were also simulated and their experimental results are compared with the RPoC algorithm in order to justify the experimental results. There are several options available that can be taken into consideration for implementing private and permissioned blockchains. An x86-64 CPU system is being used since high performance is a crucial requirement for the use case. Although it is technically possible to mimic the network using an i3 processor and 4GB of RAM, some components can be slower, but the ratio of

benefits remains the same for the suggested approach.

5.3.2 Performance Evaluation

This section compares the experimental results of PoW, DPoS, Ripple, PBFT, PoI and RPoC consensus algorithms. The provided results are an average of 10 simulation runs. Three key factors: throughput efficiency, latency and scalability, are used to evaluate the performance of the RPoC consensus method.

1. Throughput: Throughput efficiency is expressed by TPS (Transactions Per Second), which can be measured by calculating how many transactions are completed w.r.t. time. It is used to measure how much processing a blockchain network is doing and how much scalability it has.

2. Latency: This measure is used to calculate the time it takes for a transaction to go from being sent to the network to being written to the ledger. This metric is calculated by comparing the time transactions take from when they were submitted to the time they were validated and stored using their timestamps.

3. Scalability: This metric evaluates the algorithm's capacity to continue to perform properly when its size or volume is modified. The re-scaling is usually to a larger size or volume.

5.3.2.1 Throughput

For measuring a system's efficiency, throughput is an important performance indicator. Starting with the initial transaction deployment time, throughput is defined as the number of executed transactions per second, where the average throughput is the total throughput divided by the execution time. In this set of experiments, the TPS value of all algorithms were obtained and compared. The graph of the average throughput of the consensus algorithms for various numbers of transactions is shown in Figure 5.3. As the number of

Tn_{x} grows from 1 to 100, the average throughput of all algorithms grows. PoI, DPOS and the RPoC algorithms have the highest throughput under 100 Tn_{x}, whereas PoW, PoC, PBFT and Ripple had the lowest. The average throughput of all algorithms drops after 1000 Tn_{x} as the number of Tn_{x} grows. Figure 5.4 presents a chart of the average throughput for DPOS, PoI and RPoC with varying numbers of Tn_{x}. As the number of Tn_{x} grows, RPoC's average throughput always exceeds that of DPOS and PoI. The PoI algorithm's TPS ranged between 6500 and 7000, while the DPOS algorithm's fluctuated between 3000 and 5000 and the RPoC algorithm's oscillated between 10400 and 95000.

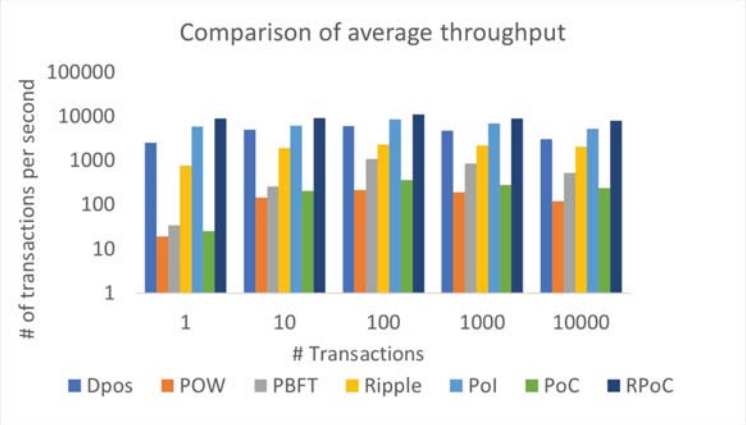


Figure 5.3: Average throughput with a varying number of transactions.

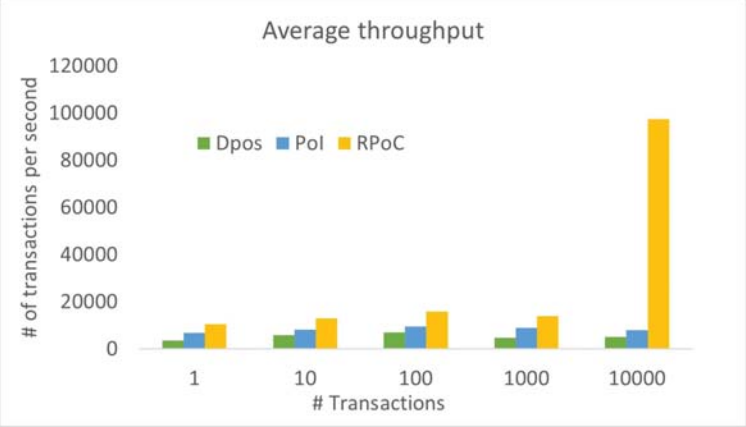


Figure 5.4: Average throughput of PoI, DPOS and RPoC with a varying number of transactions.

5.3.2.2 Latency

Latency is a key metric for assessing a network's performance and determining an algorithm's delays between nodes. A system with minimal latency is advantageous, since it can return transaction processing results more quickly. For instance, the block processing time frame for the PoW algorithm is around 10 minutes, which means that a transaction is successfully written to the blockchain after an average waiting period of 5 minutes [249].

In this test, the average latency performance of all consensus algorithms is evaluated with the same amount of Tn_{x}s, ranging from 10 to 10000. When dealing with a limited number of Tn_{x}, all algorithms have low latency. For instance while there are 100 Tn_{x} in the system, all algorithms have a low transaction processing latency, but as the number of Tn_{x} grows, the latency increases, as shown in Figure 5.5. In comparison to all other algorithms, PoW's average latency dramatically increased after 100 Tn_{x}. PoW's average latency when dealing with 10,000 Tn_{x} is 1307.56s, which is 900 times higher than RPoC. The RPoC algorithm offers a more consistent transaction processing latency that does not vary significantly as the number of Tn_{x} grows. The PoI, DOPS and RPoC algorithms are compared in Figure 5.6 to gain a clearer understanding. DPOS has a lower latency of 1.7s at 1000 Tn_{x}, compared to 1.5s at the same Tn_{x} for RPoC. In terms of latency performance, the DPOS and RPoC algorithms are competitive. In conclusion, the PoW and RPoC algorithms have significant latencies, whereas the DPOS and RPoC algorithms have shorter latencies.

5.3.2.3 Scalability

Scalability allows a system to respond dynamically depending on the latest settings. The scalability of a distributed system is determined by how the consensus mechanism allows for flexible joining and removal of nodes. The impact of increasing or decreasing the number of nodes during the operation of the consensus algorithm was investigated in the scalability test. Each system's TPS and transaction latency was investigated with various numbers of nodes. This analysis was applied to the PoI, DPOS and RPoC algorithms.

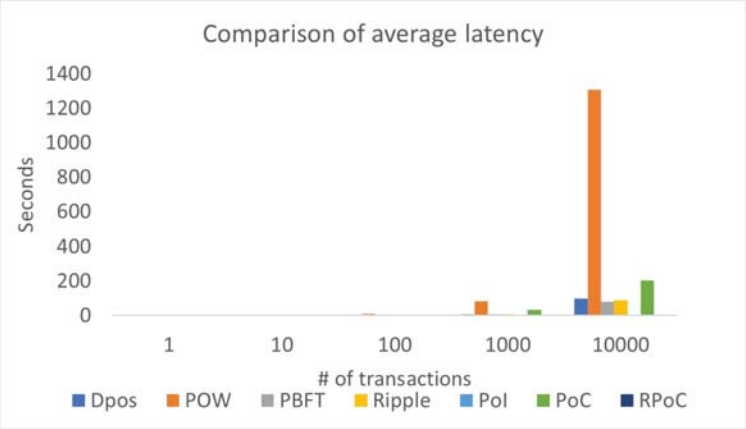


Figure 5.5: Impact of different numbers of transactions on latency.

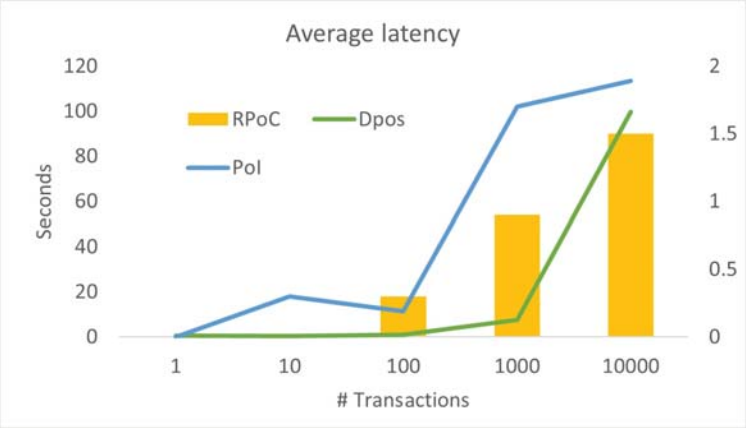


Figure 5.6: Comparison of average latency among PoI, DPOS and RPoC.

Figure 5.7 shows the transaction processing performance of the system under varying numbers of nodes. It is obvious from the plot that the performance of the DPOS and PoI algorithms degrades with higher numbers of nodes. When there are 50 and 100 nodes in the system, the TPS values of the DPOS algorithm are around 4500 and 3500, respectively. PoI has a little better performance than DPOS. However with 50 and 100 nodes, the proposed algorithm outperforms 8500 and 6900 TPS, respectively.

Figure 5.8 compares the average latency of the DPOS, PoI and RPoC algorithms with the same number of nodes. It is apparent that as the number of nodes increases, the average latency of each algorithm rapidly increases. The average latency for all three algorithms

is identical for a set of 10 nodes. But as the number of nodes grows, the latency starts to increase for both the DOPS and PoI algorithms. On the other hand, when the number of nodes is increased from 50 to 100, the average latency of PoI and DPOS increases about 2 times faster than that of RPoC. In this set of experiments, PoI, DPOS and the proposed algorithm all perform reasonably well. The performance of the consensus algorithms is not greatly affected by the growth in the number of nodes. The DPOS algorithm was found to have low scalability, but the other two algorithms have relatively high scalability. In some cases, PoI and DPoS performance is close to RPoC. It should be noted that the PoI and DPoS protocols are lottery-based Consensus algorithms that were designed to encourage coin circulation. The two algorithms could be well suited for cryptocurrency use cases, where it is crucial to maintain coin circulation, instead of keeping them in a hoarded state. However in the use case of the supply chain, trust in the network is highly desired.

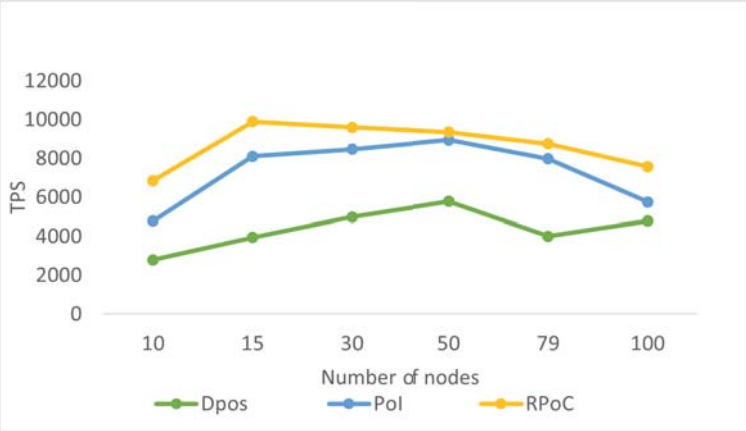


Figure 5.7: Average throughput with varying number of nodes.

5.3.3 Model Validation

In the previous section, the RPoC algorithm is compared with the conventional consensus algorithm; however to evaluate against proof of reputation consensus algorithms, the proposed RPoC is validated and tested against the Proof-of-X-Repute (PoXR) algorithm [250]. PoXR proposes a consensus mechanism that relies on the reputation of a system's

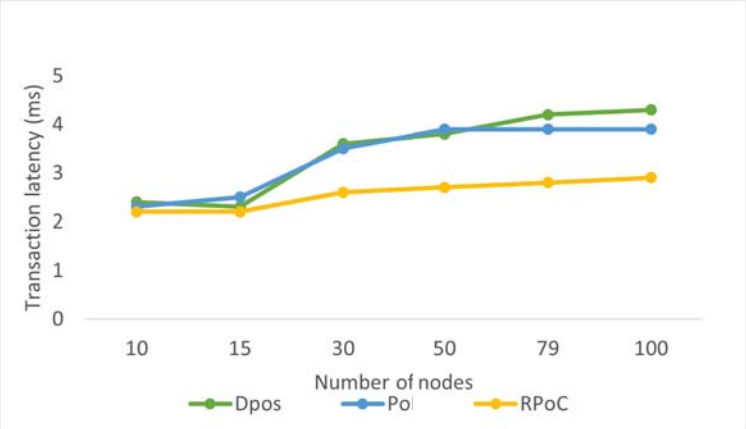


Figure 5.8: Average latency with a varying number of nodes among PoI, DPOS and RPoC.

nodes to lessen the difficulty of reaching PoX consensus in a public chain. In terms of how they work, the proposed RPoC and PoXR are polar opposites. RPoC is purely based on reputation scores, whereas POXR, like PoW, uses a mainstream protocol with a reputation layer. In PoXR, the likelihood of receiving the next honest block rises with an increase in reputation, making the process iterative. Furthermore, PoXR has issues with privacy preservation as each user protects their identity, which allows them to avoid being punished for malicious behavior.

In order to provide validation for the RPoC algorithm and an unbiased comparison, both algorithms are evaluated in the same setting (public network). Both models are compared in terms of throughput and security, and the average throughput performance of both consensus algorithms is compared with the same number of Tn_{x}s, ranging from 10 to 10000. Figure 5.9 shows the throughput performance comparison. Note that unlike PoXR, RPoC does not require resources to mine a block. Table 5.2 summarizes the important conclusions from the comparison of PoXR and the proposed RPoC in terms of attack resistance. In conclusion, the proposed approach works satisfactorily in terms of security and outperforms PoXR considerably in terms of throughput efficiency.

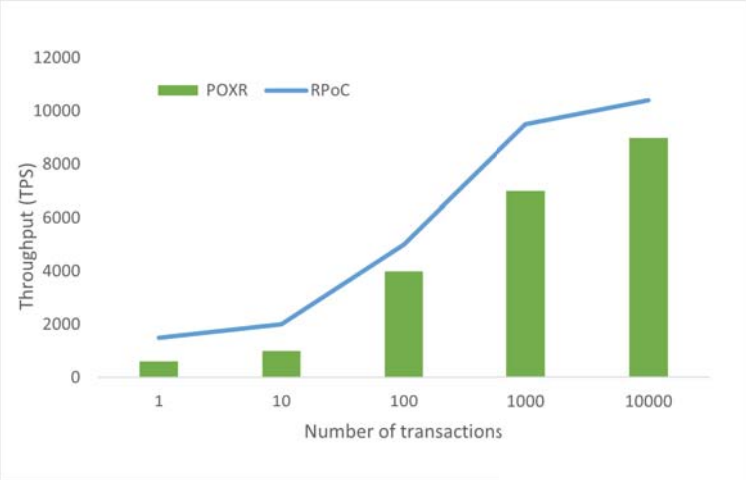


Figure 5.9: Comparison of average latency with varying number of nodes among PoXR and RPoC.

Table 5.2: Attack resilience.

Attacks	POXR	RPoC
Liveness	✓	✓
Selfish mining attack	✓	✓
Denial of services attack	✓	✓
Double spending attack	✓	✓
Sybil attack	✓	✓
51% attack	✓	✓

5.3.4 Security and Privacy Analysis

This section examines the security of the RPoC against a variety of malicious attacks as described in section 5.2.5. State-of-the-art secure encryption mechanisms are presumably in place and it is assumed that At_{k} will not be able to crack them. These threats are examined below:

5.3.5 Safety and Liveness

To demonstrate the consensus algorithm's BFT characteristic, the algorithm's safety and liveness must first be proved. In RPoC, attackers cannot use their mining power to break the system; instead they must develop a reputation and thereby contribute to the blockchain.

An attacker could never be among the top reputed miners in a network where there are trustworthy miners Val_{a}. For example, if the number of trustworthy miners is great enough, they all have a reputation score. However an outside attacker who does not have a reputation score can never become a member of the consensus group. Therefore the system's safety and liveness are always assured.

5.3.5.1 Double Spending Attack:

When At_{k} tries to do a second Tn_{x} with the same data that was already confirmed on the network, this is known as a double-spend attack. It assumes that At_{k} uses a double spending attack to transfer the same resource to two nodes in the network.

Defense: In RPoC, storing new blocks does not require solving a challenge or expending resources, it is predicted that a large number of validators will work in parallel. Since RPoC has two consensus layers, the network's large number of participating nodes will eventually recognize the double spending attack. Secondly, the blockchain's distributed nature itself prevents double spending attacks. Because all Tn_{x} are broadcast, validators will eventually receive blocks containing the double spend Tn_{x} and will be able to detect them during block verification. In this situation, At_{k} is removed from the validators list, and node details are sent to CA, preventing them from rejoining the network.

5.3.5.2 Attacks in Consensus Groups

Assume those malicious nodes are present across both layers that control the block generation and validation processes.

Defense: When the number of At_{k} in a cluster is less than 1/3, this consensus cluster has no effect on the generation of correct blocks. When the proportion of At_{k} in a subordinate cluster approaches 2/3, the At_{k} has the ability to package a fabricated mini-block. In this case, the higher authority nodes will create the correct large block, and then the fabricated mini-block will be recognized and excluded from large blocks and the subordinate cluster will be eliminated from the cycle after a certain amount of time. Consider that the ratio of At_{k} in the higher authority layer, which is responsible for appending blocks to the blockchain, is greater than 1/3. No matter how many fake mini-blocks are received, the proposed consensus protocol will only include correct mini-blocks. This is supported by the fact that the proposed protocol takes reputation into account when selecting block validators and creating blocks.

In addition to increasing liveness, RPoC is designed in such a way that it guarantees fairness by default, owing to its randomized validator selection process. Furthermore, RPoC distinguishes between safety, which is based on the reputation scores of the validators and liveness, which is determined by the framework.

5.3.5.3 Sybil Attack:

Sybil Attack is a sort of threat in which a At_{k} in the network deliberately operates several identities to compromise the legitimacy of reputation systems.

