Permissioned blockchain for data provenance in the supply chain with fine grained data authorisation

Huang, Elmo Xuyun

2021

Huang, E. X. (2021). Permissioned blockchain for data provenance in the supply chain with fine grained data authorisation. Master's thesis, Nanyang Technological University, Singapore. <https://hdl.handle.net/10356/148328>

<https://hdl.handle.net/10356/148328>

<https://doi.org/10.32657/10356/148328>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 21:27:56 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

**PERMISSIONED BLOCKCHAIN FOR DATA
PROVENANCE IN THE SUPPLY CHAIN
WITH FINE GRAINED DATA
AUTHORISATION**

**ELMO HUANG XUYUN
G1702889C**

**School of Computer Science and Engineering
2021**

**PERMISSIONED BLOCKCHAIN
FOR DATA PROVENANCE IN THE
SUPPLY CHAIN WITH FINE
GRAINED DATA AUTHORISATION**

ELMO HUANG XUYUN

**SCHOOL OF COMPUTER SCIENCE AND
ENGINEERING**

**A THESIS SUBMITTED TO NANYANG
TECHNOLOGICAL UNIVERSITY IN PARTIAL
FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING**

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

13-04-2021

Date

Elmo Huang Xuyun

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

17 April 2021

Date

Prof. Lam Kwok Yan

Authorship Attribution Statement

This thesis contains material from a published Journal paper where I was the co-first author.

Chapter 4 is published as:

Ma, Haiying & Huang, Elmo & Lam, Kwok-Yan. (2020). Blockchain-based mechanism for fine-grained authorization in data crowdsourcing. Future Generation Computer Systems. 106. 10.1016/j.future.2019.12.037.

The contribution of the paper is as follows:

1. I provide design the blockchain protocols and architecture. All the related experiments are also written and created by me. I have also written majority of the manuscript as well as the revision and response for the reviewers.
2. Ms Ma Haiying provided the calculations for CP-ABE protocol as well as some revision of the paper.
3. Prof. Lam Kwok Yan supervised the research work and provide suggestions for the areas of application and reviewed the manuscript for the final submission.

13-04-2021

Date

Elmo Huang Xuyun

Acknowledgement

I would like to express my greatest gratitude to my supervisor Prof. Lam Kwok Yan, who has guided me with endless patience, helped me complete my masters in engineering degree. He has also provided me with key opportunities for attachment to industrial projects. These projects offered invaluable experiences in the learning and application of Blockchain in the industries.

I would also like to express my gratitude to Ms Ma Haiying in her guidance on CP-ABE. Although her visit to NTU is short, she has taught me essential lessons in paper writing and novel crypto-primitives such as CP-ABE.

Lastly, I would like to express my appreciation to NTU's Spirit Lab and Scripts Lab members. They continuously provide me with new research ideas and assistance when times are tough.

Permissioned blockchain for data provenance in the supply chain with fine grained data authorisation

Elmo Huang Xuyun

Supervisor: Prof Lam Kwok Yan

Abstract

The sustainability of supply chains has become one of the critical focus of the 21st century. Consumers, investors and government regulators are increasingly scrutinising companies in terms of environmental impacts, fair trade, working conditions, and compliance with international rules and regulations. This issue is especially prevalent in the apparel industry. In recent years, some large apparel companies have outsourced production globally to sweatshops or factories with poor environmental standards, leading to boycotts by consumers and investors. This lack of transparency in modern supply chains is due to two significant technical challenges. With a globalised and integrated supply chain network, the flow of information is primarily controlled by large corporations in data silos. Consumers and regulators find it hard to track and trace the source of information as well as its accuracy. Consumers have to trust the manufacturers, while regulators have to conduct long and expensive investigations to determine these companies comply with regulations. However, malicious corporations often provide incomplete or inaccurate information. Secondly, with

cross border trade and third-party outsourcing, a large amount of information is lost in the complex supply chain network. In cases, small third-party contractors do not have the proper IT infrastructure to support the exchange of information, leading to missing information at the lower level of the supply chain. Thus, there is a need to create a multi-party data-sharing platform which encompasses the complexity of the supply chain and increases its transparency through data provenance. The platform should also be immutable and somewhat decentralised to without the need to trust the manufacturer.

To address this, Blockchain is a suitable candidate which provides a novel method of recording and information storage in Merkle tree hashes, such that the stored information is immutable. By storing supply chain data on the Blockchain, supply chain stakeholders can trace the provenance of the recorded data. Many blockchains attempt to increase transparency by using public Blockchain along with smart contracts. However, public blockchains are not scalable and cannot handle a large volume of transactions in real-time and at a low cost. Secondly, implementations of permissioned Blockchain for the supply chain focused primarily on the inventory management of large corporations and rarely considered the sustainability factor mentioned above. Moreover, these platforms also do not adequately address the privacy and security requirements of a supply chain blockchain platform, like the identities of the employees in the company or transactional data. Privacy laws regulate the privacy of the employees, while transaction data such as type of product, raw material sources directly affects the competitiveness of the company. Existing methods such as Public Key Cryptography Standards are not practical in a multi-tier decen-

tralised data-sharing platform and comes with considerable overhead. These platforms also failed to consider the problem of keys and access rights revocation when a new user or stakeholder joins or leave the blockchain platform.

There are two objectives of the thesis. The first objective is to design and implement a data-sharing platform where the market watchdogs(NGOs) and regulators are stakeholders in the Blockchain such that information can be independently verified and traced to the origin. The second objective is to develop and integrate an efficient and flexible privacy protection mechanism with low computational overheads into the blockchain platform to protect the privacy of stakeholders.

To address the first objective, we propose a new permissioned blockchain platform and protocol with data provenance capability to address the lack of transparency of the apparel supply chain. Regulators, as well as third party certifying agencies in the blockchain platform, are required to sign off key transactions and join into the proof of authority consensus protocol. Fraudulent information can thus be detected at the earliest stage, improving the data integrity and transparency of the data-sharing platform.

For the second objective, data owners in the Blockchain should be able to decide who, when and how the data will be accessed. The stakeholders in the Blockchain should not be able to collude by sharing multiple decryption keys. These requirements are not found in existing cryptographic infrastructure of existing blockchains. Thus in the thesis, we implemented a Ciphertext policy

Attribute-based encryption protocol (CP-ABE) in the Blockchain to allow data owners in the supply chain to perform fine grained authorisation based on the attributes of other stakeholders. CP-ABE allows data owners to set access policy based on parameters such as time, local and attributes or issue new decryption keys without the need to perform new encryption. In our protocol, the attributes can be revoked when the blockchain users leave the platform. The revocation prevents ex-blockchain users from gaining unauthorised access to data in the Blockchain. Hence, CP-ABE fulfils the requirements of an efficient, flexible, data protection mechanism which allows for fast and secure key issuing and revocation.

We have implemented the CP-ABE scheme on a pilot data-sharing platform and show that it is viable in a blockchain platform. We also implemented the proposed supply chain Blockchain platform with a modified CP-ABE. The implementation result proved that our protocol design is feasible and improves the transparency of the supply chain. Lastly, implementation also shows that traditional CP-ABE scheme can be modified to be decentralised in a Blockchain.

List of Figures

1.1	Typical manufacturing flow of apparel supply chain	6
2.1	Blockchain records	9
2.2	The two main types of P2P connections	14
4.1	The system model of BC-FGA-DCrowd	32
4.2	BC-FGA-DCrowd framework	35
4.3	Cost and time requirements of the smart contracts	56
5.1	Overview of the architecture layers of the cotton supply chain blockchain protocol	61
5.2	Overview of stakeholders' attribute of the cotton supply chain blockchain protocol	67
5.3	Plot of the linear relationship between the size of the database with the number of blocks	70
5.4	The time required to create transactions scale linearly with the number of transactions	70

List of Tables

2.1	Comparison of public and permissioned blockchain	10
4.1	The costs of smart contracts in BC-FGA-DCrowd	54
A.1	Table of information access rights	78

Contents

Statement of Originality	i
Supervisor Declaration Statement	ii
Authorship Attribution Statement	iii
Acknowledgement	iv
1 Introduction	1
1.1 Supply chain consortium platform	2
1.2 Application scenario	3
1.3 Motivation and objectives	7
1.4 Organisation of the Master Thesis	7
2 Preliminaries	8
2.1 Blockchain	8
2.2 Blockchain system Architecture and properties	8
2.2.1 Public Blockchain	9
2.2.2 Permissioned Blockchain	11
2.2.3 Blockchain architecture	12
2.2.4 Network and Nodes	13
2.2.5 Consensus protocol	16
2.3 CP-ABE	22
3 Literature reviews	24
3.1 Supply chain and logistics data sharing platforms and manage- ment software	24
3.2 Existing implementation of supply chain and logistics blockchain	26
3.2.1 Permissioned Blockchain for supply chains	27
3.2.2 Use cases and case studies	28

4	CP-ABE For privacy protection in data crowdsourcing Blockchain	31
4.1	BC-FGA-DCrowd system model	31
4.2	BC-FGA-DCrowd system framework	34
4.2.1	Application layer	34
4.2.2	Blockchain layer	35
4.2.3	Data storage layer	36
4.2.4	The process of BC-FGA-DCrowd	36
4.2.5	System security threat model and security assumption	38
4.3	Blockchain-based mechanism for fine-grained authorisation in data crowdsourcing	41
4.3.1	Smart contracts for data crowdsourcing	42
4.4	Concrete BC-FGA-DCrowd scheme	45
4.4.1	User Register	45
4.4.2	Encrypt Data	46
4.4.3	Data Sales	47
4.4.4	Fine-grained authorisation	49
4.4.5	Decrypt	49
4.4.6	Reward Assignment	50
4.4.7	Confirm Transaction	51
4.5	Security Analysis	51
4.6	Performance	54
4.7	Conclusion	57
5	CP-ABE in a permissioned blockchain	59
5.1	Implementation	59
5.2	System model	59
5.2.1	Users Archetypes	59
5.2.2	Information and documentation flow and access control	60
5.3	Blockchain architecture	61
5.3.1	Middle application server	63
5.3.2	Concrete Supply chain Framework	64
5.4	CP-ABE Implementation	66
5.4.1	Attributes and access policy	66
5.4.2	Roles definition	66
5.4.3	Integration of CP-ABE with blockchain	68
5.5	Experimental results	69
5.5.1	Database size	69
5.5.2	Speed of transactions	69
5.6	Security analysis	71
5.6.1	Security threats	71

6	Planning and Future research	73
6.1	Conclusion	73
6.2	Future works	74
A	Appendix	75

Chapter 1

Introduction

Fairtrade, environmental sustainability and product provenance have become areas of interest in global trade and supply chains. These interests stem from the rising pressure from consumers to reduce inequality, environmental degradation and counterfeiting that is prevalent in modern societies. However, improvements have been slow due to the lack of transparency in existing supply chains. Large global corporations like Apple Inc, Nike, Nestle, GAP, often have massive control over the entire supply chain. This includes the management of information regarding sources of materials, human resources and manufacturing contracts. Consumers and the public often have to rely on whistleblowers to expose unethical practices like sweatshops or unfair trading contracts [1] [2]. In 2018, reports by Global Labour Justice revealed that GAP and H&M have repeated outsourced manufacturing services from sweatshops [3]. As multinational corporations' supply chains become increasingly integrated, companies also are facing increased pressure from ethical consumers and investors to increase the transparency of supply chains. Ethical consumers and investors demand reliable and trustworthy information to make correct purchasing and investing decisions. Small stakeholders at the bottom of the supply chain would also require higher transparency to make better business decisions. In most cases, the stakeholders and consumers have to trust information released by the companies or undercover journalists. Traditional supply chain management platforms are often ill-equipped to manage information sharing among stakeholders, governmental regulators and the public [4].

The development of distributed ledger technologies(DLT) such as blockchains, lead to many companies attempting to integrate DLTs in their supply chain to improve transparency and efficiency. The increase in transparency is due to traceable and immutable transaction information available to all the stakehold-

ers in the supply chain. Specifically, each stakeholder in the platform has a copy of the ledger, and no single stakeholder has control over the entire platform. Stakeholder does not need to rely on any trusted party for book-keeping. Regulators and certifying organisations such as FairTrade can also be included as participants or stakeholders in the platform to improve transparency for the public.

Lately, Blockchain has become the most common DLT platform implemented. This rise in popularity is due to its ability to engage a large number of participants in a trust-less platform as well as being a fully decentralised payment system. Although there are a large number of blockchains that are being tested out by various companies, the actual adoption of blockchains into the supply chains have been slow for several reasons. Firstly, several of these blockchains do not fulfil the security and data privacy requirements of stakeholders. In the supply chain, the secrecy to information such as prices, raw materials and existing stocks affects the competitiveness of the company. Any security breach or data leakage is very costly. Adopting non-certified or proven systems post a considerable risk to large companies, reducing the willingness of companies to adopt blockchain technologies. Secondly, existing blockchains have a low rate of transaction per second as compared to existing SCM platforms. In large supply chains with high trade volume, some blockchain platforms can also be easily overwhelmed. This is due to incorrect and inefficient consensus protocol as well as wrong network protocols which leads to a low rate of transaction per second. Lastly, most existing businesses are still based on traditional business processes which assume trust between the stakeholders. It would be challenging to disrupt the entire supply chain using blockchain technologies.

1.1 Supply chain consortium platform

To facilitate the adoption of blockchain platform into the existing supply chain, we build a Proof of concept blockchain framework which solve the problems as mentioned earlier. The new permissioned blockchain architecture takes into the existing business flow of the supply chain, as well as the current infrastructure and key characteristics of each stakeholder. The first key consideration of the framework is the cost of implementation and training required for each of the stakeholders in the supply chain. The Blockchain has to be accessible to everyone in the supply chain, including the low-income farmers and producers in the supply chain. Thus the blockchain platform uses web RestAPIs that is usable low powered devices like a mobile phone. Consequently, there is a

lower cost of installation and a lower knowledge barrier as not much technical expertise is required. Furthermore, the blockchain nodes can be hosted by a cloud server where the cost can be shared among a large group of farmers and producers with a tiny margin.

The next area of consideration is the performance of the Blockchain. New transaction structures have to be created specifically for the type of supply chain that the platform is required to service. Cloud storage is required to store other essential information, such as invoices and pictorial evidence. The hashes and URI of the data are recorded on the Blockchain to keep the Blockchain lean and efficient. In our application setting, we based our implementation based on the international cotton and apparel trade.

The last area of consideration is the security and privacy protection of the data in the platform. The Blockchain platform implemented Distributed Ciphertext-Policy Attribute-Based Encryption to allow users to fine-tune authorisation and user control for encryption of key data. Data owners can encrypt their data and set the access policy based on the identity and user group of the person accessing the data. This allows for more efficient and safer data sharing, ensuring the privacy protection of the user. Besides, the network protocol is replaced from a standard peer-to-peer network protocol to publish and subscribe broadcasting model to increase the rate of transactions. Lastly, a Proof of Authority consensus protocol to improve privacy protection and performance further.