Defense: As previously stated, RPoC is a two-layer consensus mechanism in which N_{i} works together with a CA. Every node that wishes to join the network requires a unique id issued by the CA. Furthermore, Val_{a} are required to provide documents in order to identify themselves, and their true identities are visible to the entire network and are at stake; if they engage in any malicious conduct and are exposed, they will be unable to rejoin the network and will lose their reputation in the business community. As a result, RPoC defends against this attack. Furthermore, let's assume that At_{k} has the ability to generate several accounts. However each time the At_{k} starts a new account, it will be given a low default reputation score. With a lower reputation score per account, the At_{k} becomes non-competitive.

5.3.5.4 Denial-of-Service Attacks:

In order to disrupt the operations of a targeted network node and make it unavailable, At_{k} sends a high number of Tn_{x} to block it.
Defense: It is feasible to protect against this attack using the RPoC mechanism: Block generation rights can only be assigned to nodes that can withstand DoS attacks, since network nodes are pre-authenticated. In the case of when a validator is offline for an extended period of time, it can be removed from the validating node list. RPoC safeguards against this attack while also taking advantage of the blockchain's distributed nature.

5.3.5.5 Under 51% Attack

The 51% attack demands that At_{k} gains control of 51% of nodes in the network.
Defense: Getting control of the nodes in a permissioned blockchain network is far more challenging than controlling nodes in a public blockchain network. In a permissioned setting, the adversary cannot control the majority of nodes. Hence the honest majority assumption holds.
To further evaluate the proposed algorithm and analyze the behavior of existing protocols, a series of experiments were conducted with a proportion of malicious nodes in the network. Two scenarios were used in the experiments: one with 20% malicious nodes and the other with 45% malicious nodes in the network. 51% proportion has not been taken into account due to the fact that, in a permissioned ledger, malicious nodes can't control the majority of nodes, as described above. Figures 5.10 and 5.11 depict the presence of malicious nodes in the network. Existing consensus protocols focus on computational capacity, simple selection algorithms or voting, for selecting a validator node without taking reputation into account. Therefore if a malicious node is chosen as a validator, it will generate a block, solve the cryptographic puzzle and broadcast the block for validation to others. Other nodes will validate the hash values and keys of the produced block and validate the blocks, disregarding the block creator's reliability. These findings demonstrate that as the number of malicious nodes increases, all existing algorithms' resiliency declines. On

the other hand, the results show that no matter how many malicious nodes are present in the network, the proposed PRoC will only publish valid blocks to the ledger. This occurs because PRoC takes reputation into account when selecting validators for both layers. Along with that, the generation of blocks in the proposed protocol does not rely solely on a single validator. Consider the scenario where a malicious node gains a high reputation score by remaining honest for a long time and so is then able to become a master node in a subordinate layer. Further, assume that that node then generated incorrect/fake mini blocks; the higher authority layer' validators would then not allow that mini block to be included in the ledger.

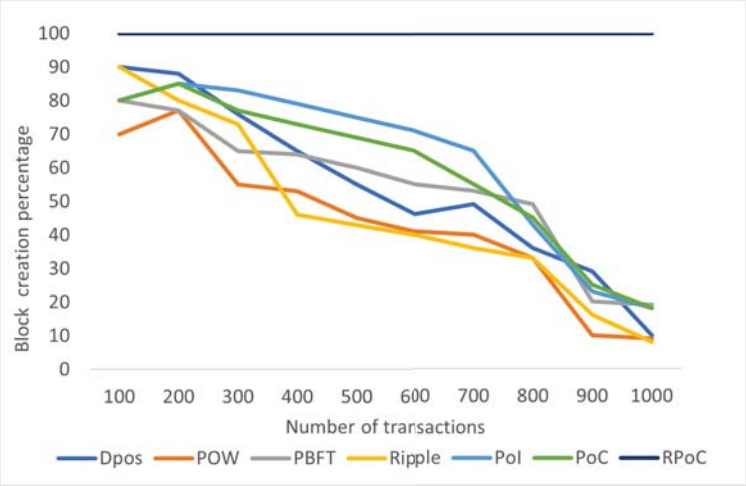


Figure 5.10: Block creation with 20% malicious nodes.

Fault tolerance The capability of a design to resist the failure of one of its nodes, is a part of what is referred to as fault tolerance. Since validators are in charge of storing new blocks, their failure might compromise an algorithm's fault tolerance. Multiple validators work together in the proposed approach to append blocks at the higher authority node layer, increasing the process's fault tolerance. These validators are chosen at random, based on their reputation scores and willingness to put their identities at stake and they change over time to maintain the system's fairness. For the subordinate node layer, if any master node in a cluster fails, the algorithm chooses a high-reputation node in the same cluster to immediately resume the verification process, as mentioned in Section 5.2.6.1. As

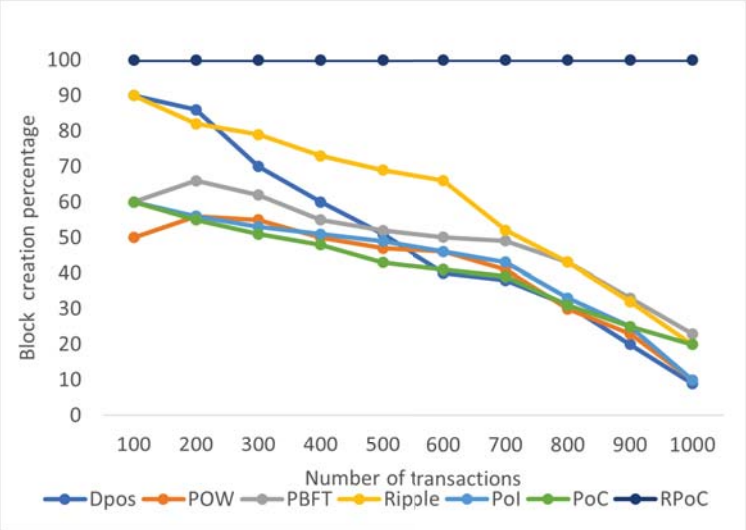


Figure 5.11: Block creation with 45% malicious nodes.

a result, a master node's failure has little influence on the transactions in that consensus cluster. This consistency, in terms of safety and liveness across the layers, leads to network reliability.

5.4 Chapter Summary

Blockchain technology has the potential to help stakeholders manage SCM more successfully with the right consensus algorithm. Scalability, low latency, high throughput and decentralization are desirable characteristics of a successful consensus algorithm and directly impact a blockchain's performance. However many existing blockchain consensus protocols are incompatible with SCMs. In this chapter, a new consensus algorithm, namely Reputation based proof of cooperation (RPoC), is proposed for blockchain-based SCM that does not involve validators to solve any mathematical puzzle, before storing a new block. The RPoC algorithm is an efficient and scalable consensus algorithm that dynamically selects the consensus node and permits many nodes to participate in the consensus process. The algorithm decreases the workload on individual nodes while increasing consensus performance by distributing the transaction verification process to every node. Furthermore,

this work highlights some current blockchain consensus algorithms and compares them to the proposed algorithm. Rigorous experiments against those existing consensus algorithms show the efficacy of the RPoC consensus algorithm in terms of TPS, latency and scalability. However, the proposed methodology has the following limitations: According to the well-known "blockchain scalability trilemma," it is impossible to create consensus algorithms that simultaneously accomplish security, scalability and decentralization. Due to the fact that we treat nodes differently based on their trust values, we cannot ensure complete decentralization. Our proposed framework also lacks detailed access control and identity management components, which are necessary to implement a practical reputation-based system effectively.

Chapter 6

Secure Online Blockchain Bidding System

This chapter presents a blockchain-based framework for an open-bid auction system, to address the privacy, security and scalability issues with blockchain that are discussed in Sections 1.2.1 and 2.3. The proposed blockchain architecture addresses privacy and security requirements by considering different cryptographic primitives. The novelty of this framework derives from an enhanced approach for integrating blockchain structures, by replacing the original chain structure with a tree structure. Throughout the online world user privacy is a primary concern because the electronic environment enables the collection of personal data. Hence this work proposes a suitable cryptographic protocol for an open-bid auction atop a blockchain. Here, the primary aim is to achieve security and privacy with greater efficiency, which largely depends on the effectiveness of the encryption algorithms used by blockchain. Essentially this approach considers Elliptic Curve Cryptography (ECC) and a dynamic cryptographic accumulator encryption algorithm to enhance security between auctioneer and bidder. The proposed e-bidding scheme and the findings from this study, should foster further growth of blockchain strategies.

6.1 Introduction

An auction is a method of selling products where a seller offers goods or services for sale, and bidders present the amount they are prepared to pay for them. Several types of auctions have been invented over the years. A Dutch auction is a technique where an initial price is set mostly by a seller and the price is reduced until a bidder accepts the current price [251]. Auction by sealed bid is a type of auction where bids are not open [252]. For it, all bidders send sealed bids that are simultaneously opened, and the highest bid wins the auction [253]. A Vickery auction is an auction where the highest bid wins, but the winner needs to pay only the second highest bids value [254]. The most well-known auction is English Auction, where the auction is won by the bidder who offers the highest price Birulin and Izmalkov [255]. The English auction protocol is more efficient to get a higher price for auctioned goods [256]. The estimated return on the auctioned goods using the English auction protocol is therefore usually higher than that of other protocols. Therefore most auction websites, such as eBay and Yahoo!, use an English auction and this has been reviewed and analyzed extensively in the literature [257]. Meanwhile, open bidding, as opposed to sealed bidding, can be a competitive method for vendors and may have many benefits [258]. In a sealed-bid auction a bidder can only submit one sealed bid and hence cannot change their bids based on competing bids. On the contrary, for the traditional English auction, participants can make several bids based on their competitor's bids. In recent years the technical development of the Internet has often succeeded in replacing offline auctions with online auctions, which are more far-reaching, more convenient, more efficient and more effective than the traditional way of conducting an auction [259]. Online auction systems use cryptographic mechanisms to be secure, but use a decentralized authority to handle seller-bidder transactions. This is because implementing a fully protected e-auction system that meets all security criteria involves very complex efforts in conventional design.

A major obstacle to the creation of a stable e-auction is fairness between bidders and auction servers. This is because it must provide a means to protect the exchange of information and to make electronic payments. Secure electronic payments must ensure

the process and provide convenience for all interested parties, much as traditional payment systems do [133]. The lack of confidence between parties is another critical problem within an e-auction network. The coalition, anonymity and link-ability of the online environment can lead to transactional misconduct. Another big problem with e-auctions is corruption by auctioneers [260]. A malicious auctioneer can manipulate auction procedures in a manner that is inconsistent with the rules of their auction. For instance, an auctioneer can opt to block offers, insert false bids, steal payments, profile bidders, open sealed bids before the winner selection process, or give the item to someone other than the actual winner. To solve these problems, many online auction schemes need assistance from trusted third parties (TTP) [261]. For such schemes, the agency's main role is to prevent possible threats to an electronic auction. For example, a third-party agency may effectively avoid any bidder coordinating with an auctioneer and maintain bidder anonymity and hidden bid prices. A TTP can be called upon to settle occasions when there is a conflict between an auctioneer and bidders. Yet such a configuration requires both parties to trust the TTP, which is a vulnerable security element and a bottleneck [262]. Wang et al. [263] used homomorphic encryption to propose an auction system that protects anonymity. It just reveals the group bid, and the auctioneer and agent all mask the bids of the users. They presume that the auctioneer and the agent are untrustworthy but are also not in collusion with each other and the scheme ensures that the auction will be safely carried out. TTP is commonly used, regardless of whether it is a trustworthy third party or a semi-trusted third party [264] [265]. It is unclear whether these conclusions are valid or not as extensive as for an actual situation. The third-party agent can be concerned about and reveal the bidders' details on the bids. The auctioneer and the agent may work with each other to make greater profit by rigging the auction results by maliciously interfering with the records, and the auctioneer or agent can be bribed by a bidder. In many online auction schemes, a not completely and trustworthy group exists. For instance, if the government is perceived as a potentially completely trustworthy party, then the issue of bribery arises. To solve the problem of protection needing a TTP, certain stable online auction protocols, without a TTP, have therefore been proposed. However certain entities in their networks can do certain unethical activities to make undeserved profit. TTP involves a lot of problems and

complex procedures, it is hard to establish a completely reliable third-party institution. So how can confidence, privacy and evaluation be maintained without a trusted authority, becomes an important question.

Blockchain technology is an integral part of today's digital economy. A blockchain is a distributed and decentralized ledger that does not allow the data stored in it to be changed without its peers' consensus. This property is commonly referred to as immutability, which is a core function of blockchain-based auctions [266]. Peers agree on the state of a ledger through a consensus protocol based on incentives. Many researchers have started looking at decentralized blockchains [267], as they have the potential to safeguard bid confidentiality, control leaks and prevent unethical practices [268]. Due to its decentralized nature, it could improve data validity with a lower cost in comparison to a typical e-bidding system. As well as maintaining a transparent and trusted application. Once a bid has been submitted to the auctioneer on a blockchain, the bid cannot be retracted and everybody can verify the winning bid. Many bid protocols have recently been added on top of blockchain . They take advantage of blockchain's decentralization and transparency properties to get rid of the weaknesses brought by a third party so that the information on the ledger can be checked and verified by all parties. In the bidding applications, there has been a growing trend towards blockchain adoption. Thus blockchain can resolve issues related to lack of trust or insufficient knowledge about a trading group, that would usually involve a mediator as the trusted party [269]. Adopting blockchain in the design of a stable e-auction system would save some of the efforts in this area. Nonetheless, previous research found that these protocols would make the scheme more complicated and complex. Despite cryptocurrency's ability and power, performance is lacking in the present form of blockchain technologies [210]. Typically each series of actions performed in a blockchain is propagated across the network and ends up being reported linearly on a blockchain. As a result, a lack of scalability is seen as a major challenge to the adoption of blockchain.

In this chapter, the lack of scalability of blockchain is argued to be problematic, despite blockchain being an efficient technology for managing auction systems, and this raises

concern about the acceptance of blockchain as a decentralized ledger operated by distributed nodes. These nodes generate transactions that are broadcast to the network, collected into blocks and, after a mining process, each block is added to its blockchain (by solving a cryptographic puzzle such as Proof-of-Work (PoW)). The current block hash is recorded in the subsequent block to thereby provide immutability of the stored data. Although blockchain has attractive features for e-bidding, its adoption in these applications requires careful consideration of scalability, security and privacy factors. For security and competitive reasons, data about bidders and their bids must remain private from other parties. However a scalable network architecture is required, given that the transaction load in e-bidding may increase significantly. Since the proposed architecture uses a private blockchain, it does not require high-overhead computations. In private blockchains the identities of its users are set up and the immutability cost conditions are smaller, because applicants are not anonymous and networks are usually not exposed to hostile public internet environments. Usually there is no native token or reward to encourage participants to participate and execute mining. Hence the blockchain would not have to offset the enormous cost of electricity that is required to avoid inflation in cryptocurrencies [270]. The prestige of the hash chain and replicas kept by multiple parties is generally adequate for immutability.

An open question in asymmetric encryption is the choice of which algorithm should be used when implementing a security system. Two of the most significant asymmetric cryptographic encryption algorithms are RSA and ECC encryption. RSA encryption was first described by Rivest et al. [35] when they introduced the concept of a public-key cryptosystem. The ECC algorithm was introduced by Chandel et al. [36] and uses elliptic curve operations, instead of modular exponentiation, as the basis for encrypting data. While there are some theoretical differences between these algorithms, they have largely the same capabilities. It then becomes important to consider which one provides faster performance when implemented, since the choice of a slow encryption algorithm could make a system inefficient and impractical for use in real life.

The goal of this chapter is to design a secure e-auction, based on an improved blockchain

for open-bid auctions. Although open-bid auctions have good theoretical characteristics, they have not yet been widely studied in terms of privacy and security. During the last decade, sealed-bid auction mechanisms have been studied and analyzed in different literature [271] [266] [149]. The privacy and security of open-bid auctions are important. In open-bid auctions, the identities of bidders should be protected, thus bidder privacy is desirable. Many auction protocols have recently been deployed on top of a blockchain. However there are conflicts in preserving the privacy of bidders and trusting the auctioneer to privately calculate the highest bid on the blockchain. Online auction systems use cryptographic mechanisms to be secure but use a decentralized authority to handle seller-bidder transactions. To solve these challenges, different cryptographic tools must be used to ensure the privacy of the bidders and the security of each bid, such as encryption algorithms and accumulators.

Most existing works describe security algorithms that use accumulators, however the efficiency of those algorithms has only been regarded in the theoretical context, rather than being implemented for a real use case. An accumulator is a binding commitment to a set of elements. A dynamic accumulator can change this set over time, as elements are added and deleted [272]. A dynamic accumulator can be used as both an authenticated data structure and a revocation authority. Most recent work on accumulators for blockchain has been introduced to reduce storage space and disk seek times [273] [274].

From an auction perspective, blockchain has been widely regarded by the community. However current studies have mostly been carried out from the perspective of sealed-bid auctions. There are only a few studies focusing on how to efficiently design a blockchain system for practical open-bid auctions. Meanwhile, among them, only a few have considered certain problems, for instance, security and privacy issues in the technical design of open-bid auctions. To this end, the scalability, security and privacy of open-bid auctions are the main focus of this chapter. This work combines blockchain with a new data structure, along with privacy-protection cryptography, to produce a distributed open bidding system that boosts bidding efficiency by allowing bidders to engage in the opening phase. Since the information on the blockchain is completely transparent and public, the bidders'

bidding documents must be completely confidential before the opening phase begins. The research focuses on the key issues of bidder anonymity, bid confidentiality and end-to-end verification. All these challenges provide the foundation of an effective open-bid system that preserves the integrity of the auction process. To do this the bids should be untraceable, to achieve strong relative privacy. The scheme must also have non-repudiation, so the winner must not be able to refuse their bid as they are obliged to claim it after the winning bid is determined. Therefore the relative privacy of each bid must be strong and at the same time recoverable.

6.1.1 Chapter Contributions

To address these challenges, an open-bid auction system is proposed using blockchain technology that makes the following novel contributions:

- A blockchain-based open-bid system architecture with a new design giving high throughput and search performance. The architecture uses a tree-based data structure to mitigate the scalability problem.
- A complete implementation of an e-bidding application without a reliable third party is established. The assessments reveal that the proposed framework has high throughput and search efficiency, compared to a blockchain framework that incorporates a linear structure.
- The platform focuses on bench-marking the performance of a concrete implementation of ECC encryption. Moreover, the ECC and RSA algorithms are compared to determine which encryption algorithm performs faster and provides better security. Furthermore, the use of an accumulator for authentication and revocation in an open-bid auction is highlighted.

6.1.2 Chapter Organization

The rest of this chapter is organized as follows. Section 6.2 discusses relevant background knowledge. Section 6.3 introduces the proposed blockchain framework for a bidding system and related components. The performance evaluation and security analysis are outlined in Section 6.4. Finally, Section 6.5 concludes the chapter.

6.2 Preliminaries

Some preliminary concepts used in this chapter (RSA and ECC), have already been covered in Chapter 2, while the rest will be covered in greater detail here.

6.2.1 Cryptographic Accumulator

One-way accumulators are an important cryptographic primitive that form the basis for many security systems. They provide a fixed-size digest, like a one-way hash function, representing an arbitrarily large set of inputs [275]. Moreover for any element of a set, a one-way accumulator can provide a fixed-size witness, which may be used in conjunction with the accumulated digest to verify the membership of that element in the set. As a result, cryptographic accumulators are often used in security applications that require some form of authentication as an alternative to digital signatures [276]. Dynamic RSA accumulator is based on modular RSA modulus exponentiation. The accumulator key is an RSA modulus, $N = pq$, where p and q are strong primes and a base $x \in Z_N$. The modulus should be at least k bits, where k is the number of bits in the largest element that will be accumulated [276]. Technically four polynomial-time algorithms form an accumulator scheme:

- Gen $(1^k) \rightarrow a$ generates the initial value of the empty accumulator, as well as any additional parameters, given the security parameter k.

- Add $(a, y) \rightarrow (a', w)$ takes in the current state of the accumulator and the value to be added, y and returns the new state of the accumulator, a0, as well as the corresponding witness, w.
- WitAdd $(w, y) \rightarrow w'$ takes in the current state of a witness, w and the new value, y, is added to the accumulator and returns an updated witness, w'.
- Ver $(a, y, w) \rightarrow (0; 1)$ takes in the current state of the accumulator, a, the value, y, whose membership in a is being checked and the witness, w and returns 1 if y appears to be in a and 0 otherwise.

Camenisch and Lysyanskaya [277] gave the RSA accumulator a way of being dynamic. They define as follows the deletion algorithm Del and an additional WitDel witness update algorithm,

- Del(a, y): compute $y' = y^{-1} \bmod \phi(n)$, where $\phi(n) = (p-1)(q-1)$
  set $a' = a^{y'}$
  return a'
- WitDel(w, y): compute $y' = y^{-1} \bmod \phi(n), where \phi(n) = (p-1)(q-1)$
  set $w' = w^{y'}$
  return w'

6.3 Blockchain Framework for a Bidding System

The use of e-bidding can make a procurement process fair and legitimate [278]. Currently they require a third party to manage the bidding procedure. Blockchain technology is used in this work to replace the third party. Figure 6.1 shows the use of blockchain technology in the bidding process, as discussed below.

1. Any authorized node can set up a tender and position it on the blockchain. The

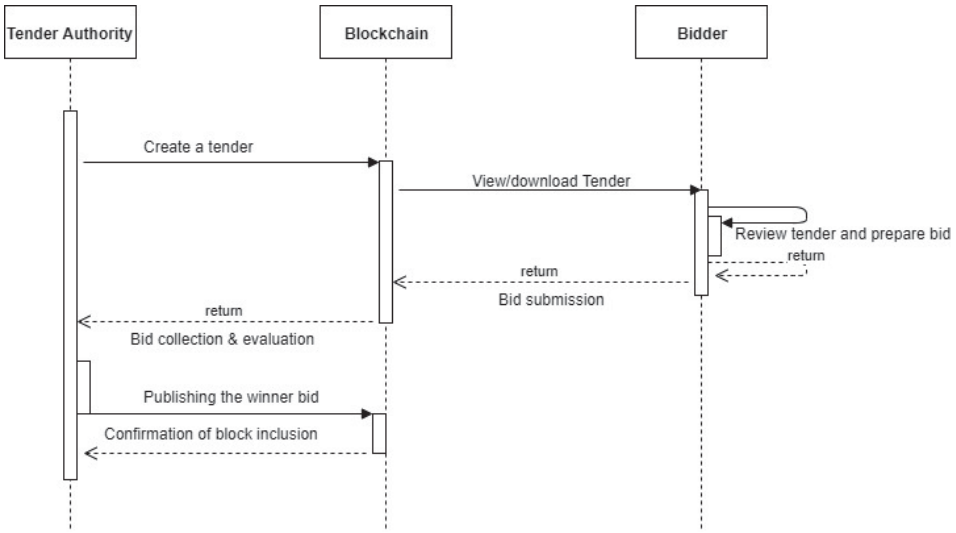


Figure 6.1: Blockchain based bidding architecture.

tender would include the terms and conditions of the offer, the information necessary for an appropriate bid and the requirements for the evaluation of any offers.

2. A prospective bidder will access the tender from the blockchain and after finding the tender requirements, make a proposal for an offer.

3. The prepared bid is published to the blockchain within a specified duration.

4. After the deadline expires, no one can submit a new or updated bid to the blockchain.

5. Once the tender is closed, the tender authority collects all the bids and starts evaluations to choose the best price.

6. The tender authority publishes the result on the blockchain and everyone who participated in bidding gets all the information about the winning bid.

6.3.1 Business Model

In this section, the open-bid auction system model is primarily discussed, which is realized by the blockchain technology's decentralization, data immutability and time irreversibility. Furthermore, all the interactions between the auctioneer and bidders are illustrated. There

are four phases from the initial deployment of the bidding system to the notification of the winner, as shown in Figure 6.2.

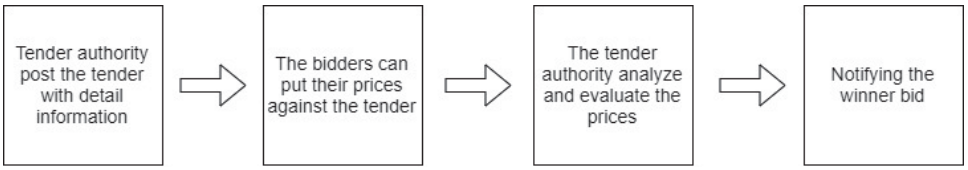


Figure 6.2: The flow chart of bidding system.

Phase 1:Placing Tender on blockchain

The invitation to bid includes what is needed, the criteria of the bid and how a bidder should respond. This is an opportunity for suppliers to make a reasonable offer to win a bid for the supply of their products or services, as highlighted in Algorithm 8. Even though the documents may vary from one organization to another, the basic elements of the bidding application documents involve:

- Definition of the goods and services to be obtained: this involves what the job will entail, any technical requirements and anything relevant to the conditions, deliverables or results of the project.
- Terms of bidding: These consist of the contract terms that must be met to qualify for the project.
- Criteria for evaluation: This specifies how the submission will be judged and assessed. This is then used as a guide when planning a submission.
- Content and format of submission: includes details on how to submit the request, which may refer to the duration of the submission, its layout, presentation, etc. Templates or types of response may also be issued.
- Application rules and data: the time limit for submitting, the place and time for application, what submissions must include, etc.

- Terms of contract: contains the general terms and conditions of the contract with additions and modifications made during the announcement of the successful bid. Flexibility in the proposed system has been offered by providing a limit that controls the number of bids a bidder can place for a particular demand.

Algorithm 8 Initiating a tender

```
1: procedure TENDER(_length,_limit)
2:     biddingTimeOut ← Time() + _length
3:     limit ← _limit
4: end procedure
```

Phase 2: Depositing Bids

In the second step, after getting the invitation, the demand is public to all the invited nodes. Any bidder's node who wishes to submit a bid can submit their bid price to the bidding authority as shown in Algorithm 9. This section describes the three variants of the open bidding system:

- To offload some of the computational complexity, bidders are authorized to place only one bid price against demand and they are not able to change it at any stage during the process.
- In this scheme, the system has flexibility in terms of changing bid prices. Once a bidder places their price, they can change it and the latest updated price will be considered for the demand. This would imply that it is no longer required to keep track of previously submitted bids. It would just consider and store the last updated bid price.
- An alternative scheme to make the system more open is presented. As in the previous scheme, bidders have the flexibility to update their bid prices. However every time when they change the bid price, it will be recorded. So the system keeps a record of all the changes made during the bidding period.

Algorithm 9 Depositing bids

```
1: procedure DepositBid(id, data )
2:     bidCount[id] = +1
3:     bid ← newBid(id, data, bitDeposited, biddingClose)
4:     bidDeposited.add(bid)
5:     return bid
6: end procedure
7: procedure BID(_id, _data, _bidDeposited, _biddingClose)
8:     id ← _id
9:     data ← _data
10:    bitDeposited ← _bidDeposited
11:    biddingClosed ← _biddingClose
12: end procedure
```

Phase 3: Finalizing the Bid

Algorithm 10 is a retrieval algorithm when the specified time is up for the bidding application. Each bid will be scanned for compliance and evaluated in accordance with the criteria set out in the bidding documentation. If a bidder fails to comply with this initial screening, it will be disqualified. Once the assessment process has been finished, the bidder will be chosen and informed of their successful bid and other failed applications will also be notified.

Algorithm 10 Finalizing the bids

```
1: procedure BIDS(_length, _limit)
2:     AfterBidding ← Time() > biddingClose
3:     if AfterBidding then
4:     return bidsDeposited
5: end procedure
```

Phase 4: Notifying the Winner

This phase determines who the winner is. Upon choosing the winning bid, the result will be written on the ledger and broadcast across the network. Key steps for broadcasting this are highlighted in Algorithm 11.

Algorithm 11 Broadcasting Winner

```
1: procedure WINNER(bidsDeposited)
2:     foreach bidsDeposited ∈ BIDS
3:         winner ⇐ Compare(bidsDeposited)
4:     B ← AddBlock(winner)
5:     Foreach n ∈ N
6:     Broadcast(n)
7: end procedure
```

6.3.2 Tree Structure Blockchain

Considering the importance of high performance, a new blockchain protocol has been introduced, which ensures high efficiency in search and throughput. Blockchain's data structure has been modified, as the linear data structure makes it more appropriate for a single-user or sequential process. So instead of using the traditional linear data structure, a tree-based data structure is proposed. The purpose of changing the data structure is to find a way to enhance the efficiency of searches. In the suggested approach, every block is related in a tree-like way to several other blocks, where the bids are sorted against a given tender. That makes the search system more convenient. For linear search, it takes approximately the same number of steps as the number of elements used to perform an operation of dimension N. When there are 1,000 elements, for instance, it takes about 1,000 steps. This is because, in the absolute worst case, every item must be visited once by the search. Unlike a chain, in a tree structure bids can be inserted in parallel. Parallel bidding enables the simultaneous occurrence of multiple tenders. Thus a tree structure changes the operation from a single-user to a multi-user system. The new data structure

supports the framework's dynamic nature and improves overall efficiency. In general tree structures are inherently quicker to edit and traverse if some logical branching structure is used. Admittedly there is a slight increase in size due to the links used by the tree.

The motivation behind changing the data structure is to find a way to improve the performance of searches by using a tree data structure instead of a linear one. So, in this approach each block is related to many other blocks in a tree-like manner, as shown in Figure 6.3. It can be interpreted from the Figure 6.3 that the number of bids for each tenders are different, whereas N=0,1,2,3,...., as shown in equation below:

$$T_2^N \neq T_3^N \neq T_N^N \tag{6.1}$$

where T_2, T_3 and T_N is the number of tenders and N is the number of bids in each tender.

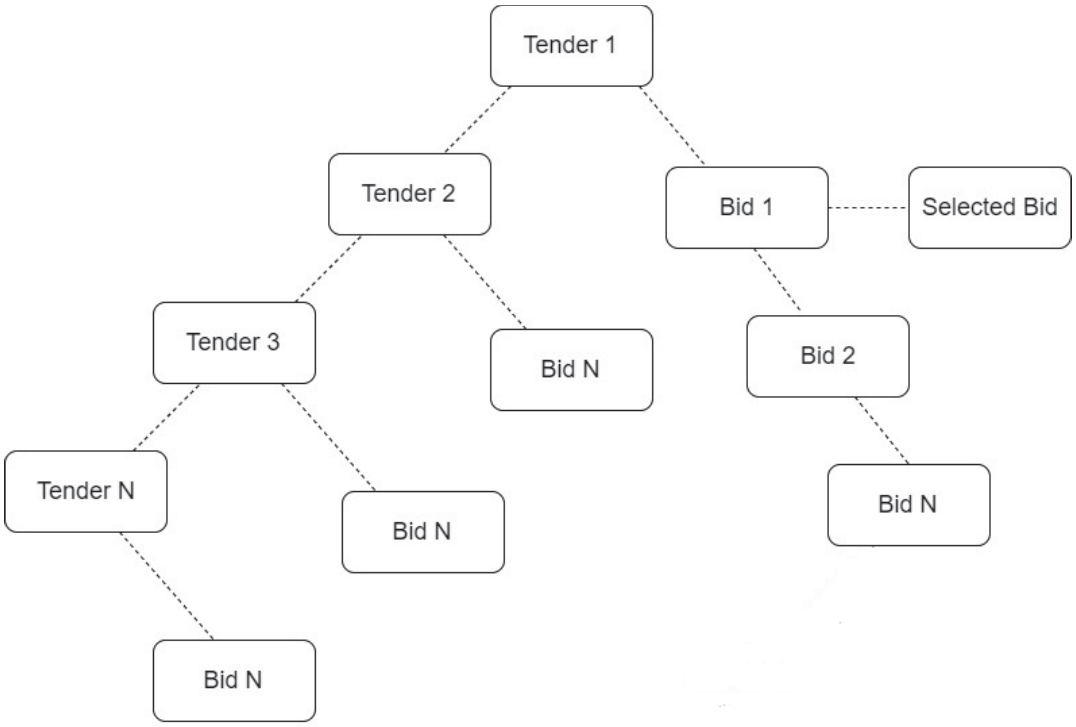


Figure 6.3: The tree structure of the bidding system.

Where the bids against a tender are sorted. That makes the system more convenient for

searching. For a linear search, in the absolute worst case, the search must visit every element once. This happens when the value being searched for is either the last element in the list or is not in the list. Nonetheless, on average, if the value sought is in the list and that each list element is equally likely to be the value searched, the search visits only n/2 elements. For the sake of simplicity let us assume that there were N offers, B bidders and b bids. The linear search structure is

$$N * B * b \tag{6.2}$$

while for the tree structure, it would be

$$N + B + b \tag{6.3}$$

The first research experiment was to calculate and compare asymmetric encryption algorithms, with and without using an accumulator. To achieve the best possible results, both the encryption algorithms and the accumulator were developed in Java and implemented with the fastest libraries available, which were also secure for threads. The whole process model is depicted in Figure 6.4.

If a bidder is participating in the bidding, the bidder generates a key pair and the key accumulator accumulates the keys for authentication and verification. The bidder submits the encrypted bid and the Key accumulator verifies the identity of the bidder. The auctioneer scans the signature, decrypts the bid and seeks to verify the tender. The Key accumulator at that stage verifies the auctioneer's identity. If no authentication error occurs the bid is accepted by the auctioneer and is posted on the blockchain. The generic requirements for a typical e-bidding system have been defined in Section 6.4.9 and the proposed blockchain-based open-bid system meets all of those security requirements. Since the bidder is registered in the system, the signature is created so that it is the unique identifier of a bidder and is protected from misuse due to the cryptographic hash's collision resistance property. Furthermore the proposed approach utilizes dynamic accumulators to enforce strong authentication mechanisms so that only authorized bidders can access the system.

ECC is suggested here because of its many advantages over other cryptographic algorithms. In order to evaluate the ECC output, the elliptic curve over a finite field with a

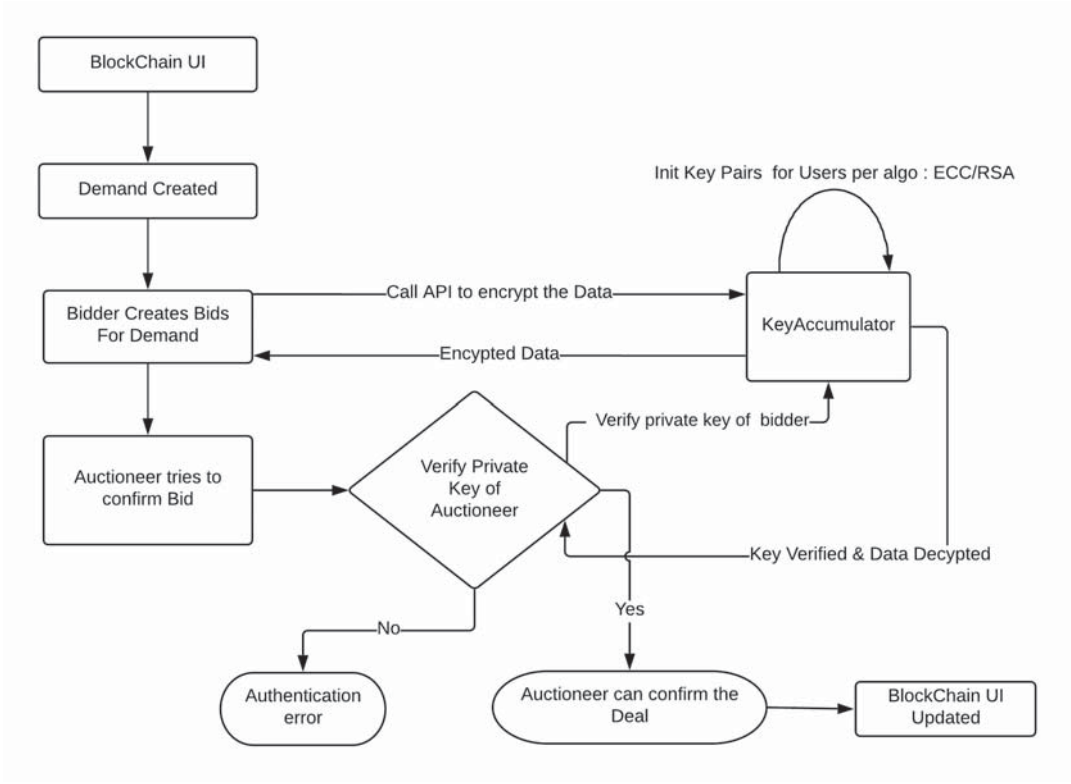


Figure 6.4: Data flow model of system architecture.

large prime number is considered, and measurements are taken in terms of the calculation time required to perform point multiplication on the curve and the computation time it takes to encrypt and decrypt a secret message. A cryptographic technique's security is largely based on the size of its key. Compared to other cryptographic techniques, ECC-based algorithms can use a smaller key size and still carry a high level of security. ECC is performed by considering various key sizes, namely 163-bit, 233-bit, 283-bit, 409-bit and 571-bit. In ECC a 163-bit key size means that the elliptic curve over a finite field is considered to have a 163-bit prime field size (prime number p is 163-bits long). For different prime field sizes, the same algorithm is then repeated. Table 6.2 provides details of the calculation time needed by ECC during the encryption process for different key sizes, which is the point multiplication method on the elliptic curve, decryption time, key generation time and signature verification time. The objective of the investigation presented in this chapter is to achieve security and privacy with the aim of achieving an efficient

performance level. Two distinct scenarios have been implemented, each corresponding to different settings, based on the type of encryption algorithm and with and without an accumulator. The first scenario was implemented using a dynamic accumulator with RSA and ECC, while the other scenario was implemented with RSA and ECC without an accumulator. The parameters monitored through these experiments were: key generation, signature verification, encryption and decryption, with respect to different key lengths. Finally, transactions were carried out in each of these scenarios to transfer a bid from a bidder to the auctioneer. The blockchain response was recorded and analyzed in the form of graphs based on the output data of these scenarios.