1.2 Application scenario

Currently, the world produces about 23 million tonnes of cotton in over 90 countries and 2 million tonnes of wool. Suppliers and manufacturers often exploit the lack of transparency in the apparel supply chain and pass off cheap products as 'fair-trade' or 'organic'. Most of the raw materials are traded through commodity exchanges, and undergoes multiple exchanges of hands, from farmers to mills, merchants and retailer. Thus, it is difficult for retailers and consumers to know the origin of the materials they are purchasing. These bring about two primary concerns to both merchants and consumers. Firstly, premium brands often promote their cotton or wool apparel as a high and luxurious quality products with a specific origin and limited production. Egyptian cotton and high-quality Merino wool are some of the most commonly counterfeited raw materials. For example, accounting for only 1% of the world's production, 90% of the premium Egyptian cotton products cannot be traced to

Egypt [5]. Secondly, concerns with ethical farming and production make the tracing of the origin of cotton and wool is especially essential. While many merchants pledge to exclude unethical farmed materials in the exchanges, non-ethical farm cotton and wool are still found widely on the market. In 2018 it was found that, while major cotton importers vow to boycott cotton produced by State-sponsored forced child labour in Uzbekistan, a large amount of cotton in the market can be traced to Uzbekistan [6]. This can be attributed to the complexity of the cotton trade and the lack of transparency. As such, an effective and efficient supply chain management solution is required to provide accurate trace and trace mechanisms. Blockchain, in many ways, solves many of the challenges in an existing information integration system for Supply chain management.

Firstly, Blockchain provides automated record tracking and verification. Secondly, the data structure of Blockchain makes it less corruptible compared to a traditional centralised database. However, the problem of information privacy still a significant problem in many adoptions of supply chain blockchains. A native blockchain does not have measures to protect the privacy of its content from non-authorised users. The inventory data of each company is supposed to be a trade secret. Thus, the new integrated blockchain Inventory management platform must have privacy control measures.

Complexity in the Apparel supply chain

The apparel production supply chain can be narrowed down to several essential archetypal entities: Farmer, Yarnspinner, Fabric Mill, Product manufacturer, retailer and finally, consumers. The flow of the apparel supply chain can be illustrated as in Fig1.1 with the cotton trade:

- The cotton farmer plants and harvest cotton from cotton plantations and sell the raw cotton directly to yarn spinners or intermediary traders. With the small output and low technological expertise, it is difficult for smaller farmers to gain market access. Intermediary traders often purchase cotton from small scale farmers and trade cotton on the international market at a much higher profit.
- Yarn spinners purchase raw cotton and convert them into yarns of processed cotton. These yarns can be sold to fabric mills or retail outlets for commercial usage
- In the next stage, fabric mills weave cotton yarns from different yarn spinners into bolts of fabric.

- The Manufacturer produce garments out of the fabrics. It is common for a manufacturer to use fabrics from a wide variety of sources.
- Retailers purchase garments from the manufacturer and sell to the end consumer. It Retailers are typically located across different countries and regions.
- Consumers are the endpoint of the supply chain. They purchase the end product either through physical stores or online stores. Typically, they are provided with information regarding the country of manufacturing as well as some promotional information.

The stakeholders in the cotton supply chain are located across a large number of locations forming a huge supply chain network that spans across the world. Also, different stakeholders have different levels of technical expertise and market access. It is commonly observed that the lower the stakeholder on the supply chain, the lower the technological expertise and market access. In general, the complexity and lack of transparency in the cotton supply chain can be narrowed down into the following reasons:

1. Yarn spinners produce yarns from a large collection of cotton farmers located across different regions. This results in each cotton yarn consisting of cotton fibres from a large number of different farmers. A professional lab is required to conduct DNA testing to trace the source of cotton fibres.
2. Large amount of cotton is being traded through different intermediary traders before being purchased by any yarn spinner or mills. This is due to small farmers not able to sell their product on the broader market by themselves.
3. Fabric mills often combined yarns from different yarn producers across different regions.
4. During the production of fabrics, the yarns and fabrics undergo different treatment and manufacturing processes such as dyeing, laundering, embellishment. Sometimes, these processes can also be subcontracted to many different smaller companies.
5. Retailers does often do not have full information regarding the product they are selling. Often retailer no having any relationship to any stakeholder in the supply chain outside of the garment manufacturer.

Figure 1.1: Typical manufacturing flow of apparel supply chain

1.3 Motivation and objectives

The thesis aims to demonstrate how blockchain technology can improve fair trade and bring about better transparency to the supply chain. The first motivation behind this goal is to allow farmers to achieve better living conditions by improving both prices of the cotton they produce as well as the natural environment they are working. Large producers will have to offer workers better working conditions and regulated by NGOs. The second motivation factor is to create a platform which can reduce counterfeits of more premium cotton products. Producers should not be able to pass off normal cotton products as organic products or sell a cheaper variety of cotton as premium products.

The following objectives are set to achieve the aim of the thesis:

1. To identify and isolate the reasons for complexity in supply chains.
2. To identify the limitations of existing blockchains the supply chains.
3. To create/improve existing solutions to the limitations identified in (2).
4. To design and implement a blockchain protocol that increases the transparency of the supply chain.

1.4 Organisation of the Master Thesis

The thesis is organised as follows:

1. Chapter 1: An introduction of the background, motivation, objective and setting of the thesis.
2. Chapter 2: Prerequisite knowledge of blockchain, CP-ABE and supply chain.
3. Chapter 3: A literature review of similar supply chain blockchains, user requirements and security risk and vulnerability of blockchains.
4. Chapter 4: Design and implementation of BC-FGA-DCrowd system.
5. Chapter 5: CP-ABE in a permissioned blockchain.
6. Chapter 6: Conclusion and future directions.

Chapter 2

Preliminaries

2.1 Blockchain

Blockchain is a list of digital records known as blocks linked using cryptography Fig 2.1. The most common cryptographic method used for the linkage is a hash chain based on a Merkle tree structure. In this structure, the hash of the current block contains the hash of the previous block. The content and records of the blocks can vary depending on the use case scenario. In the initial design of Bitcoin, the records are transactions of digital transfers. Namely, a party A transferring X amount of currency to another party Y. The transactions are then appended to a block containing the hash of the previous block, timestamp and a nonce.

As the technology matures, new features are added. Mainly, the Elliptic Curve Digital Signature Algorithm for account creation and transaction signing, decentralisation and consensus protocols to manage all the transactions and synchronisation between the nodes.

2.2 Blockchain system Architecture and properties

The basic Blockchain system consists of several main components, database, nodes, network, applications and a user interface. The implemented configuration of the platform will vary depending on the type of blockchain that is required. In general, there are two types of blockchain: Public Blockchain and Permissioned Blockchain. Public blockchains do not place any restriction on the identity of users that are allowed to participate. On the other spectrum, permissioned blockchain requires the user's identity to be verified and authen-

Figure 2.1: Blockchain records

ticated. Permissioned blockchain can be further sub-divided into Consortium based blockchain and private blockchain. In Consortium blockchains, the platform is owned and managed by a group of organisations while blockchains owned by a single organisation are defined as private blockchains. Lately, there are also prototype platforms which aim to combine the performance of permissioned blockchain with audibility and decentralisation of public blockchains. The main properties of the different types of blockchain are summarised in Table 2.1.

2.2.1 Public Blockchain

Public blockchains are fully decentralised platform, solely maintained by a large group of miners/nodes. These miners free to join, read/write, participate and leave the blockchain network with little to no restrictions. This decentralisation also restricts public blockchains to operate on P2P networks. Without any centralised authority or company to maintain the platform, incentives also have to be provided to nodes and miners to perform computation task like the validation of transactions, ledger storage, transaction broadcasting and other forms of computations. These incentives take the form of cryptocurrency to-

Table 2.1: Comparison of public and permissioned blockchain

Comparison of public and permissioned blockchain		
Properties	Public	Permission
Efficiency	Low	High
Audibility and Immutability	High	Low
Centralization	No	Yes/partial
Consensus	Permissionless miners	Permissioned
Read permissions	Public	Access controlled
Number of nodes	Large	small

kens which can be used for payment for real-life products and services or sold on coin exchanges. Bitcoin [7] was the first blockchain and public blockchain with an initial primary objective of developing a decentralised monetary system. With the success of Bitcoin, many different variations of public blockchain were developed and launched. As of the writing of this thesis, there is 5500 version of blockchain cryptocurrency accepted on the coin exchange, with Bitcoin amounting to 66% of the total market value.

In general, public blockchains have three main critical architectural features that are distinct from permissioned blockchain. These features are specialised consensus protocols, cryptographic protocols and a network with a large number of nodes and miners. Finally, any platform changes or update must be decided based on votes by the blockchain nodes.

For public blockchains, consensus protocols are required to generate, distribute, and allocate rewards fairly to miners' network. Proof of Work and Proof Stake are the two most popular types of consensus protocol for public blockchains with a large number of public blockchains have consensus protocol based on the different implementation of Proof of Work or Proof of Stake or a combination of both.

Public blockchains also implemented Elliptic-curve Digital Signature Algorithm (ECDSA) to create keys and sign transactions. In the ECDSA blockchain protocol, 256bit private key is first randomly generated. The private key is used to sign transactions or generate public keys. The public key acts as the user's address and is used to make transactions and receive payment.

Lastly, public blockchains require a large number of nodes to operate correctly. Small public blockchains with a small number of nodes are prone to Sybil attacks. It is assumed that it is difficult for a single party to own a large enough number of nodes to affect the blockchain's operation. If any party controls more than 51% of the nodes, the blockchain platform is compromised, and the ledger is no longer immutable and is prone to fraudulent transactions. While POW can increase its resistance to Sybil attacks, there lies a trade-off in transaction speed and energy cost. These key features and requirements make public blockchain not suitable for large scale implementation in an industrial or supply chain setting compared to permissioned blockchain.

2.2.2 Permissioned Blockchain

Permissioned blockchain is developed to adopt blockchain technology into existing business infrastructure and industrial requirements. Hyperledger was one of the first platform offering blockchain solution to businesses in the form of permissioned blockchain. Permissioned blockchains offer manufacturer and supply chains higher transaction rates, improved privacy and security protection, customizability and reliability. In return, permissioned blockchains are less decentralised and scalable. However, in most application settings, these platforms do not need to be fully decentralised or run a large number of nodes. Instead, the entire blockchain platform would be preferred to be owned and maintained by a small number of authorised nodes. In some cases, the blockchain is used by a single entity purely as enterprise solutions. As more companies become increasingly interested in blockchain enterprise solutions, it is becoming common for companies to collaborate and join develop consortium blockchains. Consortium blockchains are permissioned blockchains with a controlled user group across different organisations. This allows for like-minded companies to leverage shared information to improve workflows, accountability, and transparency without the compromises of traditional blockchains. With these objectives, permissioned blockchains have several architectural considerations and designs that are different from the public blockchain.

Firstly, the consensus protocol used for permissioned blockchains removes any features that generate and distribute cryptocurrency to improve the platform's performance. Since the consensus protocol's main focus in permissioned blockchain is to distribute and synchronise information across every node as fast as possible, transactions are directly broadcast to the entire network. This simplifies the validation and block generation process, allowing for as little latency as possible. To further reduce any latency, most permissioned blockchain

utilised network messaging services such as Publish-subscribe or just basic TCP/IP stack. In such cases, consensus protocols like PBFT or PoA most commonly implemented. In special occasions, permissioned blockchains with a larger number of nodes and whose sole purpose is State replication will implement a P2P network with complimentary consensus protocol such a RAFT or Paxos.

Any nodes that wish to join a permissioned blockchain have to be authenticated. In blockchain with a high level of autonomy, the nodes are often split into two class of authorisation. In most cases, the permissioned blockchain will contain a group of founding/governing nodes with the highest authorisation level during initialisation. These nodes will authorise new secondary nodes to join the blockchain. These secondary nodes will make up the second group of authorised nodes in the blockchain and have a lower level of user access rights and authority. Secondary nodes are allowed to read and write transactions into the blockchain, but not allowed to authorise new nodes to join the network. The promotion of secondary nodes to the governing node's role requires consensus of the pool of governing nodes. Changes in the blockchain would also be decided based on the voting session results by the governing nodes. In the majority of industrial applications, it is preferred to have a more centralised approach, where there is a single or small group of decision-making entities/stakeholder. Stakeholders of the business framework will own the nodes. The nodes in the blockchains are also connected through secure communication channels, with the identity of the nodes being authenticated by Secure Sockets Layer certificate(SSL). SSL prevents non-authorised personnel from accessing the transaction data on the permissioned blockchain, protecting the stakeholders' privacy. Protecting stakeholders and users' privacy is crucial in heavily regulated industries such as the finance industry and supply chain. Some of the successful permissioned blockchain platforms include Hyperledger for supply chain purposes and Quorum, Corda for the financial industry.

2.2.3 Blockchain architecture

Transactions

Transactions are the basic build block of any distributed ledger platform. The security of and streamlining of transactions plays a major role in the design of the platforms. In a blockchain, transactions are data structures that record an asset's transfer from one party to another. The bookkeeping content of the transaction can differ based on the blockchain setting with more sophis-

ticated blockchain catering to multiple types of asset transfers. Catering to multiple types of assets can be resource heavy and inefficient. Thus, blockchain designers decide on what essential information should the transaction record. All transactions are also required to be signed by the sender's signature. The authorisation improves the security of the transactions.

All transactions are divided into two categories: Verified and non-verified/open transactions. Verified transactions are validated and added to the blocks by the consensus protocols. Non-verified transactions are transactions that are not verified or queued in the memory pool and waiting list of the blockchain nodes. Each node will broadcast non-verified transactions to other nodes in the network for verification and logic checks. Once verified, these transactions will move on to the verified transaction list. There are also exceptions where transactions are merely used for simple data storage such as URIs or statements. In such cases, the transactions do not need to be verified.

Blocks

A genesis block is the first block in the blockchain. The genesis block contains a secret string as well as the very first hash. This block will not contain any transaction data, and the hash of this block will be stored as the previous hash in the second block. The block hash also acts as a pointer for the blockchain to locate the previous block. New blocks are generated based on a fixed interval or a fixed number of transactions. For the former, after a set interval, the list of verified transactions will be added to the new block. In the latter, after the list of verified transactions reaches a fixed number n , the process to generate a new block will be initiated. The block will then undergo the consensus protocol to decide if the block is valid before being added to the chain.

2.2.4 Network and Nodes

Blockchain, as a distributed ledger, consists of nodes connected through network communication protocols. The network protocol is determined by the type of consensus protocols implemented. It also directly affects the performance and robustness of the blockchain. Formally, the blockchain network follows the CAP theorem, which states that a distributed network cannot have more than two of the following properties [8]:

- Consistency: Every node receive the latest read and write request
- Availability: Every request always receive a response

Figure 2.2: The two main types of P2P connections

- Partition tolerance: The blockchain continues to operate regardless of the number of dropped messages between the nodes.