Scenario A: Assume that B_i is the ith bidder, bid_i be the bid price that is placed by B_i in an auction and SK_i and PK_i are B_i's private and public keys respectively. Assume also if one is needed, then V denotes the Dynamic Accumulator for the bid. For the first scenario with the accumulator set, the key pair SK_i and PK_i are generated by BCSytemKeyHndler for all the bidders B_i (which will be used for bidders to encrypt bids) for an open-bid auction. Algorithm 12 describes the process of accumulating keys. In the case of this experiment, the public keys have been accumulated to V for verification. For RSA encryption, the bidder encrypts the bid value using the RSA encryption algorithm described in section 6.3 and sends it to the auctioneer. The bidder will encrypt the message using the auctioneer's public key, and the auctioneer will decrypt it using their private key. Algorithm 13 describes the process of encrypting data. Moreover the bid can only be retrieved by the intended auctioneer holding the correct private key. So the encryption scheme is both private and secure (i.e., it authenticates and encrypts the data). The auctioneer collects the cipher text of bids and can decrypt them and verify the validity of their signature. Algorithm 14 describes the process of decrypting data. Thus the auctioneer computes the auction result. The user with the best bid is the winner. Finally the results will be written into the ledgers and published. The same series of steps are repeated with accumulator settings for the ECC algorithm. For this, the bidding value is encrypted and decrypted by using the ECC encryption and decryption algorithms described in section 6.3.

Scenario B: This scenario was used to perform further experiments to see the difference in both algorithms' efficiency, by removing the accumulator setting, while following the initial experiments described in scenario A. So this time the accumulator is disabled, thus keys are generated each time instead of being stored. This variation significantly influences the overall behavior of the system. Whenever a bidder wants to place a bid, in the first scenario the accumulator simply checks the signature because the keys are only generated once and stored there. Therefore not using the accumulator results in increased time and energy costs. Both the RSA and ECC encryption techniques are compared in this set of experiments.

6.4 Evaluation and Results

6.4.1 Performance Evaluation

A system prototype was developed that incorporates the proposed Tree data structure, the encryption technique and the authentication of the dynamic accumulator. The implementation was developed using Java 8 and the tests were carried out on an Intel Core i7 64-bit, 3.4 GHz machine, with an 8 MB cache and 16 GB of RAM, running windows. The timing of all the experiments was carried out using the computer's system clock. The Java Cryptography Extension (JCE) has been used for different hash and encryption features. For the performance evaluations, the proposed implementation was compared with an alternative implementation of a traditional blockchain which did not use a tree structure. A baseline traditional blockchain was taken into consideration for the alternative implementation and named linear structure blockchain, where the standard linear data structure was used. For a fair comparison, both linear and tree structures were incorporated in the same environment. By considering the number of transactions, a performance comparison of the tree structure blockchain with the linear blockchain system is provided. Next performance measurements of the throughput and searching complexity are presented, where the number of transactions varies from 1 to 100 transactions per second (tps) for a simulation interval of 100 seconds. In this section, the experimental results are analyzed and

Algorithm 12 Keys Accumulation/Generation

$L(keys) = List$ of user keys pairs

$AlgoType = ECC|RSA$

$KeyPair(User) =$ (public, private) $\in L$

$KeyFile(Location = SYS) =$ KeysData.csv

```
1: procedure SystemInit(L, AlgoType, KeyPair, KeyFile)
2:     KA ← KeysAccumulatorAndEncrypter()
3:     Read Secure.properties
4:     if AlgoType == ECC then
5:         init EccAcuumulator
6:         init EccKeyGenarator
7:     else
8:         init RSAAcuumulator
9:         init RSAKeyGenarator
10:    end if
11:    for entry ∈ KeyFile do
12:        if AlgoType == RSA then
13:            kpair ← RSAKeyGenarator
14:            L ← kpair
15:        else
16:            kpair ← ECCKeyGenarator
17:            L ← kpair
18:        end if
19:    end for
20: end procedure
```

evaluated. it also provides a qualitative security and privacy analysis, a proof of concept implementation and a quantitative performance evaluation of the framework.

Algorithm 13 Data Encryption

procedure CREATEDEMANDANDENCRYPTDATA(L,$KeyPair$,$KeyFile$)
  if $UserKey \neq RSA \wedge UserKey \in L$ **then**
    $PublicKey \leftarrow UserKey$
  else
    $kpair \leftarrow UserKey$
    $L \leftarrow kpair$
    $KeyFile \leftarrow kpair$
    $EncryptData$
  end if
end procedure

Algorithm 14 Data Decryption

procedure BIDCONFIRMATIONANDDECRYPTDATA( )
  if $Prikey(user) \neq null \wedge checkUserVerification == true$ **then**
    $decryptData$
  else
    $AuthenticationError$
  end if
end procedure

6.4.2 System Throughput

In blockchain technology, performance can be set in terms of transactions per second. Traditionally, public blockchain struggles with restricted scalability in terms of transaction throughput, transaction latency and requirements for storage. For example, standard cryptocurrencies have a fixed throughput (e.g. the bitcoin blockchain has a fixed bandwidth of 7 transactions per second [106]) whereas most advanced payment processing systems such as Visa have an average transaction rate of 2000 transactions per second. Parallel systems allow for different transactions to occur concurrently. In the proposed framework, several simultaneous bidding operations have been conducted, which increases

the system throughput. The number of tenders is increased in the proposed implementations to test system throughput. The different cases are illustrated in Figure 6.6. Each case was tested for a different number of tenders with varying numbers of bids. Note that even with 25 tenders and 50 bids, the computation cost is only 50.263 ms. These preliminary results suggest that the introduction of a tree data structure significantly decreases the computation cost, thus implying that the proposed framework can achieve high performance. To support the above statement, the performance of linear blockchain and tree-based blockchain were compared in terms of average throughput. Figure 6.5 shows a comparison of the average throughput for each system in five different sets of experiments. Tree blockchain's average throughput is significantly greater than linear blockchain's. Figure 6.5 also shows that the throughput increases linearly, as anticipated, until it reaches the limit at about 40 tps and then begins to decline beyond this point. So there is no point in preferring a 10 to 1000 shift. For any blockchain, the number of transactions is different and both are simulated in a similar way for consistency. The same pattern is noted in the linear blockchain method, where the throughput begins to decrease at about 40-50 tps. It can be observed that tree-based blockchain can process up to 27 transactions per second, compared to around 20 transactions per second by linear blockchain. From

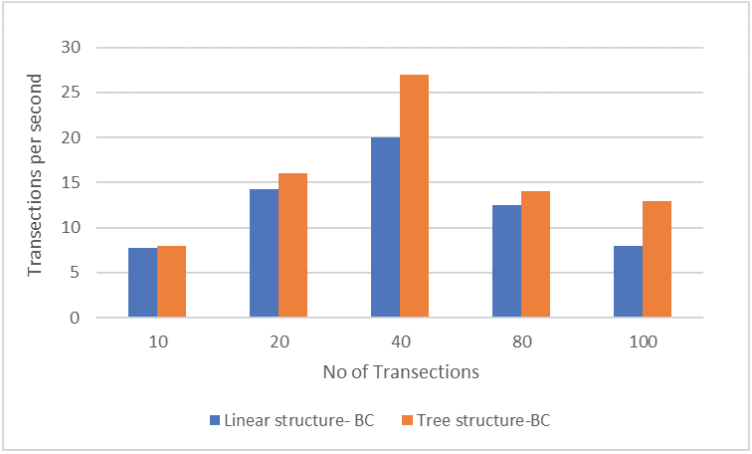
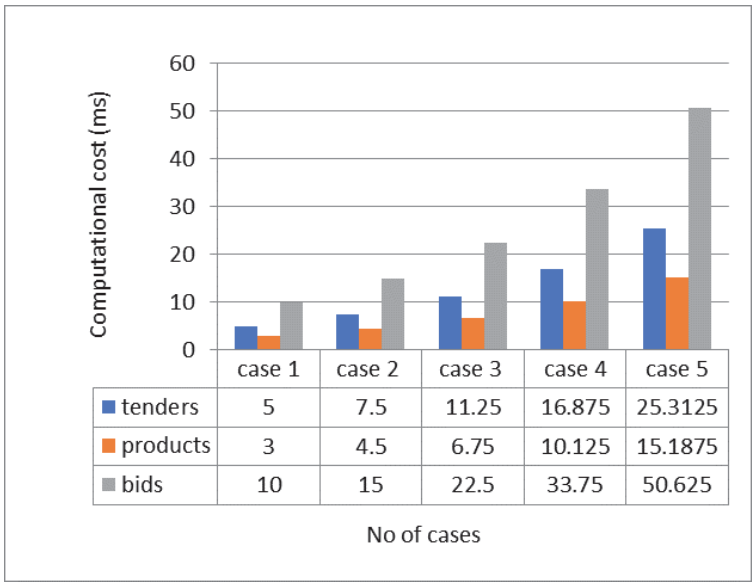


Figure 6.5: Average throughput comparison between linear and tree based blockchain.

the experiments, it was observed that the more tenders it has, the more bidders could be accommodated in parallel (as illustrated in diagram 6.3 that it can have N number of bids



	case 1	case 2	case 3	case 4	case 5
tenders	5	7.5	11.25	16.875	25.3125
products	3	4.5	6.75	10.125	15.1875
bids	10	15	22.5	33.75	50.625

Figure 6.6: Computation cost of the tree-structure blockchain.

against a tender and bidders can submit as many bids as they want against any tender). The system has been tested at an acceptable computational cost with 25 tenders. In view of the experimental results, the proposed framework achieves the quality needed to be implemented in a real-world environment.

6.4.3 Searching Complexity

Search complexity is a rough approximation of the number of visits to be taken, for finding a given value in a list of values, depending on the size of the input data. Specifically the effect of the number of multiple tenders on searching complexity was analyzed. For settings, the system starts with 5 tenders along with 10 bids. Then the number of tenders was increased gradually from 5 to 25 and the number of bids rose from 10 to 50. Investigating the searching complexity for both linear and tree data structure blockchains indicates that the proposed protocol reduces the number of searches. When using a linear search on a list of N items, one must first browse through half the list before finding the item. Therefore It will need to perform N/2 operations. The most significant thing, however, is that the algorithm scales linearly, as N increases, the algorithm's cost increases proportionally to

N, not N^2 or N^3. On the other hand, the tree search algorithm finds solutions by crossing the tree structure starting from the root node and systematically inspecting (expanding) the child nodes. In Table 6.1 and Figure 6.7, one can observe the difference between linear and tree searching time. For case 1 where it has 5 tenders along with 10 offers, the complexity of the tree search is five times less than the complexity of the linear search. From Table 6.1 the number of searches for the proposed protocol is much less than the number of searches for the linear data structure blockchain.

Table 6.1: Comparison of searching complexity.

Case No.	Tenders	No.of Bids	Linear time [s]	Tree time [s]
1	5	10	100	16
2	7	15	210	23
3	11	22	484	34
4	16	33	1056	50
5	25	50	2500	76

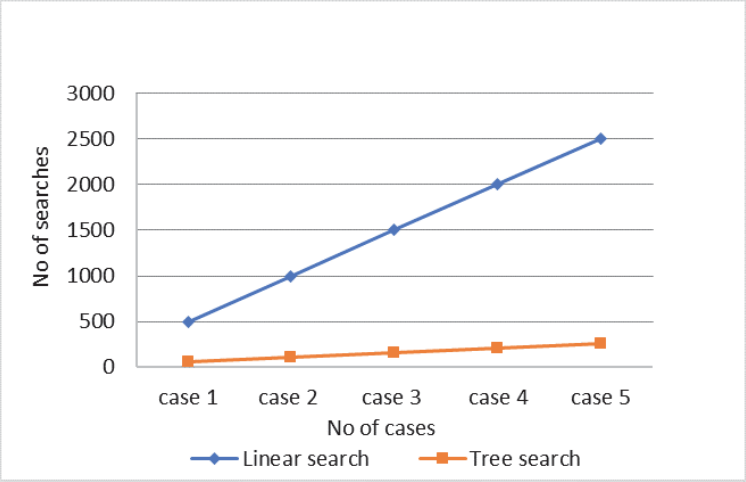


Figure 6.7: Searching complexity.

6.4.4 Public Verifiable Accuracy

The results of the bidding process with each user's bid message will be announced without revealing their identity. The bidder can verify that their offer has been registered on the chain. If the tender results are official, everyone can check its accuracy. Subsequently the winner will privately contact the auctioneer for any follow-up transactions.

6.4.5 Fairness and Correctness

This property guarantees that no internal or external hackers will be able to know the bidding patterns and outcomes, thus ensuring the fairness of the bidding process. It is obvious that after a certain time, that the bidders will get the result. It is possible to design the blockchain as an ideal public ledger, authentication is reliable and confidential and signature with high accuracy. Furthermore by evaluating the data, bidders can confirm that their prices have been correctly reported. If the result is inaccurate, it can be reported for further confirmation.

6.4.6 Rationality

This property means that no internal or external hackers have the ability to maliciously tamper with other people's prices, thus ensuring the integrity of the bidding process. No one can tamper with the data on the bidding price because the blockchain is inherently restricted in how it may be manipulated, thus ensuring that the bidding process remains fair.

6.4.7 Accumulators for Key Verification

Cryptographic accumulators solve the issue of verifying a public key without also having to incur significant overhead storage [279] and computational cost. An accumulator is a digital object which is used to test membership in a set. The accumulator will store

type tuples (PK_i, expT), where PK_i is a public key and expT is an optional closing time. The accumulator can then be used to determine whether pk has been registered or not. There are two ways in which accumulators can be integrated into a system: either each bidder can maintain their own accumulator, or a single accumulator can be maintained in the blockchain. In the proposed approach, one global accumulator will be stored on the blockchain. The tuple to be added to the accumulator is broadcast each time a public key is created or updated, much like a transaction in bitcoin. A single accumulator can be stored and maintained in the blockchain; all users can check that the new values are correctly incorporated in the updated accumulator and individuals who express the values will then determine their own witnesses, as all modifications are public and therefore locally reproducible. The experiments were run with and without an accumulator to see any differences. Figure 6.8 displays the time difference for RSA by using or not using the accumulator and and it is obvious from the graph that encryption and decryption are much quicker when using an accumulator. Figure 6.9 shows the comparison for ECC, with and without, an accumulator. Signature verification, encryption and decryption are much faster with an accumulator and show that using an accumulator improves the performance of the system. These preliminary findings indicate that the addition of an accumulator does not substantially increase authentication time, thus suggesting that the system can achieve scalability with minimal penalty.

Table 6.2: RSA dynamic accumulator average time per element in seconds, for each operation performed.

	Accumulation with PK	Accumulation with PK	Witness gen. with PK	Witness gen. with PK	Verification
RSA Accumulator	1.533	0.0026	0.0059	5.4446	0.0049

Table 6.2 shows the average time per element, in seconds, for each operation in the dynamic accumulator. The first and second columns show the time taken respectively to accumulate the private and public keys. The execution time for public and private key

witness generation is given in columns three and four, respectively. And the last column contains the times it takes to verify them. The table shows significant variations in the times for accumulation and witness generation using the private and public keys, the round-trip time per element is 10.2 ms and 5.4 seconds, respectively.

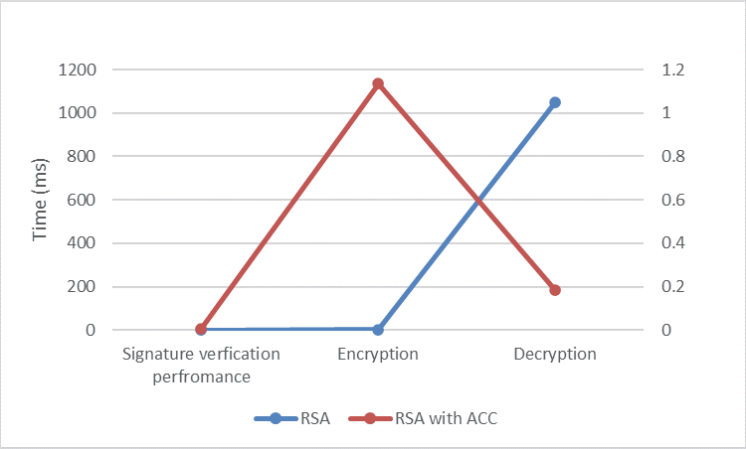


Figure 6.8: Performance comparison of RSA with and without accumulator in terms of signature verification, encryption and decryption.

6.4.8 Comparison of RSA and ECC

The widely used encryption algorithms RSA and ECC are compared in terms of their key generation time, encryption and decryption time and signature verification time. For RSA, 1024-bit, 2240-bit, 3072-bit, 7680-bit and 15,360-bit key sizes are used. In RSA, 1024-bit means that the modulus has a length of 1024-bits, i.e. is an integer greater than 2^1023 but lower than 2^1024. Such an integer can be encoded as a sequence of 1024 bits. Table 6.3 shows the average time for the ECC and RSA encryption algorithms. Note that the key size varies for both algorithms. For ECC, it is evident that the signature verification time increases linearly with the key length size. Also observe that the encryption time increases with key length size, which is also expected. For RSA, the key lengths do not have an impact on signature verification and encryption time. However observe that the rate of increase is greater with decryption time. Moreover, ECC outperformed RSA in both key and signature generation. RSA began by executing faster than ECC. Neverthe-

less, as the bit sizes for each increased, RSA slowed down relative to ECC and so ECC eventually surpassed its counterpart on the largest key length size. Lastly, ECC's times are considerably faster than RSA's with signature verification times and barely increased as key lengths rose.