Thus blockchain platform designers often sacrifice one of the fore-mentioned properties to the platform its application scenario. In public blockchains, blockchain platform creators will trade consistency for availability and partition tolerance through a peer-to-peer network protocol. Private/permissioned blockchain platforms that do not require availability often utilised a publish-subscribe network messaging protocol, while platforms that do not require availability will use a Broadcast/Multicast network messaging protocol instead.

Peer-to-Peer

Peer-to-peer or P2P networks consist of nodes with equal power and perform similar tasks. In the case of a public blockchain, an unstructured P2P architecture is more commonly used.

In a structure P2P network, the nodes maintain a Distributed Hash Table(DHT). The DHT maintains the joining and leaving of the distributed network and assigns specific tasks to each network's peer nodes. An unstructured P2P network allows peers to join and leave the network freely. When new nodes join the network, existing links are copied from the nodes' peer list that the new nodes are connected. It will then start to find and form new connections by itself. Each node also downloads a copy of the blockchain data/records. However, only nodes with the latest copy of the blockchain records can take part in any transactions. This functionality makes unstructured P2P blockchains extremely suited for its primary design goal of generating cryptocurrencies. Users

can join or exit the network without going through any regulatory bodies, and the consensus protocol purely decides the generation of new tokens. A large peer to peer platform is also resistant to Denial of Service attacks. Even if attackers attack several nodes, the partition tolerance property of P2P architectures will allow P2P blockchains to perform its functions still. To overwhelm the P2p blockchain, the attacker would have to attack or control the majority of all nodes(51%), which is unlikely in major platforms such as Bitcoin.

Given the advantage of P2P networks, there are two major concerns. Firstly, P2P systems are extremely inefficient and slow in achieving consensus for the transactions. In large P2P blockchains take a long time for an event(transaction) to be distributed to every node in the platform. Studies by Pappalardo, G., Di Matteo, T., Caldarelli, G. et al. [9] found that on average, 42% of the transactions are still not recorded on the blockchain after 1 hour from creation, and 20% of the transactions are not recorded on the blockchain even after 30 days. Due to the size based weighted probability of transactions, approximately 7% of the transactions will be 'forgotten' by the blockchain. The authors attributed this problem partially to the peers in the network. Since propagating messages to a large number of nodes consumes more energy, miners in the network will limit the number of connected peers. Adding on to the inherent storage latency that exists in Bitcoin, the number of transactions per second is very low. Secondly, it is difficult to protect the privacy of the transactions. All the transactions in the blockchain are publicly available. There is little to no restriction for peers to join and leave the network. Peers are also free to distribute information and transactions in the blockchain. Even though there are native privacy protection mechanisms in public blockchains, the identity of the user can still be recovered. In the earlier version of Bitcoin, up to 40% of the user's identity can be recovered [10] in a paper by Androulaki E. et al. Even though later versions of a public blockchain like Bitcoin and Ethereum reduced these percentage by adopting new protection measure. This is still a major privacy problem.

For public blockchains, the advantages of a P2P protocol outweigh the disadvantages. However, this form of network architecture is not suitable for industrial application where speed and privacy are key considerations. Thus alternative network architectures are implemented in many permissioned and private blockchains instead.

Publish/Subscribe

In newer blockchains, namely permissioned blockchains such as Corda and some versions of Hyperledger, publish and subscribe messaging network protocol(Pub/sub) has become the new standard network protocol. In pub/sub, the sender(publisher) sends messages and information to a list of receivers(subscriber). The identities of the subscribers in the list are not revealed to the publisher. The anonymity of the subscribers protects the privacy of the subscribers.

In general, compared to P2P, the main advantages of using pub/sub in a blockchain platform are transmission speed and ease of access and privacy control while keeping communication simple. In a pub/sub blockchain, the transaction is sent to receiving peers directly from the sender for consensus. Thus the time required for anyone in the blockchain to receive the transactions is mainly dependant on the latency of the network and the type of communication protocol used(UDP vs TCP). The difference in speed due to communication protocol is minute in a real-life scenario. The new communication protocol makes pub/sub more compatible with permissioned and private blockchain that requires secure and fast transactions as compared to a P2P network. Lastly, it enables the nodes to host multiple blockchains simultaneously, which allows the node owners to have more flexibility.

2.2.5 Consensus protocol

The consensus protocol is the set of rules which governs the nodes and servers in a distributed system. Even before the existence of blockchain, there are numerous consensus protocols such as Practical byzantine fault tolerance(PBFT) [11]. In any distributed systems, all the different databases, servers and clients need to be in-sync and agree on the same system state .i.e., agree on the same inputs and outputs to a given function at any point in time. A distributed system is prone to byzantine faults due to missing information for corrupted due to network conditions and component failures. A non-distributed version of blockchain will not have any consensus protocol. In a distributed ledger, the current transactions' validity relies on the net balance of the previous transactions. A set of rules have to be created for the nodes to generate a common output. Any transactions and block generation will also have to follow these rules. Simply, the primary goal of a blockchain consensus is to synchronise transaction data distributed among the nodes. The verification of the validity of transactions and blocks is a key part of that synchronisation process. In public blockchains, the consensus protocol has an additional role replacing the

centralised authority and allocate crypto-currency rewards to the miners. These miners operate and maintain the nodes which make up the entire blockchain network. The consensus protocol will then contain rules created by the creator of the blockchain to allocate compensation to these miners. Currently, the most popular consensus protocols are Proof of Work, Proof of Stake for public blockchains and Proof of authority, PBFT and RAFT for permissioned blockchains.

Proof of work

The first blockchain, Bitcoin, utilised a consensus known as Proof of work *PoW*. The concept of PoW is based on an existing hashcash algorithm. [7] In the hash cash algorithm, a secret string S is hashed to a length with a fixed bit size by a function F . The function F , is designed such that $H = F(S)$ is easy to compute while the reverse function is computationally difficult. With PoW, the subsequent hash $H = F(S)_n$ is generated by $H_n = F(H_{n-1} + B_{n-1})$ where B_{n-1} is the data in the previous block.

Each block will also contain a randomly generated string (nonce). The platform selects the difficulty of mining based on the number of 0s required in the hash. The miners will have to validate the list of transactions it received before solving for the nonce. The first node which found the nonce which generated a hash with the required number of 0 bits gets to published or create the block and obtain the mining rewards. The rest of the miners will then extend the blockchain's length by solving for the next block. The probability for each miner to solve the computationally hard problem is directly proportional to their computational power. PoW provides Bitcoin with a method platform maintenance without any centralised authority. The reward given to the miners acts as an incentive for the miners to continue providing computational services to validate transactions as well as for new miner to join the network. In theory, PoW protects the platform by allowing up to 51% malicious or faulty nodes. The main deterrence lies in the huge computational cost involved to attack the blockchain.

PoW is computationally resource-intensive and wasteful. In 2019, Bitcoin accounted for 0.21% of the world consumption of electricity, equivalent to the consumption by the whole of Switzerland [12]. For private and permissioned blockchains, such as energy consumption and hardware hungry protocol are undesirable. The rate of transaction per second is also too slow for much real-life practical or industrial purpose. In 2019, the average transaction per second for

Bitcoin was 7 [13], which is several orders of multitudes smaller than platforms like Visa or Paypal. A high-difficulty setting results in low transactions per second, while a low difficulty setting is not effective in DOS attacks. Lastly, PoW systems are also vulnerable to forking attacks. An attacker can create their secret version of blockchain and broadcast the chain when it becomes longer than the original honest chain. Since the longest chain will be deemed as the correct chain, the attacker's chain will replace the honest chain. New improvements and hybrids are made in POW to improve both performance and new attacks, but the outcome is still far from desirable.

Practical byzantine fault tolerance

The goal of PBFT [11] is to provide distributed systems with a reliable method of state replication that is byzantine fault resistant. To tolerant f faulty nodes, a PBFT system require only $n = 3f + 1$ nodes. Since its introduction in 1999, a series of new Byzantine fault-tolerant algorithms have been released and adopted. In the PBFT protocol, each node in the protocol is assigned one of the two types of roles, primary/leader nodes and secondary/backup nodes. Each epoch in the original PBFT protocol can be divided into five stages: Request, pre-prepare, prepare, commit, reply. In the blockchain variant, PBFT is simplified to main four stages: Request, prepare, commit, reply.

Every epoch of the consensus protocol only has a single leader node. In the start of each epoch, a leader node leader is selected either randomly or sequentially.

1. Request: The application client sends transaction data to all the nodes in the network and request service from the primary node.
2. Prepare: The primary node acknowledges the request and broadcasts the request to the rest of the nodes.
3. Each backup node will verify the transaction data based on their blockchain database. The result of the verification process will be sent to the client and the rest of the nodes.
4. The client and node awaits $f + 1$ replies from different nodes with the same result. The sum of the total positive replies will determine the result.

After a few cycles, the leader node will do a checksum on the blocks generated in this epoch to ensure that the blocks generated in this epoch are identical. Secondary nodes can also request the client to change the leader node if they

believe the leader node is faulty. A supermajority will lead to a change in the leader node.

One of the major benefits of using PBFT is its energy and time efficiency compared to Proof of work. Once the result is determined in the last phase, the block is generated. No additional steps are required to confirm the block. As no mining is required, nodes do not need resource-intensive hardware to solve computationally difficult problems. By complementing PBFT with public key infrastructure, the platform can also prevent spoofing, corrupted messages or relays.

However, due to a large number of communication messages between each node, such a protocol is not scalable for blockchains with many nodes. The leader node system also causes the platform to be prone to Sybil attacks where a single node controls the majority of the network. With these disadvantages, PBFT is not suitable for a large public blockchain. Instead, PBFT can be optimised for smaller permissioned blockchains by supplementing additional off-chain modules or other consensus protocols. In Zilliqa [14], PBFT is combined with POW to achieve a trade-off between scalability and speed. Meanwhile, Hyperledger Sawtooth [15] employs PBFT to create small permissioned blockchain for industrial usage.

Proof of Stake

Proof of Stake was introduced in Peercoin in 2012 to improve block mining and reward allocation efficiency. Instead of solving computationally difficult problems, miners have to 'invest' or 'stake' a fixed amount of their cryptocurrency. A random node will be selected to generate the block after a fixed interval. This probability is weighted based on the number of coins that the miner has staked in. The total reward consists mainly of transaction fees and depends on the total number of transactions in the block. This entices miners to increase the number of transactions validated per block, increase the number of transactions per second validated. As a stake is required, not everyone in the network can join the mining process, i.e. block creation and validation. In the case of Peercoin, the initial pool of coins is still generated via PoW. However, no new coins will be generated after the initial process. Hence, the total number of coins in the pool will remain constant. Most changes and developments in POS are related to improving the stochastic process in which the miners are being selected or how they interact with the staking process.

The POS protocol enables platforms like Ethereum to have a much faster transaction rate with much less energy than POW platforms. Miners in POS platforms also do not need to invest heavily in expensive mining hardware. For PoW, miners that create any block will waste the energy and computation resources they have committed in the PoW process. For POS, the right to mine, the next block is determined before the current block is created.

However, POS cannot be integrated into any application scenario beyond cryptocurrency since there is 'nothing at stake'. Thus a newer consensus protocol was introduced by Ethereum ecosystem for its smaller private blockchains.

Proof of authority

Proof of Authority (PoA) is a relatively new family of consensus protocols used in permissioned blockchain. Proof of authority relies on a set of authorised nodes to publish the blocks. Since the nodes in permissioned or Consortium blockchains have to be pre-authenticated before they are added to the network, it can be assumed that the nodes in the network can somewhat be trusted. Once authenticated, nodes can be given rights to publish blocks in the blockchain or revoke access rights. New transactions that are generated can be validated and added to the block by the nodes in the blockchain and blocks can be published by the nodes using a digital signature. Famous implementation of POA consensus includes Clique in private Ethereum network (Kaleido), Alpa, and an Istanbul Byzantine fault-tolerant protocol in Quorum. The three protocols are different and cater to the different specific requirements and restriction of their environments.

The development team of Etheruem created Clique in 2017 as a quick fix to Proof of work problem in its parity implementation. Attackers would often exploit the simple PoW problems which lead to an inflated block gas limit, crashing the network. The ease of implementation and integration to existing Ethereum ecosystem prompts the parity team to adopt the version of POA known as Clique. Each block header in the chain will contain a list of signatures of the authorised signers. Only when the signature of the publishing nodes matches the signature in the list, will the block be published. To be posted, the rest of the nodes have to verify the signature and cast votes. As a result, only one round of messages exchanged among the authorities is required, compared to multiple rounds of communication needed in PBFT. This result in lower latency at a lower communication cost as compared to PBFT and related protocols. However, the tolerance level of the protocol to byzantine fault is reduced to

$N/2 - 1$ byzantine authority nodes. In other words, Clique can only operate correctly when a simple majority of the authority nodes, $N/2 + 1$, are honest. Specifically, there is a trade-off of consistency for performance and availability.

Inspired by Clique, JPMorgan Chase(JPMC) and AMIS implement a hybrid of PBFT and PoA known as Istanbul byzantine fault tolerant(IBFT) Fintech-based blockchain Quorum in late 2017. The main drive behind the implementation is to improve the security of the consensus protocol. In normal PBFT and related consensus protocols such as RAFT, it assumes that the leader is always trusted. As such, all the follower nodes will blindly follow the leader nodes. This is potentially dangerous in highly valued and regulated sectors like banking and finance. In addition to multiple rounds of voting, IBFT also records the validating parties' signature on the block. The role of "leader", or "block proposer" is removed and instead, verifies the new block just like other consensus protocols operating in trust-less environments. IBFT has the same fault tolerance with PBFT of up to $\frac{1}{3}N$ number of faulty nodes. To tamper with the blocks, attackers will have to access the private keys used to sign off the block, which is very unlikely. The major trade-off lies in the communication overheads. It is slower than existing PBFT due to additional verification process of the signatures. An additional effect of IBFT is the fixed interval of block generation. After every user-defined interval, a block is always generated, regardless of the number of pending transactions. This can be deemed as positive or negative, depending on the configuration and application scenario. In periods of low volume, a large number of empty blocks can be generated, leading to unnecessary storage and junk data. However, the predictability of the blocks allows for stakeholders in the supply chain to better plan and schedule deliveries and production.

A simple version of PoA is implemented in Alpa. Compared to Clique and IBFT, where the environment is defined to be trustless, Alpa assumes that all nodes are trusted, and the communication network is secure. Namely, Alpa is designed to connect businesses in an integrated data sharing and payment platform. This trust model allows Alpa to use a PoA protocol with very high transaction rates and performance at a meagre cost. In the PoA, each of the nodes takes turns to be the leader node. The leader is given a time interval to generate a block containing all the transaction queued on its open transaction list. The block will be signed using the private key of the node generated by ECC protocol and broadcast to the other nodes in the network. Upon receiving the new block from the leader nodes, the rest of the validating nodes will check the transactions in the new block and the signature of the leader node. If the

validation process fails, the block is rejected. If the validation node generates a set number of bad nodes, its publishing rights can be revoked and removed from the network. Finally, the role of 'leader' node will then be passed on to the next validating node on the list.