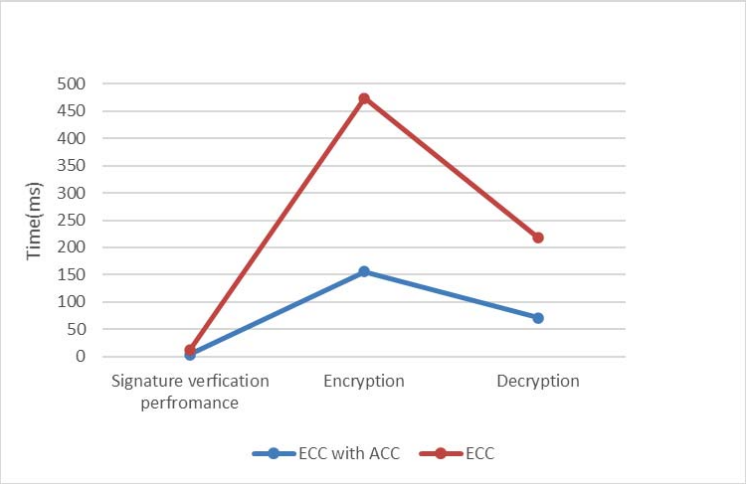


Figure 6.9: Performance comparison of ECC, with and without accumulator, in terms of signature verification, encryption and decryption.

6.4.9 Security and Privacy Analysis

The records stored in the proposed system are secured by means of public key cryptography, whereas network users are assigned private keys for signing and validating transactions. Encryption and digital signatures are used in the network to provide security, privacy and control of access to the stored records. Elliptic curve cryptography (ECC) has been used in the proposed approach to implementing encryption and digital signing, and it has also been compared to the RSA encryption algorithm. Note that ECC and RSA have a similar degree of protection but ECC uses far fewer bits. A 160-bit key in ECC, for example, offers the same level of security as one provided by RSA using a 1024-bit key length. Usually a shorter key means lower CPU consumption, lower memory utilization and faster key generation. These gains also benefit the proposed framework by facilitating rapid transactions when creating and sealing blocks. A comparison of RSA and ECC key

length, key generation performance, encryption, decryption and verification is provided in Table 6.3.

Table 6.3: Performance comparison of RSA and ECC in terms of key generation and performance, signature verification, encryption and decryption time.

Key Length		key Gen Time		Sign Ver time		Encryption time		Decryption time	
ECC	RSA	ECC	RSA	ECC	RSA	ECC	RSA	ECC	RSA
163	1024	0.10	0.20	0.23	0.01	8.90	0.55	22.88	19.31
233	2240	0.21	8.74	0.51	0.01	49.80	0.56	26.33	102.03
283	3072	0.29	10.80	0.86	0.01	67.40	0.57	27.40	209.60
409	7680	0.71	130.90	1.8	0.01	87.50	0.58	32.15	311.06
571	15,360	1.39	610.06	4.53	0.03	104.40	0.59	37.55	408.78

It is immediately clear from Figure 6.10 that as the key size increases, the difference between ECC and RSA becomes increasingly visible. It is clear that under the same degree of security, that ECC requires smaller key sizes than RSA. The minimum required key size for a stable ECC cryptosystem is 160 bits. Consequently the key size is chosen as 160 bits for ECC and 1024-bit for RSA as the starting point. The difference between the two algorithms is most noticeable at the 5th key size value.

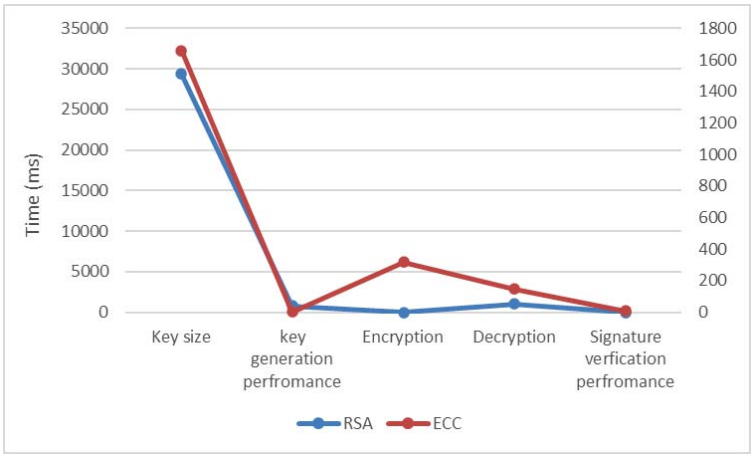


Figure 6.10: Overall comparison of ECC and RSA in terms of key size, key generation performance, encryption, decryption and signature verification.

Figure 6.11 confirms that ECC as the encryption algorithm, performs better than RSA. This also explains why when designing blockchain, ECC is preferred over RSA. This result is important because blockchain relies heavily on its encryption algorithms. According to the research findings, ECC meets all the features that are required to satisfy blockchain's security needs, better than RSA. Thus for the bidding protocol, ECC is preferred for data security.

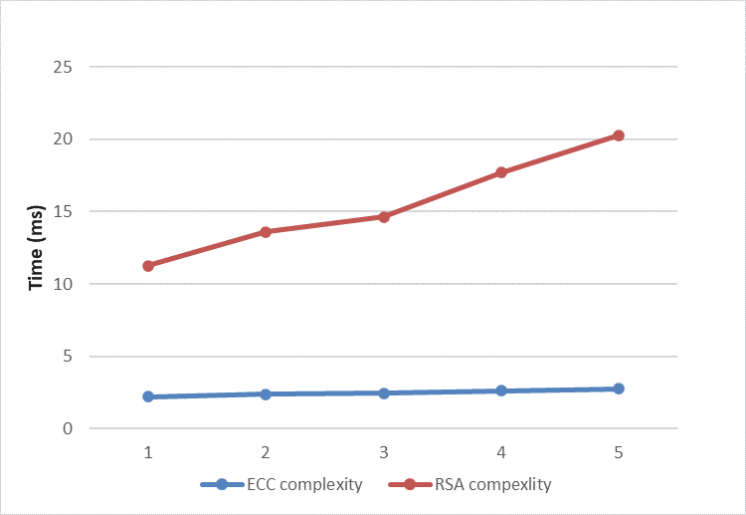


Figure 6.11: Algorithm's time complexity comparison.

Moreover the proposed framework is a decentralized p2p network where the user's data are stored in various nodes, thus ensuring the system's stability and preventing any single point of failure. Any opponent will have difficulty launching DDoS or DoS attacks [280] against the system, as every bidder is required to have a public key which will be verified by the accumulator. Thus DoS or DDoS attacks, where the attacker is motivated by exogenous incentives to stop a blockchain, would be prohibitively costly, as they require the attacker's mining resources to be at least equal to those of all other miners combined [281]. In addition to preserving security and privacy, the blockchain-based e-bidding system also offers several other benefits, such as scalability, verifiability and improved efficiency and speed. These features make blockchain technology useful in implementing an e-bidding system that can provide a convenient, safe and fault-tolerant channel of communication between auctioneers and bidders. Continuous implementation of different transactions

in a blockchain is increasingly required to handle smaller transaction sizes and greater consistency in transactions. All these specifications are closely related to the transactional encryption algorithms used. The small size of a key will also require less memory. The good key performance uses less time and provides a higher rate of transaction production. Good key verification performance will take less time to verify each transaction and thence provide higher speed [282]. Security requirements for the online auction protocol[283] are as follows:

1. Verifiability: Anyone can confirm the validity of B_i, the validity of a bid_i and the winning bidder's real identity. The algorithm is assumed to verify each bid for the authenticity of the signature. Through each user's bid request, the auction outcome will be accessible without exposing their identity. For B_i, they will verify that their bid on the chain has been registered. As the auction results are public, everyone can verify the correctness.

2. Non-repudiation: Signatures are hidden inside the bidding information and it has the characteristics of no framing. Therefore the winning B_i of the auction shall not be able to deny their signature. The blockchain itself has the features of non-repudiation and blockchain technology is the foundation for the proposed framework.

3. Robustness: The Elliptic Curve Cryptosystem can reduce the computation loads that are generated by bidding operations. By adding ECC, the amount of computation in the scheme is significantly reduced. The algorithm must verify whether or not each signature is valid and whether or not a ledger auction occurs when each B_i places a bid. There are no complex calculations for the proposed scheme, only a limited computational effort is needed in the framework and this is the strength of ECC itself.

4. No framing: The system achieves protection against framing threats in such a way that an entity cannot impersonate another legitimate B_i. In the protocol, B_i must prove that they know the SK_i of B_i in order to impersonate them. So unless an attacker gets B_i's SK_i, B_i's signature cannot be forged. It is difficult for an attacker to

obtain SK_i because of the Elliptic Curve Discrete Logarithm Problem (ECDLP)[284].

5. Unforgeability: Attackers will be unable to forge any valid bidding information because they must spend a great deal of time trying to solve ECDLP. The SK_i of any B_i cannot be obtained by other bidders, so their bid cannot be forged by anyone. Even if some B_i conspires with the auctioneer, and gets the winning B_i, if SK_i, attempts to claim to be the winning bidder, their scam will be revealed. If bidders have an altercation over the outcome of the auction, the actual winning B_i can demonstrate their signature to show their identity as the winner.

6. Unlink ability: The keys generated by BCSytemKeyHndler are different for each auction. No one else can know B_i's relationship with the various auction rounds.

7. Revocation: A revocation of B_i can also be carried out frequently in an electronic auction. It should then be quick and convenient if a B_i wishes to withdraw from an auction or an auctioneer wants to cancel a certain B_i. In the protocol, it is easy for an auctioneer to delete B_i'sPK_i from V. Once the information is removed from V, the bidder loses the right to participate in an auction.

6.5 Chapter Summary

In this chapter, an open-bid management system has been developed in a private blockchain environment, to provide a secure bidding scheme. The framework also contributes to providing a new protocol, by replacing its usual linear data structure with a tree data structure. This new protocol is proposed to improve the performance of the blockchain-based bidding framework. Additionally a dynamic accumulator design has been employed in conjunction with the ECC encryption algorithm for the open bid auction system to achieve security and privacy. A formal analysis and experimental findings of the proposed approach are provided, together with the details of the utilized protocols and models. Security analysis has demonstrated how the suggested approach is resistant to a variety of threats.

Chapter 7

Conclusions and Future Research Directions

This chapter in Section 7.1 presents a summary of the research carried out for this thesis and discusses its potential future study avenues for each of these contributions in Section 7.2.

7.1 Summary of Research Conducted

The specific contributions of this thesis are summarized below:

- *AccessChain* 3 was developed, it is a blockchain network coupled with the ABAC model to assure data privacy by adopting a distributed framework to enable fine-grained, dynamic access control management for supply chains. In order to solve the scalability issue, the framework helps by offering a two-tiered network design. A *global ledger* is used to record transactions, while access policies and business contracts are kept in multiple local ledgers. The framework enables a systematic approach that advantages the supply chain, and the experiments yield convincing

results.

A comprehensive analysis was done using Proof-of-Concept (PoC) implementation which demonstrated minimal overheads for AccessChain. Security analysis depicts how resilient the proposed framework is against a broad range of network attacks. AccessChain has distinct attributes that make it particularly suited for supply chains, including network scalability, security and privacy. Simulation results of performance monitoring show that *AccessChain's* response time with four local ledgers is acceptable, and therefore it provides significantly greater scalability.

- While exploring the application of blockchain technology to lessen the bullwhip impact in supply chains, a private blockchain-based architecture was presented, which focused on combining data visibility with trust 4. Full sharing of demand data has been shown to help improve the robustness of overall performance in a multi-echelon supply chain environment, especially for bullwhip effect mitigation and cumulative cost reduction. It is observed that when it comes to providing access to data, information sharing using a blockchain has some obvious benefits in a supply chain. Furthermore, when data sharing is distributed, stakeholders in a supply chain will have fair access to other parties' data, even though they are farther downstream. Sharing customer demand is important in an SC to enhance decision-making, reduce costs and promote the final end product. This work also explored the ability of blockchain technology as a solution in a distributed ledger approach to creating a trust-enhanced environment where trust is established so that stakeholders can share their information effectively. The findings have managerial implications, as they can allow businesses to exchange real-time data through BCT. Regardless of the consistency of existing coordination and communication in Supply Chain Management (SCM), managers should look more closely at BCT. The findings suggest that information lead time can be eliminated with a blockchain and thereby also decrease physical lead times. The proposed framework provides insights into the significance of SC information collaboration in reducing the bullwhip effect. In the event of a disruption in an SC, SC coordination helps to minimize the bullwhip effect. The inventory level in the proposed framework stays below Reorder point

(ROP) throughout the period, avoiding queues and lowering inventory costs. Furthermore the trust component gives stakeholders the confidence that they can safely share their data. Security analysis showed the robustness of the framework against several attacks. Extensive experiments were carried out to demonstrate the effectiveness of information sharing in a supply chain via blockchain, as well as that trust between partners tends to increase overall supply chain efficiency and reduce the bullwhip effect.

- A new consensus algorithm, called Reputation-based proof-of cooperation (RPoC), for blockchain-based supply chains was proposed 5, which does not involve validators to solve any mathematical puzzle before appending a new block to the blockchain. The RPoC algorithm is an efficient and scalable consensus algorithm that selects consensus nodes dynamically and permits a large number of nodes to participate in the consensus process. The RPoC algorithm uses a layered architecture to segment the nodes that participate in the consensus phase, in order to improve scalability and efficiency while maintaining trust among peers. The layered design addresses the issues of flexibility and scalability, as well as breaking down the extensive mining process into segments. Rather than choosing a few nodes for mining, the proposed consensus process involves all network nodes, making it more efficient, decentralized and scalable. The RPoC algorithm is an efficient and scalable consensus algorithm that dynamically selects the consensus node and permits many nodes to participate in the consensus process. Furthermore this work highlighted some current blockchain consensus algorithms and compared them to the proposed algorithm. Rigorous experiments against those existing consensus algorithms showed the efficacy of the RPoC consensus algorithm in terms of TPS, latency and scalability. Through extensive theoretical analysis and experimentation, the suitability of the proposed algorithm was shown to be well grounded in terms of scalability and efficiency. Security analysis showed the resilience of the security of RPoC against a variety of malicious attacks.

- A open-bid management system 6 was developed for a private blockchain environ-

ment to provide a secure bidding scheme. The novelty of this framework derived from an enhanced approach for integrating blockchain structures by replacing the original chain structure with a tree structure. This new protocol was proposed to improve the performance of the proposed blockchain-based bidding framework. Furthermore a dynamic accumulator architecture was utilized for the proposed open bid auction system with the Elliptic Curve Cryptography (ECC) encryption algorithm to achieve security and privacy. A formal analysis and experimental findings of the proposed approach was provided, together with the details of the utilized protocols and models. Security analysis demonstrated how the suggested approach is resistant to a variety of threats.

7.1.1 Significance of the Thesis

This thesis adds to the body of scientific literature, since it gives state-of-the-art data on blockchain solutions. The research's findings also present numerous avenues for future study and offer practical implications for managers and business owners. Besides answering the primary research questions, the thesis has also provided additional information related to the scope of the projects and future development.

7.1.2 Limitations of the Thesis

The proposed research in this thesis facilitates the development of blockchain-based systems for SCM. Although this thesis is representative of many frameworks for blockchain-enabled supply chains, it comes with a number of limitations.

- The Proof-of-Work (PoW) consensus algorithm has been used for the access point ledger in Chapter 3, but no comparison or evaluation of the PoW mining cost has been made.
- Second, this study's focus in Chapter 3 is solely on assessing the time-cost trade-off, latency, throughput and scalability matrix. Access control and ledger maintenance

costs are not taken into account.

- In Chapter 4, only information lead time is considered, whereas production and transportation lead time are neglected.
- The well-known "blockchain scalability trilemma" states that it is impossible to design consensus algorithms that provide security, scalability and decentralization at the same time. Due to the fact that in Chapter 5, nodes are treated differently based on their trust values, complete decentralization cannot be achieved.
- Chapter 5 also lacks detailed access control and identity management components, which are necessary to implement a practical reputation-based system effectively.
- Chapter 6 is focused on e-auctions, however no storage management system exists to manage extensive auction activities.
- This thesis takes a broad view of the blockchain and makes no distinctions between various blockchain technical configurations. The outcomes would have been more accurate as a result. Therefore, it is suggested that future research might focus on application software (Hyper-ledger Fabric, Ethereum etc.).

7.2 Future Research Directions

The research that is presented in this thesis contributes novel dimensions to the use of blockchain technology in supply chains. Solutions for data security, scalability and trust were provided. By developing PoC implementations, the scalability, security and computational overheads of each framework were evaluated. The following sections explore the prospective areas of further investigation for each main chapter of this thesis.

7.2.1 AccessChain

The AccessChain framework offers scalability by maintaining several parallel functioning local ledgers for the purpose of enforcing data access policies.

- Although the time-cost trade-off, latency, throughput and scalability matrix were calculated in this proposed model, it would be interesting to compute the cost measures of access control and ledger maintenance.
- One of the potential avenues for future research, is to implement the suggested solutions on an enterprise platform network in order to do additional real-time testing.
- PoW is utilized for the access point ledger, it would be interesting to compare and contrast the mining costs of PoW with other known consensus algorithms.
- Utilizing high-performance hash algorithms to improve the efficiency of data processing is another topic for investigation.

7.2.2 Blockchain-coordination for SCM

This thesis investigated how blockchain technology enables information sharing along a supply chain and equips each stakeholder to quickly assess the relevant situation. And, as a result of the ability to share real-time information, the supply chain became more productive and efficient.

- The models that were proposed for this thesis only took into account information lead time; a more realistic model may additionally take into account production and transportation lead times.
- The generalized model used in this thesis was specifically applied to a use case of retail supply networks. By adding additional variables, the model can be expanded to be adapted to a variety of decentralized applications where data sharing is crucial, such as perishable product supply chains, automobile supply chains and complex supply chains.
- As a future work, blockchain data can be utilized for new business and logistical models, like carbon trading and supply chain decarbonization. Different digital

value-added services may be offered by providing this kind of information. Therefore the creation of production economics models that might assess the costs and advantages of new information services and related business models can prove to be a very worthwhile endeavor.

- Another area for investigation is the post-pandemic problems that manufacturing, operations and SCM are now dealing with. For instance, new business and governance models brought about by changing consumer consumption patterns may be mostly irreversible in the near future, compelling businesses to create new business platforms using blockchain to replace their prior business models.

7.2.3 Scalable Blockchain for SCM

Integration of blockchain technology with SCM will result in new challenges and concerns. RPoC increases blockchain scalability by organizing a network into layers, where the higher authority nodes and subordinate nodes work together to manage transactions and blocks.

- In terms of economics, incentive design is a component that builds on the value proposition of a platform and constructs the system for which a platform's tokens will be built. Another research direction can be working on the economics of blockchain-based SCM to add a credit incentive mechanism to consensus nodes. Since cooperative behavior is more important in the design of supply chains, acceptable management mechanisms, like incentive and value-sharing mechanisms, are needed to enable the system to set monitoring, reward and penalty regulations.
- RPoC performance was examined using simulation-based PoC implementation. In future, an RPoC prototype implementation could demonstrate how well it performs in practical situations.

7.2.4 Secure online Blockchain Bidding System

A solution for an open-bid auction system is offered to address the security and privacy concerns in e-auction. In order to develop a general solution that can be used on top of any existing blockchain instantiation, the original chain structure was replaced with a tree structure.

- As the value of privacy protection continues to rise, one of the future research directions could be to investigate the use of cryptographic algorithms to ensure privacy protection methods at the network level, transaction level and application level.
- The focus of this thesis was the security of e-auctions, however it is possible to consider about employing new storage technologies or combining off-chain storage, to accommodate larger auction activities in the future.
- Blockchain-based supply chains and the integration of digital currencies are relatively new subjects, and there has not been much research done on them. A cryptocurrency could be adopted in the future to provide swift payments to stakeholders.

It is important to acknowledge that although our model may seem well-developed, it has not yet been tested in the practical context of supply chains. Implementing a blockchain-enabled supply chain model in practice can face several challenges and limitations. Here are some potential ones to consider:

1. Technical complexity: Blockchain technology can be complex and require significant technical expertise to implement and maintain. This can pose a challenge for companies that lack the necessary resources or expertise.
2. Resistance to change: Implementing a new technology like blockchain can require changes to existing processes and systems. This can be met with resistance from stakeholders who are comfortable with existing processes and may be reluctant to adopt new approaches.

3. Integration with existing systems: Integrating blockchain technology with existing systems can be a challenge, particularly if there are compatibility issues between different systems.

4. Regulatory compliance: The regulatory landscape surrounding blockchain technology is still evolving and can vary across different jurisdictions. Ensuring compliance with relevant regulations can be a challenge for companies looking to implement blockchain-based supply chain solutions.

5. Cost: Implementing a blockchain-enabled supply chain model can be expensive, particularly for small and medium-sized enterprises. The cost of hardware, software, and ongoing maintenance can be a barrier to adoption.

6. Interoperability: Ensuring interoperability between different blockchain networks and systems can be a challenge, particularly in complex supply chain networks that involve multiple stakeholders.

Collaboration and stakeholder engagement are essential to the successful implementation of blockchain-based supply chain solutions, particularly in complex supply chain networks with numerous stakeholders. Future studies can investigate various strategies for encouraging cooperation and involvement among various stakeholders, such as the use of participatory design techniques or stakeholder training/workshops. Future research that focuses on these topics can aid in addressing the difficulties and constraints associated with putting blockchain-enabled supply chain models into practice and encourage the successful adoption of these solutions in realistic supply chain environments.

References

[1] S. Ganesan, M. George, S. Jap, R. W. Palmatier, and B. Weitz, "Supply chain management and retailer performance: emerging trends, issues, and implications for research and practice," *Journal of retailing*, vol. 85, no. 1, pp. 84–94, 2009.

[2] G. J. Hahn, "Industry 4.0: a supply chain innovation perspective," *International Journal of Production Research*, vol. 58, no. 5, pp. 1425–1441, 2020.

[3] B. Bigliardi, S. Filippelli, A. Petroni, and L. Tagliente, "The digitalization of supply chain: a review," *Procedia Computer Science*, vol. 200, pp. 1806–1815, 2022.

[4] I. N. Pujawan and A. U. Bah, "Supply chains under covid-19 disruptions: literature review and research agenda," in *Supply Chain Forum: An International Journal*, vol. 23, no. 1. Taylor & Francis, 2022, pp. 81–95.

[5] C. Braziotis, M. Bourlakis, H. Rogers, and J. Tannock, "Supply chains and supply networks: distinctions and overlaps," *Supply Chain Management: An International Journal*, vol. 18, no. 6, pp. 644–652, 2013.

[6] F. Ricciotti, "From value chain to value network: a systematic literature review," *Management Review Quarterly*, vol. 70, no. 2, pp. 191–212, 2020.

[7] A. Kanda, S. Deshmukh *et al.*, "Supply chain coordination: perspectives, empirical studies and research directions," *International journal of production Economics*, vol. 115, no. 2, pp. 316–335, 2008.

[8] L. Liu, F. Li, and E. Qi, "Research on risk avoidance and coordination of supply chain subject based on blockchain technology," *Sustainability*, vol. 11, no. 7, p. 2182, 2019.

[9] M. Vosooghidizaji, A. Taghipour, and B. Canel-Depitre, "Supply chain coordination under information asymmetry: a review," *International Journal of Production Research*, vol. 58, no. 6, pp. 1805–1834, 2020.

[10] C. Chauhan, V. Parida, and A. Dhir, "Linking circular economy and digitalisation technologies: A systematic literature review of past achievements and future promises," *Technological Forecasting and Social Change*, vol. 177, p. 121508, 2022.

[11] A. Sheel and V. Nath, "Effect of blockchain technology adoption on supply chain adaptability, agility, alignment and performance," *Management Research Review*, 2019.

[12] J. Lohmer, N. Bugert, and R. Lasch, "Analysis of resilience strategies and ripple effect in blockchain-coordinated supply chains: An agent-based simulation study," *International journal of production economics*, vol. 228, p. 107882, 2020.

[13] D. Jimenez-Jimenez, M. Martínez-Costa, and C. Sanchez Rodriguez, "The mediating role of supply chain collaboration on the relationship between information technology and innovation," *Journal of Knowledge Management*, vol. 23, no. 3, pp. 548–567, 2019.

[14] K. Katsaliaki, P. Galetsi, and S. Kumar, "Supply chain disruptions and resilience: A major review and future research agenda," *Annals of Operations Research*, pp. 1–38, 2021.

[15] S. van Engelenburg, M. Janssen, and B. Klievink, "A blockchain architecture for reducing the bullwhip effect," in *International Symposium on Business Modeling and Software Design*. Springer, 2018, pp. 69–82.

[16] D. Ivanov, A. Dolgui, A. Das, and B. Sokolov, "Digital supply chain twins: Managing the ripple effect, resilience, and disruption risks by data-driven optimization, simu-

lation, and visibility," *Handbook of ripple effects in the supply chain*, pp. 309–332, 2019.

[17] A. Banerjee, "Blockchain technology: supply chain insights from erp," in *Advances in computers*. Elsevier, 2018, vol. 111, pp. 69–98.

[18] E. Tijan, S. Aksentijević, K. Ivanić, and M. Jardas, "Blockchain technology implementation in logistics," *Sustainability*, vol. 11, no. 4, p. 1185, 2019.

[19] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[20] A. Deshpande, K. Stewart, L. Lepetit, and S. Gunashekar, "Distributed ledger technologies/blockchain: Challenges, opportunities and the prospects for standards," *Overview report The British Standards Institution (BSI)*, vol. 40, p. 40, 2017.

[21] E. Foundation. (Feb. 2020) Ethereum 2.0 - the beacon chain. [Online]. Available: https://ethereum.org/en/upgrades/sharding/

[22] D. Khan, L. T. Jung, and M. A. Hashmani, "Systematic literature review of challenges in blockchain scalability," *Applied Sciences*, vol. 11, no. 20, p. 9372, 2021.

[23] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper, August*, vol. 19, no. 1, 2012.

[24] B. Bhushan, P. Sinha, K. M. Sagayam, and J. Andrew, "Untangling blockchain technology: A survey on state of the art, security threats, privacy services, applications and future research directions," *Computers & Electrical Engineering*, vol. 90, p. 106897, 2021.

[25] M. T. de Oliveira, L. H. Reis, D. S. Medeiros, R. C. Carrano, S. D. Olabarriaga, and D. M. Mattos, "Blockchain reputation-based consensus: A scalable and resilient mechanism for distributed mistrusting applications," *Computer Networks*, vol. 179, p. 107367, 2020.

[26] (2018) Maersk and ibm introduce tradelens blockchain shipping solution. [Online]. Available: https://newsroom.ibm.com/2018-08-09-Maersk-and-IBM-Introduce-TradeLens-Blockchain-Shipping-Solution

[27] M. Abramowicz, "Cryptocurrency-based law," *Ariz. L. Rev.*, vol. 58, p. 359, 2016.

[28] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Conference on the theory and application of cryptographic techniques*. Springer, 1987, pp. 369–378.

[29] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and G. Das, "Everything you wanted to know about the blockchain: Its promise, components, processes, and problems," *IEEE Consumer Electronics Magazine*, vol. 7, no. 4, pp. 6–14, 2018.

[30] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *International conference on financial cryptography and data security*. Springer, 2014, pp. 436–454.

[31] E. Zaghloul, T. Li, M. W. Mutka, and J. Ren, "Bitcoin and blockchain: Security and privacy," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 288–10 313, 2020.

[32] C. Chen, X. Chen, J. Yu, W. Wu, and D. Wu, "Impact of temporary fork on the evolution of mining pools in blockchain networks: An evolutionary game analysis," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 400–418, 2020.

[33] H. Delfs and H. Knebl, "Introduction to cryptography vol. 2 berlin etc," *Springer*, no. 0.004, pp. 0–008, 2002.

[34] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 839–858.

[35] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures

and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[36] S. Chandel, W. Cao, Z. Sun, J. Yang, B. Zhang, and T.-Y. Ni, "A multi-dimensional adversary analysis of rsa and ecc in blockchain encryption," in *Future of Information and Communication Conference*. Springer, 2019, pp. 988–1003.

[37] N. I. Koblitz, *Introduction to elliptic curves and modular forms*. Springer Science & Business Media, 2012, vol. 97.

[38] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," *Designs, codes and cryptography*, vol. 19, no. 2-3, pp. 173–193, 2000.

[39] M. Bednara, M. Daldrup, J. von zur Gathen, J. Shokrollahi, and J. Teich, "Reconfigurable implementation of elliptic curve crypto algorithms," in *Proceedings 16th International Parallel and Distributed Processing Symposium*. Citeseer, 2002, pp. 8–pp.

[40] K. Rabah, "Theory and implementation of elliptic curve cryptography," *JApSc*, vol. 5, no. 4, pp. 604–633, 2005.

[41] G. da Silva Quirino and E. D. Moreno, "Architectural evaluation of algorithms rsa, ecc and mqq in arm processors," *International journal of Computer Networks & Communications*, vol. 5, no. 2, p. 153, 2013.

[42] A. Kumar, S. Tyagi, M. Rana, N. Aggarwal, and P. Bhadana, "A comparative study of public key cryptosystem based on ecc and rsa," *International Journal on Computer Science and Engineering*, vol. 3, no. 5, pp. 1904–1909, 2011.

[43] W. Fang, W. Chen, W. Zhang, J. Pei, W. Gao, and G. Wang, "Digital signature scheme for information non-repudiation in blockchain: a state of the art review," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, pp. 1–15, 2020.

[44] R. Fotohi and F. S. Aliee, "Securing communication between things using blockchain

technology based on authentication and sha-256 to improving scalability in large-scale iot," *Computer Networks*, vol. 197, p. 108331, 2021.

[45] B. Preneel, "Cryptographic hash functions," *European Transactions on Telecommunications*, vol. 5, no. 4, pp. 431–448, 1994.

[46] H. Dobbertin, A. Bosselaers, and B. Preneel, "Ripemd-160: A strengthened version of ripemd," in *International Workshop on Fast Software Encryption*. Springer, 1996, pp. 71–82.

[47] J.-P. Aumasson, S. Neves, Z. Wilcox-O'Hearn, and C. Winnerlein, "Blake2: simpler, smaller, fast as md5," in *International Conference on Applied Cryptography and Network Security*. Springer, 2013, pp. 119–135.

[48] D. Eastlake 3rd and T. Hansen, "Us secure hash algorithms (sha and sha-based hmac and hkdf)," Tech. Rep., 2011.

[49] H. Gilbert and H. Handschuh, "Security analysis of sha-256 and sisters," in *International workshop on selected areas in cryptography*. Springer, 2003, pp. 175–193.

[50] M. S. Turan, R. Perlner, L. E. Bassham, W. Burr, D. Chang, S. jen Chang, M. J. Dworkin, J. M. Kelsey, S. Paul, and R. Peralta, "Status report on the second round of the sha-3 cryptographic hash algorithm competition," *NIST Interagency Report*, vol. 7764, 2011.

[51] C. V. Helliar, L. Crawford, L. Rocca, C. Teodori, and M. Veneziani, "Permissionless and permissioned blockchain diffusion," *International Journal of Information Management*, vol. 54, p. 102136, 2020.

[52] F. Benhamouda, C. Gentry, S. Gorbunov, S. Halevi, H. Krawczyk, C. Lin, T. Rabin, and L. Reyzin, "Can a public blockchain keep a secret?" in *Theory of Cryptography Conference*. Springer, 2020, pp. 260–290.

[53] X. Chen, K. Nguyen, and H. Sekiya, "On the latency performance in private blockchain networks," *IEEE Internet of Things Journal*, 2022.

[54] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.

[55] G. S. Ramachandran, S. Malik, S. Pal, A. Dorri, V. Dedeoglu, S. Kanhere, and R. Jurdak, "Blockchain in supply chain: Opportunities and design considerations," in *Handbook on Blockchain*. Springer, 2022, pp. 541–576.

[56] X. Xu, I. Weber, and M. Staples, *Architecture for blockchain applications*. Springer, 2019.

[57] O. Rodríguez-Espíndola, S. Chowdhury, A. Beltagui, and P. Albores, "The potential of emergent disruptive technologies for humanitarian supply chains: the integration of blockchain, artificial intelligence and 3d printing," *International Journal of Production Research*, vol. 58, no. 15, pp. 4610–4630, 2020.

[58] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," *Telematics and informatics*, vol. 36, pp. 55–81, 2019.

[59] H.-Y. Paik, X. Xu, H. D. Bandara, S. U. Lee, and S. K. Lo, "Analysis of data management in blockchain-based systems: From architecture to governance," *Ieee Access*, vol. 7, pp. 186 091–186 107, 2019.

[60] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, "A systematic literature review of blockchain cyber security," *Digital Communications and Networks*, vol. 6, no. 2, pp. 147–156, 2020.

[61] J. Park and R. Sandhu, "The uconabc usage control model," *ACM transactions on information and system security (TISSEC)*, vol. 7, no. 1, pp. 128–174, 2004.

[62] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the internet of things," *Mathematical and Computer Modelling*, vol. 58, no. 5-6, pp. 1189–1205, 2013.

[63] R. S. Sandhu and P. Samarati, "Access control: principle and practice," *IEEE communications magazine*, vol. 32, no. 9, pp. 40–48, 1994.

[64] G. D. Skinner *et al.*, "Cyber security management of access controls in digital ecosystems and distributed environments," in *6th International Conference on Information Technology and Applications (ICITA 2009)*, 2009, pp. 77–82.

[65] S. Pal, T. Rabehaja, A. Hill, M. Hitchens, and V. Varadharajan, "On the integration of blockchain to the internet of things for enabling access right delegation," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2630–2639, 2019.

[66] Y. Wang, M. Singgih, J. Wang, and M. Rit, "Making sense of blockchain technology: How will it transform supply chains?" *International Journal of Production Economics*, vol. 211, pp. 221–236, 2019.

[67] S. A. Abeyratne and R. P. Monfared, "Blockchain ready manufacturing supply chain using distributed ledger," *International Journal of Research in Engineering and Technology*, vol. 5, no. 9, pp. 1–10, 2016.

[68] D. D. F. Maesa, P. Mori, and L. Ricci, "Blockchain based access control," in *IFIP international conference on distributed applications and interoperable systems*. Springer, 2017, pp. 206–220.

[69] A. Ouaddah, A. Abou Elkalam, and A. Ait Ouahman, "Fairaccess: a new blockchain-based access control framework for the internet of things," *Security and communication networks*, vol. 9, no. 18, pp. 5943–5964, 2016.

[70] A. Ouaddah, A. Abou Elkalam, and A. A. Ouahman, "Towards a novel privacy-preserving access control model based on blockchain technology in iot," in *Europe and MENA cooperation advances in information and communication technologies*. Springer, 2017, pp. 523–533.

[71] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2018.

[72] S. Ding, J. Cao, C. Li, K. Fan, and H. Li, "A novel attribute-based access control scheme using blockchain for iot," *IEEE Access*, vol. 7, pp. 38 431–38 441, 2019.

[73] H. Liu, D. Han, and D. Li, "Fabric-iot: A blockchain-based access control system in iot," *IEEE Access*, vol. 8, pp. 18 207–18 218, 2020.

[74] S. Sun, R. Du, S. Chen, and W. Li, "Blockchain-based iot access control system: towards security, lightweight, and cross-domain," *IEEE Access*, vol. 9, pp. 36 868–36 878, 2021.

[75] S. Algarni, F. Eassa, K. Almarhabi, A. Almalaise, E. Albassam, K. Alsubhi, and M. Yamin, "Blockchain-based secured access control in an iot system," *Applied Sciences*, vol. 11, no. 4, p. 1772, 2021.

[76] J. Li, D. Han, Z. Wu, J. Wang, K.-C. Li, and A. Castiglione, "A novel system for medical equipment supply chain traceability based on alliance chain and attribute and role access control," *Future Generation Computer Systems*, 2022.

[77] C.-H. Liao, X.-Q. Guan, J.-H. Cheng, and S.-M. Yuan, "Blockchain-based identity management and access control framework for open banking ecosystem," *Future Generation Computer Systems*, 2022.

[78] F. Costantino, G. Di Gravio, A. Shaban, and M. Tronci, "The impact of information sharing and inventory control coordination on supply chain performances," *Computers & Industrial Engineering*, vol. 76, pp. 292–306, 2014.

[79] H. L. Lee, V. Padmanabhan, and S. Whang, "The bullwhip effect in supply chains," *Sloan management review*, vol. 38, pp. 93–102, 1997.