In general, there are several key performance and security benefits of using the POA protocol.

1. In POA, complex computation or specialised hardware is not required. Access right policies can grant the stakeholder's nodes the right and ability to publish blocks. This reduces the cost and time required for transactions to be added on the blockchain.
2. In addition, a routine and time frame can be created to decide who gets to publish the blocks. This makes the publishing of transactions and blocks predictable, which is important in a supply chain perspective.
3. PoA platforms are resistant to Denial of service attacks as the nodes are pre-authenticated. If attackers compromise a node, the node will be excluded from the list of authorised nodes rapidly. Also, any block or transaction generation and posting require the private key of the node owner. Because of the operating conditions, Blockchains running PoA consensus are also resistant to Sybil attacks, as all identities have to be verified before joining the network.

2.3 CP-ABE

Attribute-Based Encryption (ABE) first proposed by Sahai and Waters [16] is a promising alternative of encryption for achieving fine-grained access control of encrypted data. There are two supplementary forms of ABE: Ciphertext-Policy ABE (CP-ABE) and Key-Policy ABE (KP-ABE). In CP-ABE [17], each user possesses a private key that is associated with his/her attributes. The data owner can specify an access policy that can be defined as a Boolean formula on the related attributes, and embed this policy into a ciphertext. The users successfully decrypt the ciphertext only if their attributes can satisfy the access policy. Whereas KP-ABE [18] embeds an access policy into a user's private key, and ciphertext is labelled with the set of attributes. Therefore, CP-ABE will be more suitable for the data owner to implement fine-grained access control of shared ciphertexts in the untrusted storage environments.

In data crowdsourcing systems, some data owners may manage databases with large amounts of data. When a large number of data requesters purchase data from a data owner, the owner has to retrieve and encrypt the data according to the requirements of each requester. In this case, the data owner will become the bottleneck of data trading. In order to improve the transaction efficiency of data owners, data owners can use CP-ABE to encrypt and pre-process a large amount of data according to the access control policies of data and upload these ciphertexts to the cloud servers. Each requester can find the minimum set of attributes that meet the policy set for the required data and submits the set of attributes to the data owner. After the data owner generates the attribute private key for each requester, the data requesters can obtain their purchase data by using the attribute private key and the location of ciphertexts.

To implement fine-grained authorisation of data owners in data crowdsourcing, we modify the traditional CP-ABE scheme and remove the trusted private key generator (PKG), and instead set data owner to control the ownership of his data and issue the private keys to data requesters. Data owners first encrypt the data to generate a data ciphertext by using symmetric encryption and a session key, then encrypt this session key to obtain a session-key ciphertext by using the CP-ABE [19] and upload these ciphertexts onto cloud servers. However, if data owners have a small amount of data, they can use symmetric encryption to complete the data trades without using CP-ABE.

Chapter 3

Literature reviews

This section contains the literature review in which our work is based. The review is divided into two main parts:

1. Challenges in data sharing in modern supply chains and existing solutions: This section explores some of the problems in existing supply chain management platforms, and the role blockchain should play in the supply chain.
2. Existing implementation of supply chain and logistics blockchain

3.1 Supply chain and logistics data sharing platforms and management software

As manufacturing and trade become globalised, supply chains are increasingly complex due to multiple cross border movement of goods and services and the high volume of demand. Manufacturing companies often require Enterprise Resource Management (ERM) or Supply chain management (SCM) platforms and software to help them manage the supply chain. The supply chain and logistics industry developed the Supply Chain Operations Reference (SCOR) Model in 1996 to standardise basic functionalities and tool kits in these platforms [4]. These range from forecasting, supply chain monitoring, inventory management, transport management to supply chain analytic. Supply chain monitoring provides an overview of the supply chain. It consists of information such as product codes, product IDs, raw material codes, shipment date and ID and consignment information. Inventory management maintains the raw material stock as well as existing manufactured stock. Transport management

or delivery management is in charge of manufactured goods in the transit process. Lastly, supply chain analytics provide forecasting of production as well as any future imports of raw material. Studies on textile and apparel industry [20] [21] [22] have listed the same crucial requirements and features that the textile and apparel SCM platform should have. However, one crucial area that existing platform falls short of is the collaborative functionality between the different platforms and partners.

ERM and SCM platforms are typically standalones, and companies have to use special integration technologies [23] during collaborations. In a supply chain with multiple collaborating partners, consensus on existing records is critical. Business to Business (B2B) technologies was developed to share business transaction records between the collaborating parties. One key technology developed in recent time was the Service Oriented Architecture (SOA) used in platforms such as IBM WebSphere or TIBCO. The Enterprise Service Bus (ESB) in SOA allows the business workflow between different ERM and SCM to be integrated. Business transactions are typically shared in JSON or XML formats while documents and files are shared using FTP and MQ. Technologies like MQ were developed to share documents and files between different platforms, operating systems and servers in a managed and auditable way. However, the effectiveness of such a platform is limited, and the barrier of the data silos between the partners is still high. However, inconsistency in data within the different ERM and SCMs is still a major problem, often leading to disputes.

One main benefit of implementing blockchain technologies in the supply chain is the reduction of data inconsistency between the multiple parties. Horst [24] explore some of the theoretical impacts of applying Blockchain existing supply chain frameworks. The major improvements come in the form of data provenance, trust, privacy, security, enforcement and accountability. These improvements also leads to changes and concerns of adaptability in the supply chain's managerial and operational levels. Moreover, the lack of knowledge in blockchain technologies and Blockchain empowered frameworks and models hamper the progress, despite huge companies' huge investments. Specifically, the reduction in complex inter-and intra-organizational dependencies would require companies to adopt new structural changes in the supply chain. Hence, the next sections explore various blockchain implementations and their impacts and shortcomings in the supply chain.

3.2 Existing implementation of supply chain and logistics blockchain

In 2018, DHL International GmbH entered a joint research venture with Accenture plc to create the first logistics blockchain [25]. The main objective of the project is to design a track and trace platform for pharmaceutical products. Using Blockchain as an asset and inventory management platform, an improvement in transparency and traceability was achieved. The usage of smart contracts also improves the efficiency of the logistics system. Two key innovations that were also introduced in the blockchain platform are editable records and new off-chain modules [26]. The first innovation seems counter-intuitive as blockchains are designed to be immutable and hence editable. However, under some circumstances such as operational errors or malicious contents, the ability to remove contents in the Blockchain is advantageous and essential.

The second major innovation by the collaboration is the close integration between existing off-chain systems and the Blockchain. Features such as QR codes and public key infrastructure improve the compliance of the platform's regulations and security. However, this platform is limited to the consignments industry and is not equipped to track and trace goods and services in a complex supply chain network.

As more research has been conducted in distributed ledger, a large number of supply chain solutions have emerged to solve problems in various industries. Leng and et al. [27] proposed a new design using a dual public blockchain platform to record agriculture rentals. Two separate public blockchains are used to track and trace the movement of agriculture capital goods. The first blockchain stores user information while the second Blockchain performs transactions. Off-chain mapping is used to connect both sets of data. There are two major flaws with this method. Firstly, the financial cost of running such protocols is enormous and inefficient. Secondly, this implementation is not suitable for complex supply chains, requiring multiple transactions and a huge amount of information storage. With the cost of cryptocurrency ever increase and non-stable, the cost of using public blockchains for trace and track of products is increasingly non-viable.

3.2.1 Permissioned Blockchain for supply chains

The need for performance drives the shift from public Blockchain to permissioned or private Blockchain. Hyperledger played an essential role in the development of permissioned blockchains. Launched in 2016 by the Linux Foundation, Hyperledger is an open-source blockchain enterprise platform. The primary objective of Hyperledger is to allow companies to develop and build business solutions or eco-system based on an enterprise-scale blockchain framework. Its modular design allows for companies to plug and play different libraries, consensus or database. This design also enabled companies to integrate existing infrastructure with the blockchain platform better. Hyperledger has a few features that make it more applicable to a supply chain scenario than other blockchain platforms like Ethereum. Namely, Hyperledger has a unique transaction mechanism, special sets of callable APIs known as Chaincode, and access controls and identity management.

Hyperledger's transaction mechanism allows companies to define a limit of endorsement peers to sign off each transaction. The transaction is only successful when all the peers sign off the transaction using a signature generated by the organisation's Certificate Authority (CA). The transaction flow can be summarised as follows:

1. The Blockchain stakeholders set an endorsement policy that includes the number of endorsing peers required and the combination of peers required.
2. When a stakeholder makes a transaction, he/she sends the transaction to the endorsing peers.
3. The endorsing peers verify the transaction's content by executing chaincodes and sign off the transaction.
4. The sign/endorsed transaction back to the sender.
5. The sender submits the signed transaction to an ordering node
6. The ordering node adds the transaction to a block and sends the block to the network's peers.
7. Peers node will check if every transaction's endorsement policy requirements in the block are met. Conflicting transactions are also checked. Else, the block will be added into the chain.

Another key architecture design of Hyperledger lies in the multi-channel communication system. A Hyperledger channel is a private sub-network between nodes in the Hyperledger blockchain platform. The channel's communication protocol is similar to a pub-sub protocol where users have to register and be authorised to join a network. In Hyperledger, this registration and authorisation are done by a membership services provider (MSP). Only registered users can send and receive transactions or call for services. Before creating any channels, companies in the network have to predefine channels rules and essential members in the channel. A new genesis block is generated when the channel is being created. Nodes outside of the channel will not be able to access the transactions. Essentially, each new channel is a new blockchain ledger. A node can host multiple ledgers by joining the equal number of channels.

There are several benefits to the designs of Hyperledger when implemented on a supply chain. The Hyperledger transaction mechanism and multi-channel design removed the need for a consensus protocol. While companies can decide to integrate a new consensus protocol into the platform, it is not essential. By using trusted endorsing nodes to sign off transactions instead of a consensus protocol, the platform is scalable and have higher number of transactions per second as compared to public blockchains. The reduction in decentralisation is not a significant concern in a supply chain consortium blockchain and more desirable. In the supply chain, Regulatory Compliance APIs are required in the platform to ensure compliance. The multi-channel design also allows Hyperledger to host different business chains. Different suppliers or manufacturers can join a common platform to share information and make transactions without leaking any confidential information.

3.2.2 Use cases and case studies

To address the problem of trace and track of vaccine manufacturing and distribution, Kumar et al. [28] have proposed using the blockchain platform to record essential delivery and storage information. Their Blockchain is built upon on Hyperledger platform, moving the cost incurred when making transactions. Their Blockchain also implemented a proof of authority to allow for access control of the smart contracts. Only authority governmental bodies are allowed to make transactions. However, there is a lack of privacy control in their implementation. In terms of supply chain platforms, privacy control is important as confidential information is only for certain parties in the supply

chain. Information such as product composition and supply level is private information that affects the stakeholder's competitiveness when leaked.

V.G. Venkatesh et al. [29]. have conduct studies relating to the architecture for a blockchain that focuses on the social sustainability of the supply chain. Their blockchain design aims to integrate multiple technologies such as IoT and big data analytics to allow manufacturers to efficiently and effectively trace the origin of their materials in a multi-tier supply chain. They also investigated the barriers to setting up such a platform. The main benefit in their protocol is integrating technology to automatically generate and store information of different stakeholders and users in the Blockchain. The users are divided into three different groups, strategic level, tactical level and operational level. The three groups have different levels of access rights in the system, with strategic level users having the highest level of access and operational level the lowest. The operation level mainly generates operational data while the data is processed and monitored at the tactical level. At the strategic level, the users can set data policies and platform configurations. However, as stated in their paper, the cost of the infrastructure is huge. Furthermore, there is no mention of the technological challenge in operating such a system or how a blockchain should be configured and designed to handle the massive number of transactions. The paper also assumed that the supply chain stakeholders are fully integrated under a single platform, and information can flow freely. Privacy among the different stakeholders is not considered in the platform. There must be a differentiation between the stakeholders in a practical platform, and the privacy between stakeholders must be preserved.

Qi Feng et al. [30] investigated the privacy requirements and threats and some of the existing solutions in their review paper. In the survey, they consider the following privacy requirements in a blockchain. Firstly, no information should be able to be derived from the transactions. Attackers should not be able to link transactions to other previous transactions. Secondly, the content of the transaction should only be made available to the parties making the transaction. To fulfil these requirements, several solutions are proposed, notably the implementation of Identity-based encryption in Blockchain.

Meng Zhang et al. [31] implemented a new approach to preserve the privacy of data stored in Blockchain was identity-based encryption. Identity-based encryption allows users to generate a public key based on a unique identifier such as mobile number, emails or employee ID. A trusted third party server, the private key generator (PKG), generates the corresponding private key based on the

public key provided. In their scheme, the encryption protocol in the Blockchain is divided into multiple phases. To initialise the protocol, the PKG generates a master public key and private key using elliptical curve cryptography. Suppose a user Bob wishes to encrypt his data stored on the Blockchain, he will have to generate his public key using Bob's ID, and the public key is bind to his mobile number. The protocol also assumes a secure channel where the PKG can pass the private key back to Bob. Bob encrypts his data in the next phase and stores the ciphertext as a search key on the Blockchain. The blockchain interface will require Bob to input a verification code to complete the query. Finally, Bob can decrypt his data using his private key. The authors also propose an additional feature in their protocol by adding a signature scheme based on IBE when the users upload any data.

The protocol's main improvement over the existing public key infrastructures is that it does not require any prior generation of user keys. However, there are several problems with their protocol and implementation. Firstly, their protocol is not ideal for information sharing in a supply chain. It mainly functions as data storage of encrypted information. Storing data in a blockchain structure is inefficient due to the multiple state replication and hashing of data. While the author provides a scheme to convert the protocol to perform encrypted information sharing, it deviates from a blockchain use case scenario. Secondly, the protocol relies on a centralised PKG, which carries a considerable security risk. The authors state that the master keys can be shared with the platforms' supervisors to regulate all the data and transactions. In the case where the master key is leaked, the entire encryption protocol will be compromised. The authors suggested a multi PKG approach where the master key is divided into multiple parts with each part of the key being held by a PKG. While the author did not provide a concrete solution on how this can be achieved, most existing secure multiparty computation protocols can perform such function.

Chapter 4

CP-ABE For privacy protection in data crowdsourcing Blockchain

In this chapter¹, a system model of BC-FGA-DCrowd is provided. Based on this model, we construct a system framework of BC-FGA-DCrowd. Next, we elaborate on the execution process of BC-FGA-DCrowd. Lastly, we discuss the users' malicious activities and propose the system security threat model and give the system security assumption.