[80] Y. Huang and Z. Wang, "Values of information sharing: A comparison of supplier-remanufacturing and manufacturer-remanufacturing scenarios," *Transportation Research Part E: Logistics and Transportation Review*, vol. 106, pp. 20–44, 2017.

[81] G. W. Tan, "The impact of demand information sharing on supply chain network," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 1999.

[82] H. K. Chan and F. T. Chan, "Effect of information sharing in supply chains with flexibility," *International Journal of Production Research*, vol. 47, no. 1, pp. 213–232, 2009.

[83] G. P. Cachon and M. Fisher, "Supply chain inventory management and the value of shared information," *Management science*, vol. 46, no. 8, pp. 1032–1048, 2000.

[84] R. Dominguez, S. Cannella, A. P. Barbosa-Póvoa, and J. M. Framinan, "Ovap: A strategy to implement partial information sharing among supply chain retailers," *Transportation Research Part E: Logistics and Transportation Review*, vol. 110, pp. 122–136, 2018.

[85] F. S. Moghadam and M. F. Zarandi, "Mitigating bullwhip effect in an agent-based supply chain through a fuzzy reverse ultimatum game negotiation module," *Applied Soft Computing*, vol. 116, p. 108278, 2022.

[86] W. Jiang, "An intelligent supply chain information collaboration model based on internet of things and big data," *IEEE Access*, vol. 7, pp. 58 324–58 335, 2019.

[87] D. J. Ghode, V. Yadav, R. Jain, and G. Soni, "Lassoing the bullwhip effect by applying blockchain to supply chains," *Journal of Global Operations and Strategic Sourcing*, 2021.

[88] M. Hrušovskỳ and A. Taudes, "Battling the bullwhip effect with cryptography," in *International Conference on Database and Expert Systems Applications*. Springer, 2022, pp. 270–281.

[89] S. Malik, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "Trustchain: Trust management in blockchain and iot supported supply chains," in *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2019, pp. 184–193.

[90] Z. Tu, H. Zhou, K. Li, H. Song, and Y. Yang, "A blockchain-based trust and reputation model with dynamic evaluation mechanism for iot," *Computer Networks*, vol. 218, p. 109404, 2022.

[91] G. D. Putra, C. Kang, S. S. Kanhere, and J. W.-K. Hong, "Detrm: Decentralised trust and reputation management for blockchain-based supply chains," in *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2022, pp. 1–5.

[92] B. Bellaj, A. Ouaddah, E. Bertin, N. Crespi, A. Mezrioui, and K. Bellaj, "Btrust: A new blockchain-based trust management protocol for resource sharing," *Journal of Network and Systems Management*, vol. 30, no. 4, p. 64, 2022.

[93] D. Guegan, "Public blockchain versus private blockhain," 2017.

[94] T. Xu, T. Qiu, D. Hu, C. Mu, Z. Wan, and W. Liu, "A scalable two-layer blockchain system for distributed multicloud storage in iiot," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 9173–9183, 2022.

[95] M. H. Nasir, J. Arshad, M. M. Khan, M. Fatima, K. Salah, and R. Jayaraman, "Scalable blockchains—a systematic review," *Future Generation Computer Systems*, vol. 126, pp. 136–162, 2022.

[96] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *Ieee Access*, vol. 8, pp. 16 440–16 455, 2020.

[97] Z. Wan, W. Liu, and H. Cui, "Hibechain: A hierarchical identity-based blockchain system for large-scale iot," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[98] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: A secure, scale-out, decentralized ledger via sharding," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 583–598.

[99] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 931–948.

[100] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *2018 41st International Convention on Information and*

Communication Technology, Electronics and Microelectronics (MIPRO). IEEE, 2018, pp. 1545–1550.

[101] Y. Chen Jr, L. Tian, L. Yang Jr, L. Zhang, and Y. Fan, "Defects analysis of blockchain pow consensus protocol," in *Third International Conference on Computer Science and Communication Technology (ICCSCT 2022)*, vol. 12506. SPIE, 2022, pp. 425–430.

[102] X. Zhang, W. Ruizhen, M. Wang, and L. Wang, "A high-performance parallel computation hardware architecture in asic of sha-256 hash," in *2019 21st International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2019, pp. 52–55.

[103] D. Larimer, "Delegated proof-of-stake (dpos). bitshare whitepaper (2014)," 2014.

[104] L. Ge, J. Wang, and G. Zhang, "Survey of consensus algorithms for proof of stake in blockchain," *Security and Communication Networks*, vol. 2022, 2022.

[105] N. Bozic, G. Pujolle, and S. Secci, "A tutorial on blockchain and applications to secure network control-planes," in *2016 3rd Smart Cloud Networks & Systems (SCNS)*. IEEE, 2016, pp. 1–8.

[106] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[107] L. S. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 2017, pp. 1–5.

[108] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems (TOCS)*, vol. 20, no. 4, pp. 398–461, 2002.

[109] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzzyva: speculative

byzantine fault tolerance," in *Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*, 2007, pp. 45–58.

[110] B. Guo, Z. Lu, Q. Tang, J. Xu, and Z. Zhang, "Dumbo: Faster asynchronous bft protocols," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 803–818.

[111] T. Crain, C. Natoli, and V. Gramoli, "Red belly: a secure, fair and scalable open blockchain," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 466–483.

[112] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of bft protocols," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 31–42.

[113] P. Barrett, "Zilliqa technical whitepaper," 2017.

[114] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "Hotstuff: Bft consensus with linearity and responsiveness," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, 2019, pp. 347–356.

[115] Q. Dai, K. Xv, S. Guo, L. Dai, and Z. Zhou, "A private data protection scheme based on blockchain under pipeline model," in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*. IEEE, 2018, pp. 37–45.

[116] D. Schwartz, N. Youngs, A. Britto *et al.*, "The ripple protocol consensus algorithm," *Ripple Labs Inc White Paper*, vol. 5, no. 8, p. 151, 2014.

[117] J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun, and L. Li, "A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services," *IEEE Transactions on Services Computing*, vol. 12, no. 3, pp. 429–445, 2018.

[118] P. Zhang, M. Zhou, Q. Zhao, A. Abusorrah, and O. Bamasak, "A performance-optimized consensus mechanism for consortium blockchains consisting of trust-varying nodes," *IEEE Transactions on Network Science and Engineering*, 2021.

[119] T. Do, T. Nguyen, and H. Pham, "Delegated proof of reputation: A novel blockchain consensus," in *Proceedings of the 2019 International Electronics Communication Conference*, 2019, pp. 90–98.

[120] F. Gai, B. Wang, W. Deng, and W. Peng, "Proof of reputation: A reputation-based consensus protocol for peer-to-peer network," in *International Conference on Database Systems for Advanced Applications*. Springer, 2018, pp. 666–681.

[121] J. Yu, D. Kozhaya, J. Decouchant, and P. Esteves-Verissimo, "Repucoin: Your reputation is your power," *IEEE Transactions on Computers*, vol. 68, no. 8, pp. 1225–1237, 2019.

[122] J. Bou Abdo, R. El Sibai, and J. Demerjian, "Permissionless proof-of-reputation-x: A hybrid reputation-based consensus algorithm for permissionless blockchains," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, p. e4148, 2021.

[123] Q. Zhuang, Y. Liu, L. Chen, and Z. Ai, "Proof of reputation: A reputation-based consensus protocol for blockchain based systems," in *Proceedings of the 2019 International Electronics Communication Conference*, 2019, pp. 131–138.

[124] G. Fortino, L. Fotia, F. Messina, D. Rosaci, and G. M. Sarne, "A blockchain-based group formation strategy for optimizing the social reputation capital of an iot scenario," *Simulation Modelling Practice and Theory*, vol. 108, p. 102261, 2021.

[125] A. Mohsenzadeh, A. J. Bidgoly, and Y. Farjami, "A fair consensus model in blockchain based on computational reputation," *Expert Systems with Applications*, p. 117578, 2022.

[126] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-ng: A scalable blockchain protocol," in *13th {USENIX} symposium on networked systems design and implementation ({NSDI} 16)*, 2016, pp. 45–59.

[127] S. G. Stubblebine and P. F. Syverson, "Fair on-line auctions without special trusted

parties," in *International Conference on Financial Cryptography*. Springer, 1999, pp. 230–240.

[128] M. Naor, B. Pinkas, and R. Sumner, "Privacy preserving auctions and mechanism design," in *Proceedings of the 1st ACM conference on Electronic commerce*, 1999, pp. 129–139.

[129] K. Q. Nguyen and J. Traoré, "An online public auction protocol protecting bidder privacy," in *Australasian Conference on Information Security and Privacy*. Springer, 2000, pp. 427–442.

[130] K. Omote and A. Miyaji, "A practical english auction with one-time registration," in *Australasian Conference on Information Security and Privacy*. Springer, 2001, pp. 221–234.

[131] Y.-F. Chung, Y.-T. Chen, T.-L. Chen, and T.-S. Chen, "An agent-based english auction protocol using elliptic curve cryptosystem for mobile commerce," *Expert Systems with Applications*, vol. 38, no. 8, pp. 9900–9907, 2011.

[132] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1–34, 2019.

[133] Y. Mu and V. Varadharajan, "An internet anonymous auction scheme," in *International Conference on Information Security and Cryptology*. Springer, 2000, pp. 171–182.

[134] H. Huang, X.-Y. Li, Y.-e. Sun, H. Xu, and L. Huang, "Pps: Privacy-preserving strategyproof social-efficient spectrum auction mechanisms," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1393–1404, 2014.

[135] Z. Chen, L. Chen, L. Huang, and H. Zhong, "Towards secure spectrum auction: both bids and bidder locations matter: poster," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2016, pp. 361–362.

[136] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer *et al.*, "On scaling decentralized blockchains," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 106–125.

[137] S. D. Lerner, "Dagcoin: a cryptocurrency without blocks," 2015.

[138] C. N. Silla and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, no. 1-2, pp. 31–72, 2011.

[139] C. Groß, M. Schwed, S. Mueller, and O. Bringmann, "enerdag–towards a dlt-based local energy trading platform," in *2020 International Conference on Omni-layer Intelligent Systems (COINS)*. IEEE, 2020, pp. 1–8.

[140] E.-O. Blass and F. Kerschbaum, "Strain: A secure auction for blockchains," in *European Symposium on Research in Computer Security*. Springer, 2018, pp. 87–110.

[141] P. Lafourcade, M. Nopere, J. Picot, D. Pizzuti, and E. Roudeix, "Security analysis of auctionity: a blockchain based e-auction," in *International Symposium on Foundations & Practice of Security FPS 19*, 2019.

[142] M. K. Franklin and M. K. Reiter, "The design and implementation of a secure auction service," *IEEE Transactions on Software Engineering*, vol. 22, no. 5, pp. 302–312, 1996.

[143] M. Harkavy, J. D. Tygar, and H. Kikuchi, "Electronic auctions with private bids." in *USENIX Workshop on Electronic Commerce*, 1998.

[144] I. Onur, "Bidding behavior in dynamic auction settings: An empirical analysis of ebay," *Electronic Commerce Research and Applications*, vol. 9, no. 2, pp. 103–110, 2010.

[145] J. Trevathan and W. Read, "Cryptographic online auction schemes," in *Proceedings*

of IASK International Conference E-Activity and Leading Technologies, Madrid, Spain: IASK, 2008, pp. 193–203.

[146] A. Kulshrestha, A. Rampuria, M. Denton, and A. Sreenivas, "Cryptographically secure multiparty computation and distributed auctions using homomorphic encryption," *Cryptography*, vol. 1, no. 3, p. 25, 2017.

[147] Z. Guo, Y. Fu, and C. Cao, "Secure first-price sealed-bid auction scheme," *Eurasip Journal on information security*, vol. 2017, no. 1, p. 16, 2017.

[148] D. F. Alex Atallah, "peer-to-peer marketplace for crypto goods." https://opensea.io/, 2017, [Online].

[149] C. Braghin, S. Cimato, E. Damiani, and M. Baronchelli, "Designing smart-contract based auctions," in *International Conference on Security with Intelligent Computing and Big-data Services*. Springer, 2018, pp. 54–64.

[150] R. Tso, Z.-Y. Liu, and J.-H. Hsiao, "Distributed e-voting and e-bidding systems based on smart contract," *Electronics*, vol. 8, no. 4, p. 422, 2019.

[151] J. Martins, M. Parente, M. Amorim-Lopes, L. Amaral, G. Figueira, P. Rocha, and P. Amorim, "Fostering customer bargaining and e-procurement through a decentralised marketplace on the blockchain," *IEEE transactions on engineering management*, 2020.

[152] I. A. Omar, H. R. Hasan, R. Jayaraman, K. Salah, and M. Omar, "Implementing decentralized auctions using blockchain smart contracts," *Technological Forecasting and Social Change*, vol. 168, p. 120786, 2021.

[153] T. Nodehi, A. Zutshi, A. Grilo, and B. Rizvanovic, "Ebdf: The enterprise blockchain design framework and its application to an e-procurement ecosystem," *Computers & Industrial Engineering*, vol. 171, p. 108360, 2022.

[154] A. Sarfaraz, R. K. Chakrabortty, and D. L. Essam, "A tree structure-based improved blockchain framework for a secure online bidding system," *Computers & Security*, vol. 102, p. 102147, 2021.

[155] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for iot," in *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 2017, pp. 173–178.

[156] A. Sarfaraz, R. K. Chakrabortty, and D. L. Essam, "The implications of blockchain-coordinated information sharing within a supply chain: A simulation study," *Blockchain: Research and Applications*, p. 100110, 2022.

[157] D. Bechtsis, N. Tsolakis, E. Iakovou, and D. Vlachos, "Data-driven secure, resilient and sustainable supply chains: gaps, opportunities, and a new generalised data sharing and data monetisation framework," *International Journal of Production Research*, pp. 1–21, 2021.

[158] T. Ferdousi, D. Gruenbacher, and C. M. Scoglio, "A permissioned distributed ledger for the us beef cattle supply chain," *IEEE Access*, vol. 8, pp. 154 833–154 847, 2020.

[159] I. Yaqoob, K. Salah, R. Jayaraman, and Y. Al-Hammadi, "Blockchain for healthcare data management: Opportunities, challenges, and future recommendations," *Neural Computing and Applications*, pp. 1–16, 2021.

[160] J. Lai, F. Guo, W. Susilo, X. Huang, P. Jiang, and F. Zhang, "Data access control in cloud computing: Flexible and receiver extendable," *IEEE Transactions on Services Computing*, 2021.

[161] A. Ouaddah, H. Mousannif, A. Abou Elkalam, and A. A. Ouahman, "Access control in the internet of things: Big challenges and new opportunities," *Computer Networks*, vol. 112, pp. 237–262, 2017.

[162] U. Khalid, M. Asim, T. Baker, P. C. Hung, M. A. Tariq, and L. Rafferty, "A decentralized lightweight blockchain-based authentication mechanism for iot systems," *Cluster Computing*, pp. 1–21, 2020.

[163] G. Yu, X. Wang, K. Yu, W. Ni, J. A. Zhang, and R. P. Liu, "Survey: Sharding in blockchains," *IEEE Access*, vol. 8, pp. 14 155–14 181, 2020.

[164] A. Altarawneh, T. Herschberg, S. Medury, F. Kandah, and A. Skjellum, "Buterin's scalability trilemma viewed through a state-change-based classification for common consensus algorithms," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2020, pp. 0727–0736.

[165] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, and J. Voas, "Attribute-based access control," *Computer*, vol. 48, no. 2, pp. 85–88, 2015.

[166] E. Coyne and T. R. Weil, "Abac and rbac: scalable, flexible, and auditable access management," *IT professional*, vol. 15, no. 03, pp. 14–16, 2013.

[167] S. Bhatt, T. K. Pham, M. Gupta, J. Benson, J. Park, and R. Sandhu, "Attribute-based access control for aws internet of things and secure industries of the future," *IEEE Access*, vol. 9, pp. 107 200–107 223, 2021.

[168] H. Kim, D.-K. Kim, and A. Alaerjan, "Abac-based security model for dds," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[169] A. Majumder, S. Namasudra, and S. Nath, "Taxonomy and classification of access control models for cloud environments," in *Continued rise of the cloud*. Springer, 2014, pp. 23–53.

[170] X. Jin, R. Krishnan, and R. Sandhu, "A unified attribute-based access control model covering dac, mac and rbac," in *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer, 2012, pp. 41–55.

[171] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006.

[172] G. Kumar, R. Saha, M. K. Rai, R. Thomas, and T.-H. Kim, "Proof-of-work consensus approach in blockchain technology for cloud and fog computing using maximization-factorization statistics," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6835–6842, 2019.

[173] V. B. Pavel Khahulin Igor Barinov. (Sep. 2018) Poa network white paper. [Online]. Available: https://github.com/poanetwork/wiki/wiki/POA-Network-Whitepaper

[174] R. A. Mishra, A. Kalla, A. Braeken, and M. Liyanage, "Privacy protected blockchain based architecture and implementation for sharing of students' credentials," *Information Processing & Management*, vol. 58, no. 3, p. 102512, 2021.

[175] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 17–30.

[176] F. Hashim, K. Shuaib, and N. Zaki, "Sharding for scalable blockchain networks," *SN Computer Science*, vol. 4, no. 1, pp. 1–17, 2023.

[177] R. B. Chase, R. Shankar, and F. R. Jacobs, *Operations and Supply Chain Management, 15e (SIE)*. McGraw-Hill Education, 2018.

[178] K. Langfield-Smith and D. Smith, "Performance measures in supply chains," *Australian Accounting Review*, vol. 15, no. 35, pp. 39–51, 2005.

[179] B. S. Sahay, "Understanding trust in supply chain relationships," *Industrial Management & Data Systems*, vol. 103, no. 8, pp. 553–63, 2003.

[180] M. M. Aung and Y. S. Chang, "Traceability in a food supply chain: Safety and quality perspectives," *Food control*, vol. 39, pp. 172–184, 2014.

[181] K. Arshinder, A. Kanda, and S. Deshmukh, "A review on supply chain coordination: coordination mechanisms, managing uncertainty and research directions," in *Supply chain coordination under uncertainty*. Springer, 2011, pp. 39–82.

[182] S. Chopra and M. Sodhi, "Supply-chain breakdown," *MIT Sloan management review*, vol. 46, no. 1, pp. 53–61, 2004.

[183] S. Hu, S. Huang, J. Huang, and J. Su, "Blockchain and edge computing technology enabling organic agricultural supply chain: A framework solution to trust crisis," *Computers & Industrial Engineering*, vol. 153, p. 107079, 2021.

[184] F. Longo, L. Nicoletti, A. Padovano, G. d'Atri, and M. Forte, "Blockchain-enabled supply chain: An experimental study," *Computers & Industrial Engineering*, vol. 136, pp. 57–69, 2019.

[185] B.-C. Ha, Y.-K. Park, and S. Cho, "Suppliers' affective trust and trust in competency in buyers," *International Journal of Operations & Production Management*, 2011.

[186] J. W. Forrester, "Industrial dynamics," *Journal of the Operational Research Society*, vol. 48, no. 10, pp. 1037–1041, 1997.

[187] J. S. Goodwin and S. G. Franklin, "The beer distribution game: using simulation to teach systems thinking," *Journal of Management Development*, 1994.

[188] M. Coppini, C. Rossignoli, T. Rossi, and F. Strozzi, "Bullwhip effect and inventory oscillations analysis using the beer game model," *International journal of production Research*, vol. 48, no. 13, pp. 3943–3956, 2010.

[189] S. F. Wamba and M. M. Queiroz, "Blockchain in the operations and supply chain management: Benefits, challenges and future research opportunities," p. 102064, 2020.

[190] S. Nakamoto and A. Bitcoin, "A peer-to-peer electronic cash system," *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, vol. 4, 2008.

[191] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2019.

[192] M. Berneis, D. Bartsch, and H. Winkler, "Applications of blockchain technology in logistics and supply chain management—insights from a systematic literature review," *Logistics*, vol. 5, no. 3, p. 43, 2021.

[193] R. G. Brown, "The corda platform: An introduction," *Retrieved*, vol. 27, p. 2018, 2018.

[194] L. Qian, Z. Luo, Y. Du, and L. Guo, "Cloud computing: An overview," in *IEEE international conference on cloud computing*. Springer, 2009, pp. 626–631.

[195] S. Balamurugan, A. Ayyasamy, and K. S. Joseph, "Iot-blockchain driven traceability techniques for improved safety measures in food supply chain," *International Journal of Information Technology*, vol. 14, no. 2, pp. 1087–1098, 2022.

[196] H. Fatorachian and C. Smith, "Impact of cps on enhancing supply chain resilience, with a focus on solutions to pandemic challenges," in *Cyber-Physical Systems*. CRC Press, 2022, pp. 109–125.

[197] A. D. Ganesh and P. Kalpana, "Future of artificial intelligence and its influence on supply chain risk management–a systematic review," *Computers & Industrial Engineering*, p. 108206, 2022.

[198] M. Filali Rotbi, S. Motahhir, and A. El Ghzizal, "Blockchain-based cps and iot in the automotive supply chain," in *Advances in Blockchain Technology for Cyber Physical Systems*. Springer, 2022, pp. 155–176.

[199] W. Ran, Y. Wang, L. Yang, and S. Liu, "Coordination mechanism of supply chain considering the bullwhip effect under digital technologies," *Mathematical Problems in Engineering*, vol. 2020, 2020.

[200] P. Kumar, D. Sharma, and P. Pandey, "Coordination mechanisms for digital and sustainable textile supply chain," *International Journal of Productivity and Performance Management*, 2022.

[201] M. Bejlegaard, I.-M. Sarivan, and B. V. Waehrens, "The influence of digital technologies on supply chain coordination strategies," *Journal of Global Operations and Strategic Sourcing*, 2021.

[202] L. YiPeng, "The impact of" cloud computing"-based information sharing on supply chain," in *2011 Fifth International Conference on Management of e-Commerce and e-Government*. IEEE, 2011, pp. 173–175.

[203] F. Zhang and Z. Gong, "Supply chain inventory collaborative management and information sharing mechanism based on cloud computing and 5g internet of things," *Mathematical Problems in Engineering*, vol. 2021, 2021.

[204] C. V. B. Murthy, M. L. Shri, S. Kadry, and S. Lim, "Blockchain based cloud computing: Architecture and research challenges," *IEEE Access*, vol. 8, pp. 205 190–205 205, 2020.

[205] S. Xie, Z. Zheng, W. Chen, J. Wu, H.-N. Dai, and M. Imran, "Blockchain for cloud exchange: A survey," *Computers & Electrical Engineering*, vol. 81, p. 106526, 2020.

[206] J. Gong and N. J. Navimipour, "An in-depth and systematic literature review on the blockchain-based approaches for cloud computing," *Cluster Computing*, pp. 1–18, 2021.

[207] S. Moin, A. Karim, Z. Safdar, K. Safdar, E. Ahmed, and M. Imran, "Securing iots in distributed blockchain: Analysis, requirements and open issues," *Future Generation Computer Systems*, vol. 100, pp. 325–343, 2019.

[208] D. Shakhbulatov, J. Medina, Z. Dong, and R. Rojas-Cessa, "How blockchain enhances supply chain management: A survey," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 230–249, 2020.

[209] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *2017 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2017, pp. 2567–2572.

[210] A. Mohsin, A. Zaidan, B. Zaidan, O. Albahri, A. Albahri, M. Alsalem, and K. Mohammed, "Blockchain authentication of network applications: Taxonomy, classification, capabilities, open challenges, motivations, recommendations and future directions," *Computer Standards & Interfaces*, vol. 64, pp. 41–60, 2019.

[211] R. Azzi, R. K. Chamoun, and M. Sokhn, "The power of a blockchain-based supply chain," *Computers & industrial engineering*, vol. 135, pp. 582–592, 2019.

[212] H. L. Lee, K. C. So, and C. S. Tang, "The value of information sharing in a two-level supply chain," *Management science*, vol. 46, no. 5, pp. 626–643, 2000.

[213] K. Singha, J. Buddhakulsomsiri, and P. Parthanadee, "Mathematical model of inventory policy under limited storage space for continuous and periodic review policies with backlog and lost sales," *Mathematical Problems in Engineering*, vol. 2017, 2017.

[214] M. M. Fadıloğlu and Ö. Bulut, "A dynamic rationing policy for continuous-review

inventory systems," *European Journal of Operational Research*, vol. 202, no. 3, pp. 675–685, 2010.

[215] I. Christou, K. Skouri, and A. G. Lagodimos, "Fast evaluation of a periodic review inventory policy," *Computers & Industrial Engineering*, vol. 144, p. 106389, 2020.

[216] I. Rizkya, K. Syahputri, R. Sari, I. Siregar, E. Ginting *et al.*, "Comparison of periodic review policy and continuous review policy for the automotive industry inventory system," in *IOP Conference Series: Materials Science and Engineering*, vol. 288, no. 1. IOP Publishing, 2018, p. 012085.

[217] A. J. Ruiz-Torres and F. Mahmoodi, "Safety stock determination based on parametric lead time and demand information," *International Journal of Production Research*, vol. 48, no. 10, pp. 2841–2857, 2010.

[218] A. C. Radasanu *et al.*, "Inventory management, service level and safety stock," *Journal of Public Administration, Finance and Law*, no. 09, pp. 145–153, 2016.

[219] S. M. Disney, B. Ponte, and X. Wang, "Exploring the nonlinear dynamics of the lost-sales order-up-to policy," *International Journal of Production Research*, vol. 59, no. 19, pp. 5809–5830, 2021.

[220] A. Gunasekaran, C. Patel, and R. E. McGaughey, "A framework for supply chain performance measurement," *International journal of production economics*, vol. 87, no. 3, pp. 333–347, 2004.

[221] R. Metters, "Quantifying the bullwhip effect in supply chains," *Journal of operations management*, vol. 15, no. 2, pp. 89–100, 1997.

[222] S. M. Disney and D. R. Towill, "On the bullwhip and inventory variance produced by an ordering policy," *Omega*, vol. 31, no. 3, pp. 157–167, 2003.

[223] T. Moyaux, B. Chaib-draa, and S. D'Amours, "Information sharing as a coordination mechanism for reducing the bullwhip effect in a supply chain," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 396–409, 2007.

[224] B. Mahadevan, D. F. Pyke, and M. Fleischmann, "Periodic review, push inventory policies for remanufacturing," *European Journal of Operational Research*, vol. 151, no. 3, pp. 536–551, 2003.

[225] S. S. Ahiska and R. E. King, "Inventory optimization in a one product recoverable manufacturing system," *International Journal of Production Economics*, vol. 124, no. 1, pp. 11–19, 2010.

[226] J. Sousa, A. Bessani, and M. Vukolic, "A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform," in *2018 48th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*. IEEE, 2018, pp. 51–58.

[227] K. Toyoda, K. Machi, Y. Ohtake, and A. N. Zhang, "Function-level bottleneck analysis of private proof-of-authority ethereum blockchain," *IEEE Access*, vol. 8, pp. 141 611–141 621, 2020.

[228] M. Atzori, "Blockchain technology and decentralized governance: Is the state still necessary?" *Available at SSRN 2709713*, 2015.

[229] J. Abdella, Z. Tari, A. Anwar, A. Mahmood, and F. Han, "An architecture and performance evaluation of blockchain-based peer-to-peer energy trading," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 3364–3378, 2021.

[230] Y.-T. Yang, L.-D. Chou, C.-W. Tseng, F.-H. Tseng, and C.-C. Liu, "Blockchain-based traffic event validation and trust verification for vanets," *IEEE Access*, vol. 7, pp. 30 868–30 877, 2019.

[231] S. Ray and G. Biswas, "A certificate authority (ca)-based cryptographic solution for hipaa privacy/security regulations," *Journal of King Saud University-Computer and Information Sciences*, vol. 26, no. 2, pp. 170–180, 2014.

[232] J. A. Berkowsky and T. Hayajneh, "Security issues with certificate authorities," in *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*. IEEE, 2017, pp. 449–455.

[233] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X. 509 internet public key infrastructure online certificate status protocol-ocsp," 1999.

[234] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," *International journal of information security*, vol. 1, no. 1, pp. 36–63, 2001.

[235] S. Axsäter, "Continuous review policies for multi-level inventory systems with stochastic demand," *Handbooks in operations research and management science*, vol. 4, pp. 175–197, 1993.

[236] H. Ohta and T. Furutani, "Effect of customer order cancellation on supply chain inventory," *Journal of the Chinese Institute of Industrial Engineers*, vol. 21, no. 1, pp. 40–45, 2004.

[237] D. Wright and X. Yuan, "Mitigating the bullwhip effect by ordering policies and forecasting methods," *International Journal of Production Economics*, vol. 113, no. 2, pp. 587–597, 2008.

[238] A. Dolgui, D. Ivanov, and M. Rozhkov, "Does the ripple effect influence the bullwhip effect? an integrated analysis of structural and operational dynamics in the supply chain," *International Journal of Production Research*, vol. 58, no. 5, pp. 1285–1301, 2020.

[239] S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen, "Blockchain technology and its relationships to sustainable supply chain management," *International Journal of Production Research*, vol. 57, no. 7, pp. 2117–2135, 2019.

[240] K. Biswas, V. Muthukkumarasamy, and W. L. Tan, "Blockchain based wine supply chain traceability system," in *Future Technologies Conference (FTC) 2017*. The Science and Information Organization, 2017, pp. 56–62.

[241] X. Xu, D. Zhu, X. Yang, S. Wang, L. Qi, and W. Dou, "Concurrent practical byzantine fault tolerance for integration of blockchain and supply chain," *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 1, pp. 1–17, 2021.

[242] H. L. Lee, V. Padmanabhan, and S. Whang, "Comments on "information distortion in a supply chain: The bullwhip effect"," *Management science*, vol. 50, no. 12_supplement, pp. 1887–1893, 2004.

[243] S. Serdarasan, "A review of supply chain complexity drivers," *Computers & Industrial Engineering*, vol. 66, no. 3, pp. 533–540, 2013.

[244] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzzyva: Speculative byzantine fault tolerance," *ACM Transactions on Computer Systems (TOCS)*, vol. 27, no. 4, pp. 1–39, 2010.

[245] G. Yu, B. Wu, and X. Niu, "Improved blockchain consensus mechanism based on pbft algorithm," in *2020 2nd International Conference on Advances in Computer Technology, Information Science and Communications (CTISC)*. IEEE, 2020, pp. 14–21.

[246] Y. Xiao, N. Zhang, J. Li, W. Lou, and Y. T. Hou, "Distributed consensus protocols and algorithms," *Blockchain for Distributed Systems Security*, vol. 25, p. 40, 2019.

[247] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[248] M. U. Hassan, M. H. Rehmani, and J. Chen, "Privacy preservation in blockchain based iot systems: Integration issues, prospects, challenges, and future research directions," *Future Generation Computer Systems*, vol. 97, pp. 512–529, 2019.

[249] T. Laurence, *Introduction to Blockchain Technology: The many faces of blockchain technology in the 21st century*. Van Haren, 2019.

[250] E. K. Wang, R. Sun, C.-M. Chen, Z. Liang, S. Kumari, and M. K. Khan, "Proof of x-repute blockchain consensus protocol for iot systems," *Computers & Security*, vol. 95, p. 101871, 2020.

[251] S. Kamma, G. Kanatas, and S. Raymar, "Dutch auction versus fixed-price self-tender offers for common stock," *Journal of Financial Intermediation*, vol. 2, no. 3, pp. 277–307, 1992.

[252] R. Alvarez and M. Nojoumian, "Comprehensive survey on privacy-preserving protocols for sealed-bid auctions," *Computers & Security*, vol. 88, p. 101502, 2020.

[253] W.-S. Juang, H.-T. Liaw, P.-C. Lin, and C.-K. Lin, "The design of a secure and fair sealed-bid auction service," *Mathematical and computer modelling*, vol. 41, no. 8-9, pp. 973–985, 2005.

[254] M. Miyake, "On vickrey-type auction procedures," *Economics Letters*, vol. 51, no. 1, pp. 71–75, 1996.

[255] O. Birulin and S. Izmalkov, "On efficiency of the english auction," *Journal of Economic Theory*, vol. 146, no. 4, pp. 1398–1417, 2011.

[256] F. Peng, C. Chang, and M. Chen, "A study of influence of different auction mechanism to no-performing assets," *Sun Yat-Sen Management Review*, vol. 16, no. 3, pp. 401–428, 2008.

[257] S. Parsons, J. A. Rodriguez-Aguilar, and M. Klein, "Auctions and bidding: A guide for computer scientists," *ACM Computing Surveys (CSUR)*, vol. 43, no. 2, pp. 1–59, 2011.

[258] C. Avery, "Strategic jump bidding in english auctions," *The Review of Economic Studies*, vol. 65, no. 2, pp. 185–210, 1998.

[259] M. Kenney and J. Curry, *Beyond transaction costs: e-commerce and the power of the Internet dataspace*. Berkeley Roundtable on the International Economy (BRIE), E-conomy Project, 2000.

[260] J. Trevathan, W. Read, and H. Ghodosi, "Design issues for electronic auctions." in *ICETE*. Citeseer, 2005, pp. 340–347.

[261] K. Sakurai and S. Miyazaki, "An anonymous electronic bidding protocol based on a new convertible group signature scheme," in *Australasian Conference on Information Security and Privacy*. Springer, 2000, pp. 385–399.

[262] F. Brandt, "How to obtain full privacy in auctions," *International Journal of Information Security*, vol. 5, no. 4, pp. 201–216, 2006.

[263] X. Wang, Y. Ji, H. Zhou, Z. Liu, Y. Gu, and J. Li, "A privacy preserving truthful spectrum auction scheme using homomorphic encryption," in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[264] X. Hu, Z. Lin, A. B. Whinston, and H. Zhang, "Hope or hype: On the viability of escrow services as trusted third parties in online auction environments," *Information Systems Research*, vol. 15, no. 3, pp. 236–249, 2004.

[265] T. Srinath, S. Kella, and M. Jenamani, "A new secure protocol for multi-attribute multi-round e-reverse auction using online trusted third party," in *2011 Second International Conference on Emerging Applications of Information Technology*. IEEE, 2011, pp. 149–152.

[266] Y.-H. Chen, S.-H. Chen, and I.-C. Lin, "Blockchain based smart contract for bidding system," in *2018 IEEE International Conference on Applied System Invention (ICASI)*. IEEE, 2018, pp. 208–211.

[267] S. Rouhani and R. Deters, "Security, performance, and applications of smart contracts: A systematic survey," *IEEE Access*, vol. 7, pp. 50 759–50 779, 2019.

[268] S. van Engelenburg, M. Janssen, and B. Klievink, "Design of a software architecture supporting business-to-government information sharing to improve public safety and security," *Journal of Intelligent information systems*, pp. 1–24, 2019.

[269] H. Qusa, J. Tarazi, and V. Akre, "Secure e-auction system using blockchain: Uae case study," in *2020 Advances in Science and Engineering Technology International Conferences (ASET)*. IEEE, 2020, pp. 1–5.

[270] J. Sedlmeir, H. U. Buhl, G. Fridgen, and R. Keller, "The energy consumption of blockchain technology: beyond myth," *Business & Information Systems Engineering*, pp. 1–10, 2020.

[271] J. Xiong and Q. Wang, "Anonymous auction protocol based on time-released encryption atop consortium blockchain," *arXiv preprint arXiv:1903.03285*, 2019.

[272] P. Camacho, A. Hevia, M. Kiwi, and R. Opazo, "Strong accumulators from collision-resistant hashing," *International Journal of Information Security*, vol. 11, no. 5, pp. 349–363, 2012.

[273] T. Dryja, "Utreexo: A dynamic hash-based accumulator optimized for the bitcoin utxo set." *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 611, 2019.

[274] D. Boneh, B. Bünz, and B. Fisch, "Batching techniques for accumulators with applications to iops and stateless blockchains," in *Annual International Cryptology Conference*. Springer, 2019, pp. 561–586.

[275] J. Benaloh and M. De Mare, "One-way accumulators: A decentralized alternative to digital signatures," in *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 1993, pp. 274–285.

[276] E. Tremel, "Real-world performance of cryptographic accumulators," *Undergraduate Honors Thesis, Brown University*, 2013.

[277] J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials," in *Annual International Cryptology Conference*. Springer, 2002, pp. 61–76.

[278] T. Liao, M. Wang, and H. Tserng, "A framework of electronic tendering for government procurement: a lesson learned in taiwan," *Automation in construction*, vol. 11, no. 6, pp. 731–742, 2002.

[279] C. Fromknecht, D. Velicanu, and S. Yakoubov, "Certcoin: A namecoin based decentralized authentication system 6.857 class project," *Unpublished class project*, 2014.