4.1 BC-FGA-DCrowd system model

The BC-FGA-DCrowd system model is shown in Fig.4.1, there are five kinds of entities: Data Requester, Data Owner, Administrative Client (AC), Third-Party Entity, and Cloud Storage Provider (CSP).

- **Data Requesters**, identified by a set $DRs = \{ R_1, R_2, \dots, R_m \}$, post their data requests by transferring data descriptions into smart contracts. Each data requester R_i is identified by the identity and attribute set $(pk_{Ri}, sk_{Ri}, a_{Ri}, \omega_{Ri} = (A_1, A_2, \dots, A_k))$, where pk_{Ri} , sk_{Ri} , a_{Ri} denote respectively his/her public key, private key and address, ω_{Ri} is a set of attributes. To ensure the fairness of data trades and the system security, each data requester

¹This chapter has been published as:

Ma, Haiying & Huang, Elmo & Lam, Kwok-Yan. (2020). Blockchain-based mechanism for fine-grained authorization in data crowdsourcing. Future Generation Computer Systems. 106. 10.1016/j.future.2019.12.037.

Figure 4.1: The system model of BC-FGA-DCrowd

must lock a deposit on the Blockchain before data trades, and the deposit consists of a certain amount of reward d_1 and a penalty d_2 .

- Data Owners**, identified by a set $DOs = \{ O_1, O_2, \dots, O_m \}$, own the intellectual property of large amounts of valuable data and can specify different access control policies for different confidential data to manage this database. Each data owner O_j is identified by the identity and attributes information $(pk_{O_j}, sk_{O_j}, a_{O_j}, \omega_{O_j} = (A_1, A_2, \dots, A_k))$, where pk_{O_j} , sk_{O_j} , a_{O_j} denote respectively his/her public key, private key and address, ω_{O_j} is a set of attributes. To ensure the fairness of data trades and the system security, each data owner must lock a deposit on the Blockchain after posting a data sales information. Assume that the owner O_j has the data $\{ F_1, F_2, F_3, \dots, F_n \}$, for $\beta = 1, 2, 3, \dots, n$, each piece of data F_β is specified to an access policy $(\mathbf{A}, \rho)_\beta$. To reduce the burden of retrieval and encryption in data sales, the owner O_j uses CP-ABE and symmetric encryption to preprocess his/her large amounts of data. To reduce the burden of data storage locally by himself, the owner O_j uploads the ciphertexts onto cloud servers. The data owner O_j posts and publishes a sales data description $D = \{ D_1, D_2, D_3, \dots, D_n \}$, where $D_\beta = \{ s_{id}, H(F_\beta), Sig(H(F_\beta), sk_{O_j}), pk_{O_j}, KS_\beta, (\mathbf{A}, \rho)_\beta, Coins(rw_\beta), Coins(\pi_{O_j}), t_{valid}, redeem(.) \}$, $\beta=1, 2, 3, \dots, n$, $H(F_\beta)$ is the hash value of data F_β , $Sig(H(F_\beta), sk_{O_j})$ is the signature of the data owner O_j , KS_β is the data keyword set, $(\mathbf{A}, \rho)_\beta$ is the access policy of data F_β , rw_β denotes the corresponding rewards of data, t_{valid} denotes the data valid date, π_{O_j} denotes the money penalty.
- Administrative Client (AC)**, viewed as an important intermediary for

the semi-trusted third party entities, data owners and data requesters, can run on user's local computer. DRs and DOs can register their identities and attributes via the **Client**, which can create smart contracts for them. The third-party entities verify their identities and attributes via the **Client** and use the smart contracts to record the result of verification. Moreover, the **Client** can help DRs and DOs to agree to their data trade and has to safeguard the smart contracts from any tampering and handle disputes between DOs and DRs.

- **Third-Party Entities.** The third-party entities can be companies and institutes that provide data expertise in the operation of the protocol. These entities can range from law firms, education institutions to research institutes, *etc.* Initially, these entities will be selected by the platform. After the initial setup, new members can be voted by the rest of the third-party entities. These entities have to be authenticated to gain access to the data and functions in the protocol before they can be voted in. These entities can provide multiple services, such as user enrollment, data evaluation, dispute management, *etc.* Firstly, third-party entities verify users' identity and attributes, and the result of the verification is recorded into smart contracts. Secondly, third-party entities evaluate the quality of the DO's data. The final rating on the quality of data will be based on the sum of votes from the panel of third-party entities. This provides a point of reference for data requesters. Thirdly, they will also handle any trade disputes between data requesters and owners. Each user verification, data evaluation, or dispute handling provides monetary incentives to third-party entities.
- **Cloud Storage Provider (CSP)** refers to the entity that manages cloud servers to store the encrypted data for the DOs. CSP may return the location URL_{CT} of the ciphertext to the DO, and send the ciphertext to the DR according to the location URL_{CT} of ciphertext.

In this system, we use the Blockchain and smart contract techniques as a trading platform without trusted central authorities. Certain Blockchain can utilise a Turing-complete programming language to implement smart contracts in Virtual Machines (VM) to provide cryptographically tamper-proof trustworthy execution and enforcement of these contracts. Secondly, the Virtual Machines environment ensures program execution without any potential interference. Data owners can employ the CP-ABE primitive to encrypt data in advance while realising securely fine-grained ciphertext sharing. We use blockchain technology to make the data trading fair transparent. Data requesters use smart

contracts and their attributes to buy data on demand. If the DO's data content can satisfy the DR's data request, and the DR's attribute set can match the DO's access policy, the DR posts the data evaluation function to the third-party entities. When the DR reaches agreements with the DO, the DO can achieve fine-grained authorisation for the DR with the DR's attributes. If the data quality is high, the DR may obtain the data, and the DO is assigned with corresponding rewards.

4.2 BC-FGA-DCrowd system framework

Our system framework contains three layers: the application layer, the blockchain layer, and cloud storage layer. As shown in Fig.4.2, the application layer mainly includes user manager, metadata manager, and program compiler. Each DO can post the descriptions of sales data, which is retrieved by DRs in the application layer. The blockchain layer takes the state changes of data trades as inputs to implement the consensus between DOs and DRs. Since the Blockchain has the limited capacity to store data, we only put the description information of sales data (such as DO's signature, data size, hash value, ciphertext location) into the blockchain layer, and store encrypted data on the cloud storage layer. Users can verify the data authenticity and integrity via the blockchain layer without trusting the data storage layer. We assume the blockchain layer has a program compiler and supports the execution of smart contracts. The user interface in AC is used for the third-party entities, DOs and DRs to interact with smart contract and Blockchain.

4.2.1 Application layer

The application layer can provide users with an interface to complete data crowdsourcing. It mainly consists of **Administrative Client** and contains three modules: user manager, metadata manager, and program compiler. User manager module manages the user's registration information. Users first register with their key pairs (a public key and secret key) by providing their identities and attributes and create their smart contracts. The third-party entities verify their identities and attributes, and the results of the verification will be recorded into Blockchain by the smart contracts. Metadata manager act as the data description information and data trade management. The program compiler module is used to convert the new smart contract into the executable language of Blockchain. The user successfully registers if the smart contract is written into the Blockchain. Then, the user can post or request description information of data trades by the metadata manager. Data crowdsourcing is

Figure 4.2: BC-FGA-DCrowd framework

depicted into smart contracts which run on the Blockchain, including posting sale data description, request data description, searching metadata information, data evaluation, fine-grained authorisation, reward assignment, dispute management. We will describe the concrete scheme of BC-FGA-DCrowd in section 4.3.

4.2.2 Blockchain layer

The blockchain layer should support the smart contract execution and provides consensus on the state changes of data trade. In addition, the Blockchain has to support cryptocurrency payment services such as e-wallet and payment transfers. The Blockchain should be callable to return machine state and receive new transaction inputs and trigger the state changes of data trades according to the valid input from the application layer. For Blockchain like Ethereum, it should be able to store smart contracts as input from the application layer. We use the smart contract to control the logic of fair data crowdsourcing. In particular, we wish to extract the metadata (such as owner signature, data hash, access policy, keyword set, location of ciphertext, *etc.*) from the storage of encrypted data and use it as an input into the Blockchain while the encrypted data is stored off-chain on cloud servers. This storage mode can significantly improve the data storage capacity of the Blockchain and contribute

to the synchronisation of the network.

4.2.3 Data storage layer

Data storage layer mainly refers to cloud storage servers and is used to store the encrypted data as well as private/sensitive information. User data are stored in cloud servers, and only authorised personnel are allowed to access them. The data hash is signed by the owner's secret key. Users can check the data integrity and authenticity via the data hash value and signature on the blockchain layer. To achieve fine-grained authorisation of ciphertext sharing, we use CP-ABE and symmetric encryption technology to encrypt the data, so that the data owner can embed the different access control policies into different ciphertexts. Therefore, data owners can preprocess the encryption work of an amount of data before data trading, and data requesters can use their attributes to buy data on demand. Compared with traditional symmetric encryption methods, CP-ABE can greatly reduce the data owner's retrieval and encryption workload in data sales. The data owner can upload the ciphertexts onto cloud servers to reduce the burden of ciphertext storage locally.

4.2.4 The process of BC-FGA-DCrowd

We describe the process of BC-FGA-DCrowd as follows:

- **Step 1:** Data owners and data requesters first register in the system of BC-FGA-DCrowd. The **Client** takes the user's identity and attributes as input and create the smart contracts of user register. The third-party entities verify their identities and attributes via the **Client** and use the smart contracts to record the result of verification into the Blockchain. Each registered user is associated with his/her identity, attributes, and public-secret key pair, which can be used to create a secure channel between two users.
- **Step 2:** Each registered DO performs the CP-ABE setup algorithm to obtain his/her system public parameters spp and master secret key msk . Assume that the DO has an amount of data $\{F_1, F_2, F_3, \dots, F_n\}$, for $\beta = 1, 2, 3, \dots, n$, each piece of data F_β is specified to an access policy $(\mathbf{A}, \rho)_\beta$. The DO uses CP-ABE and symmetric encryption to encrypt his/her large amounts of data, and uploads the ciphertexts onto cloud servers and gets the location URL_{CT} of the ciphertexts returned by CSP.
- **Step 3:** Each registered DO extracts the metadata (such as DO's signature, data hash value $H(F_\beta)$, URL_{CT} , $(\mathbf{A}, \rho)_\beta$, keyword set KS_β , reward

rw_β , data validity period t_{valid} , timestamp, *etc.*) about the raw data F_β . In order to ensure the data quality, each DO has to make a deposit when he/she posts a data sales message to the blockchain network.

- **Step 4:** Each registered DR retrieves the sales data information in the DOs' evaluation of smart contracts and finds the appropriate DOs only if the DR's information can satisfy the necessary condition functions. The DR can find his/her minimum attribute set i that can satisfy all access policies $\{ (\mathbf{A}, \rho)_\beta \}$ of the required data, and submits it to the DO. After the DO verifies the attributes, they make an agreement. To avoid the DR's refusal to pay, he/she must lock a deposit on the Blockchain before obtaining the data. The DR posts a data evaluation function to the third-party entities, which can evaluate the raw data from the data owner.
- **Step 5:** After receiving the attribute set of the requester R_i , the DO runs the CP-ABE key generation algorithm to generate the attribute private key dk for the DR. The DO runs an asymmetric encryption algorithm $Enc(pk_{R_i}, dk)$ to get the attribute-key ciphertext CT_{dk} , and embeds it into the smart contract of data trade ($ROTC$). When the address of the $ROTC$ contract is recorded as a transaction on the Blockchain, the DO sends the transaction index, the $ROTC$ contract address, the source code to the DR through a secure channel, which is created by the public-private key pair of two trade users.
- **Step 6:** The data requester DR reads the ciphertext CT_{dk} from the Blockchain and computes the attribute private key $dk = Dec(sk_{R_i}, CT_{dk})$. Then he/she reads the metadata from the Blockchain and downloads the ciphertext $CT = \{ (CT_{K_\beta}, CT_\beta)_{\beta=1,2,3,\dots,n} \}$ from the cloud server according to the location URL_{CT} . The DR performs the CP-ABE decryption algorithm to get the session key K_β , and computes the data $F_\beta = Dec(K_\beta, CT_\beta)$. Then the DR obtain his/her purchase data set $\{ F_\beta \}$.
- **Step 7:** This step consists of data evaluation and reward assignment. Before the DR gets the purchase data, the third-party entities output the evaluation results, which must be recorded on the Blockchain. Only if the data is high quality, the data reward is paid to the DO. Otherwise, the deposit is unlocked.
- **Step 8:** Updating smart contracts can be viewed as a transaction which will be verified by nodes of the blockchain network. The above steps are associated with this step, and the metadata and status of each confirmed

transaction are permanently stored on the Blockchain. In case of data transaction disputes, both the DR and the DO will select an equal number of third-party entities to review and verify the data and trading details, which are recorded on the Blockchain.

4.2.5 System security threat model and security assumption

In the BC-FGA-DCrowd system, there are two different security threat models which are based on the permissioned Blockchain and the public Blockchain. We combine both models due to our protocol having characteristics of both types of Blockchain. The characteristics of different blockchains can affect how each party in the data trade interact with one another, and thus, the level of security threat is different. Our protocol mainly consists of three kinds of entities: data owner, data requester, and third-party authorisation entity, who can bring different levels of security risks to the system on top of the tradition external attacks to the Blockchain. Our protocol should have four basic trust assumptions as follows:

1. There exist a governance framework, and some system audit functions to enforce accountability of the third-party entities.
2. Third-party enterprises have their own access controls measures to prevent malicious activities.
3. The public Blockchain is secure.
4. Data owners and data requesters must take measures to ensure the safe storage of their passwords and private keys.

Firstly, we assume that there exist a governance framework and some system audit functions to enforce accountability of the third-party entities. In the unfortunate situation, when the third party companies or users conspire to cheat the system, their malicious activities are recorded in the Blockchain. These blockchain records function as audit logs, and they will be held liable for their misbehaviour by a governance framework which is outside of the scope of this paper.

In permissioned Blockchain, each user is accountable to their activities on the system as their identities are authenticated and verified. In this context, all unauthorised users cannot be allowed to perform any operation, and the operating environment can be assumed to be semi-honest. Thus the main concerns

of the users are from external attacks and privacy leaks from off-chain storage of personal data.

Public blockchains are deemed to be a trustless environment due to its user's anonymity and decentralising nature. In the platform, users are of unknown origins and are prone to attacks by both internal malicious users and external attackers, and any single party cannot control over the platform, which is unlike permissioned or private Blockchain. The public Blockchain has several prevention and protection mechanisms to safeguard the platform from security threats. Therefore, we assume that the consensus protocol of the public Blockchain is secure and cannot generate the fork chain. Considering secure transfer with wallet in the Blockchain, we assume that every user may securely possess his/her secret key and transfer coins by using the client wallet. To ensure the security of data storage and trade, we assume that the cryptographic algorithms are secure. To further protect the fairness and the security of data trades on the public Blockchain, we modify a time-locked deposit protocol to resist malicious activities.

Personal privacy is a major concern for both data owners and data requesters in the data trading platform. Since personal information (identity and attributes) of users must be verified by third-party entities, this may lead to the leakage of personal privacy. To protect user's privacy, in the permissioned Blockchain, many access control policies are set to restrict unauthorised users to access such private data, which are usually stored off-chain in a separate database. The data trading information of permissioned Blockchain can be only accessed to authorised users. In public blockchains, all users can trade data purely based on unique wallet addresses and hashes. To ensure the security of data trades, our protocol links each user's unique hash and address with his/her personal information, and the data transaction information is stored in the public Blockchain in the form of ciphertext. Personal information must be stored in off-chain databases, accessible only by approved and authenticated third-party entities. Our protocol allows public Blockchain to mimic the operating model of a permissioned blockchain while retaining a level of flexibilities for users. From the above, the risk of personal privacy leakage mainly comes from the malicious activities of the third-party entities. In this respect, we make major assumptions on top of assumption(2). In a similar fashion to proof of stake, these third-party entities are assumed to have a stake in the operation of the protocol and can obtain monetary revenues by maintaining the proper operations. With both assumptions in mind, off-chain data storage can be protected using typical access control mechanisms by these third-party entities. In assumption (2), we assume there is an internal auditing framework that exists

in third party companies. This can be done by setting up access control infrastructures that issue access tokens in third party companies, and any employee must provide credentials to gain the access tokens. Their activities will then be monitored and logged. In addition, we assumed that the third party companies would adopt a secure internal infrastructure to prevent heavy liabilities arising from their data breach. In the unfortunate scenario where there are malicious employees, third party companies prove forensic evidence from their company logs.

In our protocol, there are several possible malicious activities of both data owner and data requester to maximise their own economic profits. Data owners can potentially trade erroneous or corrupt data. As data owners encrypt their data, data requesters cannot fully know what the data/files contains. Data requesters can be at risk of obtaining data infected by a virus, which can infect other documents or download malicious payloads. Secondly, Data owners can also sell data that does not match their data descriptions or refuse to send the correct data after receiving payments. This malicious behaviour can easily occur when the amount of data is very large. To reduce the data owner's malicious behaviours, our protocol uses authorised third-party entities to verify the trading data and filter viruses. When the malicious activities of data owners are found, the authorised third-party entities also keep track of their actual identities. In addition, we introduce a time-locked deposit protocol to have a cool-off period in case of any corrupted data or missing data. In the dispute function of the protocol, multiple third-party entities will come together to match data content and description under a non-disclosure agreement. Malicious data requesters aim to obtain high-quality data without paying rewards. To reach the goal, they will deliberately misreport the evaluation result as low-quality even if data owners provide high-quality data, and even deny they have got the data. To withstand the data requester's malicious behaviours, our protocol uses the authorised third-party entities to verify the trading data and uses a time-locked deposit protocol, which requires them to make deposits before data trading. If data requesters deny they obtain high-quality data, our protocol may provide real evidence to guarantee the rights of data owners and trace their accountabilities. In addition, we require assumption (3) and (4) to be true. If the blockchain platform is compromised, it is extremely difficult for our platform to function. It is common that the platform users are typically the weakest link in any security protocols. All users are prone to fishing attacks or unsafe practices. Any leakage of private key and password will result in data loss as well as malicious sales of data. To mitigate the impact of any security breach

in the Blockchain, each third party enterprises will maintain backup logs of all transactions. Lastly, to improve on the assumption(4), tougher authentication methods can be used to increase system security, such as multi-factor authentication.

4.3 Blockchain-based mechanism for fine-grained authorisation in data crowdsourcing

In this section, we propose a blockchain-based mechanism for fine-grained authorisation in data crowdsourcing by combining the Blockchain, smart contract, off-chain applications, CP-ABE, and the cloud storage system. We design the Blockchain to support smart contracts which are pre-defined programs and construct four kinds of smart contracts: Register User Contract (*RUC*), User Evaluate Contract (*UEC*), and Requester-Owner Trade Contract (*ROTC*), Dispute Contract (*DC*). We separate user's (data owner or data requester) information into two parts: identity information and attribute information. The identity information includes the user's name and physical address and is stored secretly in an off-chain database to protect the privacy of users by the third-party entities. The attribute information which will be stored using *RUC* contains user attribute set ω , data keyword set KS , data type, and access policy (A, ρ) , etc. When a user successfully registers using *RUC*, the corresponding *UEC* will be created. Moreover, the data owner and requester can use *ROTC* to conduct the data trade according to the set of constraint conditions.

We define four important algorithms: deposit-locked algorithm *LockDeposit*(.), policy matching algorithm *MatchPolicy*(.), data searching algorithm *SearchData*(.), data evaluation algorithm *EvaluateData*(.). The algorithm *LockDeposit*(.) is to lock the necessary deposits on the Blockchain before the deadline and allocates rewards to the designated entities according to the results of the algorithm *EvaluateData*(.). The algorithm *MatchPolicy*(.) is used to check whether DR's attributes can satisfy the access policies of his/her required data, and is automatically executed by the smart contract on Blockchain-based on existing policies to check its validity before calling the trade request.

The algorithm *SearchData*(.) is used to search data sources to find the appropriate DOs according to the data-required description of DRs. The algorithm *EvaluateData*(.) can accurately evaluate the data quality by multiple

third-party entities, which are selected randomly by the system. In this work, we assume that there already exists a completely reliable data search algorithm *SearchData(.)* and a completely reliable evaluation algorithm *EvaluateData(.)*, which will not be described in detail and regarded as our future work. The reward assignment depends on the output of *EvaluateData(.)*. For simplicity, the evaluation output is “H” (high-quality data) or “L” (low-quality data).

In particular, the off-chain parts of the system mainly consist of data encryption, decryption, key generation, smart contract creation, metadata extraction, data evaluation, dispute management, and lastly user interaction. The on-chain parts consist of four smart contracts to read and create and process transactions in the Blockchain: Register User Contract (*RUC*), User Evaluate Contract (*UEC*), Requester-Owner Trade Contract (*ROTC*), and Dispute Contract (*DC*), which need record the results of the off-chain algorithms.

4.3.1 Smart contracts for data crowdsourcing

Register User Contract (*RUC*). Upon registration, each data owner or requester submits his/her true identity and attributes and is assigned a key pair (public key, secret key). After the identities and attributes of users are certificated by the third-party entity of platform, the *RUC* receives the inputs from the user register program and generates transactions to register the user into the Blockchain. It generates the user’s address by the hash of his/her public key and contains user identity information, attributes set, data description, his/her public key and a digital signature signed by the third-party entity. Data requester’s attribute set is an important parameter and determines whether the requester can enter the data purchase process. Data description mainly includes a keyword set *KS*, an access policy (A, ρ) , size, type, valid period about the owner’s data. Noticed that the transaction fee for creating or updating smart contracts is paid by the contract publishers.

User Evaluate Contract (*UEC*). The *UEC* stores user profile, attribute set, data description, and two algorithms *MatchPolicy()* and *SearchData()*. Profile mainly contains user identity information and a digital signature signed by the third-party entity, and users’ identities can be verified via their public keys. DR’s attribute set and keyword set will be two important parameters which can determine whether the DR can enter the data purchase process. Data description mainly includes a keyword set *KS*, an access policy (A, ρ) , signature, data size, data type about the owner. DRs can find their required data sources by querying DO’s *UEC* and verify the signature with

DO's public key. *UEC* also includes an address list of data which can point to DOs' previous data in *ROTC*.

Requester-Owner Trade Contract (*ROTC*). The *ROTC* enforces and conducts the trade agreement between DOs and DRs, and includes the processes of data post, data trade, and reward assignment. It can be created after the data owner O_j posts and publishes a data description information $D = \{ D_1, D_2, D_3, \dots, D_n \}$, where $D = \{ s_{id}, H(F), Sig(H(F), sk_{O_j}), pk_{O_j}, KS, (\mathbf{A}, \rho), Coins(r_w), Coins(\omega_{O_j}), t_{valid}, redeem() \}$, $= 1, 2, 3, \dots, n$. The data requester R_i retrieves the list of data description in *RUC* and checks the required data description $\{ D \}$ via the owner O_j 's public key. The *ROTC* contract can be created after the *UEC* returns the required data description set $\{ D \}$, and will take in the participant blockchain addresses, *UEC*'s contract addresses, price, data hash values and the ciphertext's location as inputs. Each DO define a pool R_{pool} to store the authorised requesters' addresses. If the authorised DRs need to be added into the pool R_{pool} , the data owner updates the *ROTC* contract by broadcasting it to the blockchain network, which means they can successfully enter the data trade process. Once the DO and the DR give final approval for the trade, the funds will be locked from the DR's wallet. After the funds are locked, the *ROTC* will require the DO to input the DR's attribute private key ciphertext. When the trade is completed, the funds will be transferred from the DR's wallet to the DO or redeemed.

Dispute Contract (*DC*). The *DC* contract can be used to resolve the dispute between user data trading. In the management of disputes, both the data requester and the data owner select an equal number of third-party entities to review and verify the data and trading details. This forms a panel of juries which will vote on the outcome of the dispute. Their votes will be recorded by smart contracts in the Blockchain. In addition, the platform will also select random third-party entities to be included in the panel of juries. For instance, in a jury of 7 third-party entities, three entities will be selected by the data owner, and three entities will be selected by the data requester. This ensures the fairness of the platform as data owner and data requestors are more likely to select juries which are more likely to vote in their favours. However, it is not fair to purely select the juries based on random selection due to the differences in data regulations and culture as well as native languages in cross border trades. This means that the platform would have to be flexible enough to allow for many third-party entities to join our platform.

Since time-locked deposits are widely used in blockchain-based protocols as a mechanism to punish misbehaving parties [29-31], we introduce a time-locked

deposit protocol to the data crowdsourcing scenario and propose a time-locked deposit protocol for fine-grained authorisation shown in Fig.4. In order to guarantee the fairness of data trade, each user (data requester or data owner) must make a deposit before executing data crowdsourcing. Any malicious activity will result in a forfeit of the deposit. Data requester can post the function *EvaluateData(.)*, which takes in the trade data from the owner. Then the data quality is verified by authorised third-party entities, and the result of data evaluation is recorded in the *ROTC* contract. We assume that such authorised third-party entities are interested in the long-term profits of maintaining the platform. Any compromise to the platform affects their long-term profits.

Time-locked deposit protocol for fine-grained authorisation

In this protocol, each user (data requester or data owner) must make a deposit before executing data trading. When the result of data evaluation is recorded in the *ROTC* contract, this protocol assigns deposits to participants in the light of pre-defined smart contracts, and the deposit cannot be redeemed until deadline even if the participant owns his/her secret key. The malicious participants will be penalised by money if they break the fine-grained authorisation protocol, which is pre-defined in the refund-or-reward scheme Π_{RR} . At the same time, the honest participants are compensated with the malicious participants' deposits—the scheme Π_{RR} allows the requester to pay rewards to the data owner under the pre-defined constraints. We define the scheme Π_{RR} below.

Time-locked deposit protocol for fine-grained authorisation Π_{RR}

- **Phase 1.** Upon receiving a data description $DDeposit = \{ \text{Coins}(\pi_{O_j}), D_1, D_2, D_3, \dots, D_n \}$ from the data owner (DO) O_j , where $D_\beta = \{ s_{id}, H(F_\beta), \text{Sig}(H(F_\beta), sk_{O_j}), pk_{O_j}, KS_\beta, (\mathbf{A}, \rho)_\beta, \text{Coins}(r_w), t_{valid}, \text{redeem}(\cdot) \}$, $\beta = 1, 2, 3, \dots, n$, s_{id} is a session id , $\sigma_{O_j} = \text{Sig}(H(F_\beta), sk_{O_j})$ is the DO's signature, pk_{O_j} is the DO's public key, KS_β is the data's keyword set, t_{valid} is the data's valid period, the DO's deposit $\text{Coins}(\pi_{O_j})$ is transferred to $DDeposit$ with the DO's signature to thwart Sybil and DDoS attacks on the blockchain. Record this message $DDeposit$ on the blockchain by creating a *ROTC* contract such that DRs can retrieve it. Taking the tuple $(\omega_{R_i}, KS_r, \{ (\mathbf{A}, \rho)_\beta, KS_\beta \})$ as inputs, the redeem script is created following the function:

$$\text{redeem}(\cdot) = \text{Verify}(\sigma_{R_i}, pk_{R_i}) \wedge (\text{Verify}(\sigma_{O_j}, pk_{O_j}) \vee \text{EvaluateData}(\cdot)).$$

- **Phase 2. RDeposit.** A data requester (DR) R_i retrieves $DDeposit$ according to his/her attribute set ω_{R_i} and the keyword set

KS_r of required data. If $KS_\beta \supseteq KS_r$ and $\omega_{Ri}(\mathbf{A}, \rho)_\beta$, the DR broadcasts a message $RDeposit = \{s_{id}, pk_{Ri}, KS_r, \omega_{Ri}, t_{deadline}, t_{confirm}, \text{Coins}(d_1+d_2), SearchData(.), EvaluateData(.)\}$ to the blockchain network, where $\text{Coins}(d_1+d_2)$ is the deposit which cannot be unlocked until the trading deadline $t_{deadline}$ by the DR or the result of $EvaluateData(.)$ with the signature σ_{Oj} in Phase 1.

- **Phase 3. Claim.** The authorised DR R_i requests the location of the encrypted data in the Blockchain. The DO generates the attribute private key dk for the DR and uses the DR's public key to encrypt it. The DR reads the result of the data evaluation from the Blockchain and sends $\sigma_{Ri} = \text{Sig}(\{H(F_\beta)\}, sk_{Ri})$ to the DO. The DO submits the **Claim** transaction in the *ROTC* contract by providing the signatures σ_{Ri} and σ_{Oj} to redeem the reward:

$$Claim = \{s_{id}, pk_{Ri}, \sigma_{Ri}, pk_{Oj}, \{H(F_\beta), \sigma_{Oj}\}, \text{Coins}(r_w), \text{Coins}(\pi_{Oj})\}.$$

- **Phase 4. Reward.** The DR or DO can initially call the reward phase. They check whether *DDeposit* and *RDeposit* are not deleted and obtain enough evidence on the Blockchain. Then they broadcast the transaction that can redeem the deposit:

If $EvaluateData(.) = H$, transferring $\text{Coins}(r_w + \pi_{Oj})$ to the DO.

If $EvaluateData(.) = L$, transferring $\text{Coins}(r_w)$ to the DR and $\text{Coins}(\pi_{Oj})$ to the DO.

4.4 Concrete BC-FGA-DCrowd scheme

In this section, we propose a blockchain-based protocol for fine-grained authorisation in data crowdsourcing. It consists of seven algorithms: User Register, Encrypt Data, Data Sales, Search Data, Fine-grained authorisation, Decrypt, Reward Assignment, and Confirm Transaction. Users (DOs or DRs) can interact with this Blockchain via the **Client**.

4.4.1 User Register

The DO O_j or DR R_i can register to get his identity and attribute information via a *RUC* contract, *i.e.* $O_j = (pk_{Oj}, sk_{Oj}, a_{Oj}, \omega_{Oj})$ and $R_i = (pk_{Ri}, sk_{Ri}, a_{Ri},$

ω_{Ri}).

4.4.2 Encrypt Data

Let λ be a security parameter, G and G_T be two multiplicative groups of prime order p . Let $e: G \times G \rightarrow G_T$ be a symmetric bilinear map, the attribute universe $U = Z_p$. Each DO choose the random elements $g, u, h, w, v \in G$ and $\alpha \in Z_p$, and outputs his/her system public parameters $spp = (g, u, h, w, v, e(g, g)^\alpha)$, the master secret key $msk = \alpha$. The DO must secretly store msk and can publish spp on a public website that is accessible to all data requesters. We assume that the DO assigns each piece of data F_β to a corresponding access policy $(\mathbf{A}, \rho)_\beta$ ($Z_p^{l \times b}$, $\rho: [l] \rightarrow Z_p$), where the definition of this policy $(\mathbf{A}, \rho)_\beta$ is the same as access structures [10], and the matrix $\mathbf{A}$ is called the share-generation matrix for linear secret-sharing scheme [10] and has l rows and b columns. The function $\rho: \{1, 2, \dots, l\} \rightarrow U$, for each row $i = 1, 2, \dots, l$, the i^{th} row of $\mathbf{A}$ is labelled by an attribute $\rho(i)$.

Algorithm 1: Post a data sales message

Input: data owner O_j , data description D , DO's deposit $\text{Coins}(\omega_{O_j})$, data validity period t_{valid} , UEC address add_UEC_{O_j}

Output: $ROTC$ contract $ROTC_D$, update UEC_{O_j}

```

if  $O_j$  is unregistered then
  | go to terminal;
end
if  $\text{DepositLock}(pk_{O_j}, \text{Coins}(\pi_{O_j}), t_{\text{valid}})$  is not success then
  | go to terminal;
end
 $\text{MatchPolicy}(\cdot) \leftarrow \{(\mathbf{A}, \rho)_\beta\}$ ;  $\text{redeem}(\cdot) \leftarrow \text{false}$ ;
 $\text{DDeposit} \leftarrow \{D, \text{MatchPolicy}(\cdot), \text{redeem}(\cdot)\}$ ;
put  $\text{add\_UEC}_{O_j}$  into  $ROTC_D$ ;  $\text{UpdateUEC}(\text{UEC}_{O_j}, RTC_D)$ ;
terminal;
Return  $ROTC_D, \text{redeem}(\cdot)$ ;

```

The DO can assign each piece of data F_β to a corresponding access policy $(\mathbf{A}, \rho)_\beta$, which involves items such as data sharing scope, data price, user attributes and so on. For each piece of data F_β ($\beta = 1, 2, 3, \dots, n$), the DO randomly

chooses a session key K_β , and computes the data ciphertext $CT_\beta = Enc(K_\beta, F_\beta)$ by the symmetric encryption algorithm. Next, the DO chooses a random vector $\mathbf{y} = (s, y_2, y_3, \dots, y_b) \in Z_p^b$, where s is the random shared secret, and computes the share's vector $\lambda = (\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_l) = \mathbf{A}\mathbf{y}$. The DO randomly picks $t_1, t_2, \dots, t_l \in Z_p$, and calculates $C = K_\beta \cdot e(g, g)^{\alpha s}$, $C_0 = g^s$, and for each $\tau = 1$ to l , computes $C_{\tau,1} = w^{\lambda_\tau} v^{t_\tau}$, $C_{\tau,2} = (u^{p(\tau)} h)^{-t_\tau}$, $C_{\tau,3} = g^{t_\tau}$, and generates the session-key ciphertext $CT_{K_\beta} = ((\mathbf{A}, \rho)_\beta, C, C_0, \{C_{\tau,1}, C_{\tau,2}, C_{\tau,3}\}_{\tau \in [l]})$. The DO uploads the ciphertext $CT = \{(CT_{K_\beta}, CT_\beta)_{\beta=1,2,3,\dots,n}\}$ to the cloud servers, and records the ciphertext's location URL_{CT} returned by CSP. Finally, DO computes the file's hash values $\{H(F)_\beta\}$ and embeds the values $\{H(F)_\beta\}$ and URL_{CT} into the transaction TX_{CT} by using a *UEC* contract. When TX_{CT} is recorded in the Blockchain, the DO can use the *UEC* contract to perform the next data trade operations.

4.4.3 Data Sales

After encrypting his/her data, the data owner O_j can post a data sales message D to the blockchain network, $D = \{Coins(\pi_{O_j}), D_1, D_2, D_3, \dots, D_n\}$, where $D_\beta = \{s_{id}, H(F_\beta), Sig(H(F_\beta), sk_{O_j}), pk_{O_j}, KS_\beta, (\mathbf{A}, \rho)_\beta, Coins(r_w), t_{valid}, redeem()\}$, $\beta = 1, 2, 3, \dots, n$. Each piece of data F_β can be specified as an access policy $(\mathbf{A}, \rho)_\beta$ which means the data is only applicable to a certain range of DRs, *i. e.*, the DRs' attributes must satisfy the policy $(\mathbf{A}, \rho)_\beta$. In order to motivate the *DO* to provide high-quality data, the *DO* has to make a deposit $Coins(\pi_{O_j})$ when he/she posts a message D by following the scheme Π_{RR} , which is shown in section 4.2. Algorithm 1 illustrates the implementation of posting

a new sales message about the data.

Algorithm 2: Search data description

Input: data requester R_i , DR's deposit Coins
 $(d_1 + d_2)$, $ROTC_D$, UEC_{R_r} , requesterlist R_D^{list}

Output: update $ROTC_D$, UEC_{O_j} and UEC_{R_i}

```

if  $R_i$  is unregistered then
  | go to terminal;
end
if  $MatchPolicy(\{A(A, \rho)_\beta\}, \omega_{R_i})$  &  $SearchData(\{KS_\beta\}, KS_r)$  are
  dissatisfied then
  | go to terminal;
end
if  $Coins(d_1 + d_2) \neq 0$  &  $LockDeposit(pk_{R_i}, Coins(d_1 + d_2), t_{deadline})$  is
  successful then
  |  $R_i$  successfully deposit certain reward on the Blockchain ;
else
  | go to terminal;
end
 $\sigma_{R_i} \leftarrow$  signature on the data D with  $Sig(D, SK_{R_i})$ ;  $R_F^{list} \leftarrow \sigma_{R_i}$ 
 $RDeposit \leftarrow \{D, pk_{R_i}, \sigma_{R_i}, KS_{R_i}, \omega_{R_i}, t_{deadline}, t_{confirm}, Coins(d_1 +$ 
 $d_2), SearchData(.), EvaluateData(.)\}$ ;  $UpdateUEC(UEC_{R_i}, add_{ROTC_D})$ ;

terminal ;
Return  $ROTC_D, UEC_{R_i}$ ;

```

The data requester (DR) R_i retrieves the sales information in the UEC contracts of data owners, and finds many appropriate data owners only if he/she can satisfy the necessary condition functions $MatchPolicy(\{(\mathbf{A}, \rho)_\beta\}, \omega_{R_i})$ and $SearchData(\{KS_\beta\}, KS_r)$ with values $\omega_{R_i} \in (\mathbf{A}, \rho)_\beta$ and $KS_\beta \supseteq KS_r$. If R_i wants to buy the part data of the owner O_j , he/she can find his/her minimum attribute set ω_i that can satisfy all access policies $\{(\mathbf{A}, \rho)_\beta\}$ of the required data, and submits it to the DO. After O_j audits these attributes, they make an agreement. To avoid the DR's refusal to pay, he/she must make a deposit before obtaining the data. If the DR's deposit is $Coins(d_1 + d_2)$, we use the function $DepositLock(pk_{R_i}, Coins(d_1 + d_2), t_{deadline})$ to lock it on the Blockchain, where $t_{deadline}$ denotes the trading deadline. R_i signs the message D with sk_{R_i}

so as to ensure the correctness of data reward assignment.

4.4.4 Fine-grained authorisation

After the DR's attribute set $\omega_{Ri} = \{s_1, s_2, \dots, s_k\}$ is authenticated, the data owner (DO) O_j employs CP-ABE key-generation algorithm [10] to compute the attribute private key for the requester R_i . O_i picks $k+1$ random elements $r, r_1, r_2, \dots, r_k \in Z_p$, computes $K_0 = g^{\alpha} w^r$, $K_1 = g^r$, and for $\tau = 1$ to k , computes $K_{\tau,2} = g^{r_\tau}$, $K_{\tau,3} = (u^{S_\tau} h)^{r_\tau} v^{-r}$, and outputs the attribute private key $dk = (\omega, K_0, K_1, \{K_{\tau,2}, K_{\tau,3}\}_{\tau \in [k]})$. The DO computes the attribute private key ciphertext $CT_{dk} = Enc(pk_{Ri}, dk)$ and embeds CT_{dk} into the $ROTC_D$ contract. When the address of $ROTC_D$ will be recorded in the transaction index TX_{dk} on the Blockchain. The DO encrypts the transaction index TX_{dk} and the $ROTC_D$ contract address, ABI, source code by the DR's public key, and sends the ciphertext to the DR.

4.4.5 Decrypt

The authorised requester first reads the data CT_{dk} in the transaction TX_t on the Blockchain and computes his/her attribute private key $dk = Dec(sk_{Ri}, CT_{dk})$. He/she reads the metadata on the data D on the Blockchain and downloads the ciphertext $CT = \{(CT_{K\beta}, CT)_{\beta=1,2,3,\dots,n}\}$ from the cloud server according to the location URL_{CT} . Since the authorized DR's attributes can satisfy the access policies $\{(\mathbf{A}, \rho)_\beta\}$ of his/her purchase data, the DR computes the set of rows in $\mathbf{A}$ that can offer a share to attributes in ω_i , i.e., $= \Gamma = \{\tau : \rho(\tau) \in \omega_i\}$, then computes the constants $\{c_\tau \in Z_p$ such that $\sum_{\tau \in \Gamma} c_\tau A_\tau = (1, 0, \dots, 0)$, where A_τ is the τ -th row of the matrix $\mathbf{A}$. The DR uses the attribute private key $dk = (\omega, K_0, K_1, \{K_{\tau,2}, K_{\tau,3}\}_{\tau \in [k]})$ to decrypt the key ciphertext CT_K , and calculates

$$K_\beta = \frac{C \cdot \prod_{\tau \in \Gamma} (e(C_{\tau,1}, K_{i,1}) \cdot e(C_{\tau,2}, K_{i,2}) \cdot e(C_{\tau,3}, K_{i,3}))^{c_\tau}}{e(C_0, K_0)} \quad (4.1)$$

Finally, the DR uses the section key K_β to decrypt the data ciphertext CT_β , and calculates $F_\beta' = Dec(K_\beta, CT_\beta)$. For each piece of purchase data F_β , if $H(F_\beta') = H(F_\beta)$, the DR accepts the file F_β . Otherwise, the DR will terminate the data trading via the $ROTC_D$ contract. The DR obtain the purchase data set F_β .

4.4.6 Reward Assignment

Before the data requester R_i gets the purchase data, he/she sends the data evaluation function to the third-party entity, which gives the evaluation result of $EvaluateData(F_\beta)$. The evaluation result is recorded on the Blockchain. Next, the data owner (DO) O_j may demand the reward payment. As shown in Algorithm 3, the data reward is paid to the DO only if the evaluation result is “H”, and $t_{deadline}$ is this data trade deadline. The contracts UEC_{R_i} , UEC_{O_j} are updated automatically according to the evaluation result.

Algorithm 3: Evaluate data and pay the reward

Input: data owner O_j , $ROTC$ contract $ROTC_D$, the list of authorised DRs R_D^{list}

Output: update $ROTC_D, UEC_{O_j}, UEC_{R_i}, UEC_{O_j}$, pay a reward to the owner O_j

```

for each requestor  $\in R_D^{list}$  do
  if  $now \leq t_{deadline}$  then
    if  $Verify(\rho_{O_j}, pk_{O_j})$  is not successful then
      continue;
    end
    if  $Coins(\pi_{O_j} = 0)$  then
       $DepositLock(pk_{O_j}, Coins(\pi_{O_j}), t_{deadline} - now);$ 
    end
    if  $evaluateResult \equiv H$  then
       $Rewards \leftarrow Coins(r_w + pi_{O_j});$ 
    end
    if  $evaluateResult \equiv L$  then
       $Rewards \leftarrow Coins(pi_{O_j});$ 
    end
  end
   $Rewards \leftarrow 0;$ 
end
 $paymentsuccess \leftarrow$ 
   $payReward(pk_{O_j}, Reward); UpdateUEC(UEC_{O_j}, ROTC_D);$ 
Return  $ROTC_D, UEC_{R_i}, UEC_{O_j}, paymentsuccess;$ 

```

4.4.7 Confirm Transaction

The process of creating or updating a smart contract is viewed as a transaction which must be recorded on the Blockchain. To simulate the transaction confirmation and building blocks, we define the state change process as a pair (BC_i, BC_{i+1}) , where BC_{i+1} is the current block, and BC_i is its previous one. $BC_{i+1} = \{ block_{id}, timestamp, H(BC_i), add_{node}, LTX = (tx_0, tx_1, \dots, tx_k) \}$, where add_{node} is a blockchain node's address, LTX is a list of smart contracts which require to be confirmed by a majority of nodes of the blockchain network. All the contracts in LTX are immutably recorded on the Blockchain only if the block is on-chain. Any disputes will be handled by the **Dispute Contract** (DC).

4.5 Security Analysis

In this section, we analyse the security of our BC-FGA-DC scheme based on the system security threat model and security assumption in section 4.2.5. We prove that our scheme satisfies the following several security properties.

Theorem 1. Assume that the Blockchain provides secure consensus mechanism, the BC-FGA-DC scheme satisfies the correctness and fairness of data trades.

Proof. If the data owner (DO) O_j and the data requester (DR) R_i are assumed to be honest, and they normally perform data trading operations, then the evidences $TX_{CT} = (\{H(F_\beta)\}_\beta, URL_{CT}, Sig(\{H(F_\beta)\}_\beta, sk_{O_j}), pk_{O_j}, \{KS_\beta\}_\beta, \{(\mathbf{A}, \rho)_\beta\}_\beta, \{rw_\beta\}_\beta, t_{valid}, timestamp), TX_{dk} = (CT_{dk}, pk_{O_j}, Sig(dk, sk_{O_j}))$, the evaluation result of purchase data and the reward payment must be recorded in the blockchain.

If the blockchain provides secure consensus mechanism, O_j first posts the metadata about the data $\{H(F_\beta)\}_\beta$ and records the tuple $D = \{Coins(\pi_{O_j}), D_1, D_2, D_3, \dots, D_n\}$ into the blockchain via his/her RUC contract, where $D_\beta = \{s_{id}, H(F_\beta), Sig(H(F_\beta), sk_{O_j}), pk_{O_j}, KS_\beta, (\mathbf{A}, \rho)_\beta, Coins(r_w), t_{valid}, redeem(.)\}_\beta, \beta=1, 2, 3, \dots, n$. Suppose R_i uses the RUC contract to find that the part of data of the owner O_j can satisfy his/her request and his/her attributes can satisfy the access policies of the data, R_i can find the minimum attribute set ω_i that can satisfy all access policies $\{(\mathbf{A}, \rho)_\beta\}_\beta$ of the required data, and submits it to O_j . Before R_i gets the purchase data, he/she sends the data evaluation function to the third-party entity, which gives the evaluation result of $EvaluateData(\{F_\beta\}_\beta)$.

The evaluation result is recorded on the Blockchain. O_j computes the attribute private key dk for R_i by using the CP-ABE key generation algorithm, and generates the attribute key ciphertext $CT_{dk} = Enc(dk, pk_{Ri})$ and embeds CT_{dk} , pk_{Oj} , $Sig(dk, sk_{Oj})$ into the Blockchain as the transaction TX_{dk} . R_i can read the transactions TX_{CT} and TX_{dk} in the Blockchain, compute the attribute key $dk = Dec(CT_{dk}, sk_{Ri})$, and download the ciphertext $CT = (CT_K, CT)$ according to the location URL_{CT} , and verify two signatures with pk_{Oj} . If the verifications can pass, R_i performs the CP-ABE decryption algorithm to get the session key set $\{K_\beta\}$ and computes the data $F_\beta' = Dec(K_\beta, CT_\beta)$. If $H(F_\beta') = H(F_\beta)$, R_i accepts the data $\{F_\beta\}$. Otherwise, R_i will terminate the data trading through the $ROTC_D$ contract. If the data $\{F_\beta\}$ is high-quality, the reward $\{rw_\beta\}$ is paid to O_j . Therefore, if the Blockchain provides secure consensus mechanism, the evidence shows that the valid data trades must be recorded in the Blockchain.

Theorem 2. If a majority of third-party entities are honest, the BC-FGA-DC scheme can effectively resist the illegal data trades, which must be performed on this data crowdsourcing platform by malicious users.

Proof. In our scheme, there are several possible malicious activities of both data owner and data requester to maximise their own economic profits: (1) Data owners can potentially trade erroneous or corrupt data; (2) Malicious data requesters aim to obtain high-quality data without paying the rewards; (3) Malicious data requesters want to illegally resell data in a low price.

In the first case, as data owners encrypt their data, data requesters cannot fully know what the data/files contains. Data requesters can be at risk of obtaining data infected by a virus, which can infect other documents or download malicious payloads. Secondly, Data owners can also sell data that does not match their data descriptions or refuse to send the correct data after receiving payments. This malicious behaviour can easily occur when the amount of data is very large. To reduce the data owner's malicious behaviours, our protocol uses authorised third-party entities to verify the trading data and filter viruses. When the malicious activities of data owners are found, the authorised third-party entities also keep track of their actual identities. In addition, we introduce a time-locked deposit protocol to have a cool-off period in case of any corrupted data or missing data. In the dispute function of the protocol, multiple third-party entities will come together to match data content and description under a non-disclosure agreement. If data owners trade erroneous or corrupt data, this platform can give valid evidence and hold them accountable.

In the second case, malicious data requesters can deliberately deny they have got high-quality data. Our protocol uses the authorised third-party entities to

evaluate the trading data quality and record the result of evaluation into the Blockchain and uses a time-locked deposit protocol to lock their deposits before malicious data requesters obtain the trading data. If data requesters deny that they obtained high-quality data, our protocol may provide valid evidence to guarantee the rights of data owners and trace their accountabilities.

In the third case, malicious data requesters want to resell data at a low price illegally. In our blockchain protocol, there exists a risk in both unauthorised reselling of data and repackaged data. To resist these malicious activities, the origin of the trading data must be verified by the protocol of the platform, and the trading data can be verified by authorised third-party entities. Only data with non-suspicious origins can be traded. If data buyers are found to sell illegal data, the platform will blacklist the seller and terminate the transaction.

Therefore, if a majority of third-party entities are honest, the BC-FGA-DC scheme can effectively resist the illegal data trading, which must be performed on this data crowdsourcing platform by malicious users.

Theorem 3. Assume that Theorem 1 and Theorem 2 hold and all attackers have not enough coins, the BC-FGA-DC scheme can effectively withstand Single-Point failure, DDos, and Sybil attacks.

Proof. In our scheme, an attacker can bribe third-party entities or can destroy the nodes of Blockchain so that the normal data trades cannot be executed. All third-party entities in the permissioned Blockchain are interested in the long-term profits of maintaining the platform; their malicious attacks infect their stake in the platform. Thus the third-party entities do not destroy the platform. According to Theorem 2, the BC-FGA-DC protocol can effectively resist malicious users to execute the illegal data trades, then the fork chains of malicious activities are difficult to be generated. Since Blockchain has distributed characters, there exist many nodes to provide the data trade service for users. According to Theorem 1, even if only one blockchain node can normally work, the evidence of fair data trades must be recorded in the Blockchain, and users can still achieve the data trade service. Thus there is not the Single-Point failure in our protocol.

In the blockchain network, nodes need to maintain the normal operation of the network, users (DRs or DOs) have to pay transaction fees to nodes. Our protocol requires users to make deposits before executing data trades. When malicious attackers launch the attacks, they need to pay a lot of expenses. Therefore, our protocol can resist the DDos and Sybil attackers who have not enough coins.

4.6 Performance

In this section, we discuss our implementation results and performance analysis. Ethereum is our blockchain testing platform as Ethereum is an opened Blockchain with large number users and well developed Turning complete smart contract environment. Gas price in Ethereum refers to the amount of Ether one is willing to pay for each unit of gas. Each execution of smart contracts requires a certain amount of gas. In Ethereum, the execution cost can be divided into two areas, storage costs, and computation cost. Users can increase the computation speed of transaction confirmation by increasing the gas price. The storage cost in Ethereum depends on the size of transaction inputs and is calculated based on each bit. A 256-bit input will cost 20k gas storing at the slowest speed of 3 gwei per gas will cost user approx. 0.005USD in 2018, increasing the speed to the maximum for the same data will cost 0.045USD at 25gwei per gas. The difference in the meantime for the transactions to be confirmed using 3gwei per gas is 3254 secs, and the meantime for the transactions using 25gwei per gas is 29 secs, depending on the transaction queue. The actual computation speed and compiling of typical smart contracts as a comparison is about 100ms, while extremely complex smart contracts used for DDOS attacks take 2 secs. Moreover, due to the confirmation time required, we used a private server (Ganache) to test run our smart contract protocols to simulate the confirmation of transactions.

For any transaction, there is a base cost of 21000 gas. Any interaction, such as compile and calling of functions, will incur additional costs. In the conversion to the actual price in USD, we assume a gwei cost of 10 per gas. Thus, the meantime to execute per function in real life is about 30 secs. To increase users' privacy protection, we need to include in a shared key for the data owner and data requester. To read any data regarding the transactions, users must input the critical information used to enter the data into the blockchain by the meta module. The Average costs of smart contracts are shown in Table 4.1.

Table 4.1: The costs of smart contracts in BC-FGA-DCrowd

Contract	Number of functions	Total compiling cost	Cost in USD
<i>RUC</i>	2	328695	0.28
<i>UEC</i>	7	686035	0.59
<i>ROTC</i>	5	1347505	1.30
<i>DC</i>	4	358446	0.35

For the *RUC* contract, we assume there are three attributes and five keywords with byte32 input format for the keywords. We also assume each keyword have 15 characters. The attributes are 171bit integers saved as 32bytes for ease of future recalls. For a total of 13 of inputs and the contract costs 347795 gas, i.e., 0.30USD. After the *RUC* is published in the blockchain, the metadata module saves the contract address of the smart contract and the data stored in the contracts. Data requesters can use the meta module to search and match possible data sources from different data owners using the keywords and attributes.

Once a match is found, the metadata module will start to generate the *UEC* Contract. *MatchPolicy(.)* function takes the transaction hash address of the matched *RUC*s functions from both the data owner and data requester that matched and use them as relevant inputs from the application layer. Both the data owner and data requester can validate each other's attributes and keywords before proceeding to the next trading contract. The *Deposit(.)* function obtain the deposit from the data owner and data requesters and locks the deposit until the trade is completed. The off-chain meta module will use the transaction address of the Deposit function to generate *ROTC*. Once the trade is completed, the meta module will return the deposits with the Release() function. The total cost for executing the functions is 553 322 gas or 0.48 USD.

The *ROTC* contract initiate the trade function after storing the transaction hash of the *Deposit(.)* function. The price of the data is also inputted. While the funds are locked, DO has to input in the URL of the encrypted file and the decryption key. When the data requestor receives the data, the requestor inputs a boolean input to denote the quality of data. In the *ROTC* contract, the size of CT_{dk} and, we assume the URL to be 45 characters. The total execution and storage cost for ROTC is 446 881 gas or 0.39 USD.

To handle any disputes, the meta layer creates a Dispute contract. The meta layer will randomly select from a pool of juries with the weight of their votes based on a reputation system. Part of the deposit will be used to handle such disputes and used as monetary incentives for the juries. The smart contract execution cost increases with the number of juries linearly. Each jury requires 43 660 gas (0.04 USD) to be assigned any voting rights and 22 345(0.03 USD) gas to vote. The juries have to select from a rating of 0 – 10 on how much the data that is provided by the data owner matches his/her keywords as well the quality of data.

(a) Transaction cost

(b) Time cost

(c) Transaction cost related to attribute number

Figure 4.3: Cost and time requirements of the smart contracts

The cost of running data trading grows linearly with respect to the number of data trades. The total cost of a trade is 3.34 USD and 334 USD for 100 trades, as is shown in Fig.4.3(a). We also investigated the total cost for running a large number of data trades linearly and concurrently, as is shown in Fig.4.3(b).

The computational time by the nodes is negligible compared to the transaction posting time. We set the block creation time in Ganache to 30 s to simulate real-life performance. The total base runtime for running *RUC*, *UEC*, and *ROTC* is 240 s. In our protocol, some of the functions in the smart contract can run concurrently, while the smart contracts must be executed in an ordered arrangement due to their dependency. For comparison, we ran the contracts from *RUC* to *UEC*, followed by *ROTC* before running the next contract. The total time required is 26 880 s. We then simulate the concurrent running of the contracts, by running *RUC* and *UEC*. For *ROTC*, the generation of the key ciphertext is not concurrently to prevent any duplication of keys. New keys will only be generated after existing keys has been uploaded to the blockchain. This is an improvement as the total time required is 4050 s. In addition, to increase the number of groups allowed in the data trade, data owners can increase the number of attributes assigned to data. But increase the number of attributes, the price of execution also increases. As the attribute sizes are constant, at 171 bits, the total price of running the constants increase linearly, as is shown in Fig.4.3(c). However, the total price is still relatively low, with 10 attributes input costing 1.2% more.

In conclusion, the Average transaction cost for a single data trade is 3.34 USD, and using mean transaction time of network conditions, the total time required for such a trade is approx. 240s. In our system, users can fine-tune the number of attributes and security features based on the sensitivity of the data and user requirements. There is no worse case execution time available due to the nature of P2P broadcasting of transactions. It is possible for transactions to not be input into the blockchain due to failure of the nodes in meeting consensus.

4.7 Conclusion

In this work, we propose and implement a blockchain-based mechanism for fine-grained authorization in data crowdsourcing (BC-FGA-DCrowd). In the BC-FGA-DCrowd, we use public blockchains to handle data trading incentives as

cryptocurrencies and payment services while providing a high level of flexibility for users. Data owners can employ ciphertext-policy attribute-based encryption (CP-ABE) to pre-process the complex encryption workload, and generate the attribute private key for data requester to achieve the fine-grained authorization of data trading. Based on reasonable assumptions, we prove the BC-FGA-DCrowd scheme can effectively resist the malicious activities of internal users and external DDos and Sybil attackers. We conduct our tests on a private Ethereum network using Ganache with a local host. In our future work, we wish to research data evaluation function that can evaluate the quality of ciphertext data.

Chapter 5

CP-ABE in a permissioned blockchain

5.1 Implementation

In this chapter, a Proof of concept (POC) inventory managements system is proposed and implemented. Firstly, the key user archetypes are analysed to provide key functional specifications and requirements for the blockchain. Secondly, the trust models and security assumptions are studied, and an overview of the architectural design is provided. Lastly, the schematics of the flow of interactions between the blockchain, applications, and users will be given based on the trust models and security assumption.

5.2 System model

5.2.1 Users Archetypes

The platform is designed to contain the following stakeholders and users: Cotton farmers, yarn spinner, fabric mills, Garment manufacturer, end consumer and certification agencies.

- Cotton Farmers(CF): Cotton farmers are the start of the supply chain. In our POC, we assume these cotton farmers are small independent farmers with small profit margins. These farmers have little to no capital and are less likely to high technical expertise and equipment. Their main objective in using the platform will be to ensure they receive a fair share

of the supply chain's profits. It can be assumed that they will access the platform with low computational power devices such as mobile phones.

- Yarnspinner(YS): Yarnspinner purchase cotton from farmers to spin cotton into yarns. They are the second stage of the supply chain. In general, yarn spinners companies have more capital than farmers and thus have access to higher computational devices such as personal computers or desktops.
- Fabric mills(FM): In the case of premium fabrics mills, fabric mills such as Holland & Sherry or Albini Group are large companies with huge capital and advanced manufacturing processes. These companies are interested in providing the provenance of their products, proving their authenticity.
- Garment Manufacturer(GM): Garment manufacturer can range from smaller ateliers to large clothing companies. Their core interest will be proving the origin of the raw material they have used. Fabric mills. In higher-end garments like suits or jackets, the fabric mills' labels are often sewed on the underside the jackets to showcase the heritage of the fabric.
- End consumers: Consumers are interested to know the origin of the product.
- Certification agency(CA): The purpose of the certification agency is to regulation processes in the supply chain. These include the treatment of workers, environmental regulations and fair trade for farmers. Products that were certified by them, often carry their logos.

5.2.2 Information and documentation flow and access control

All the essential documentation and data transfers between various platform parties are charted out in Table A.1. The table also contains the data source and the targeted receiver and various access rights to the information. For example, vital production information of fabrics is only available to certifying agencies in the platform. Thus, this information has to be encrypted and can only be viewed by the certification authority. Information such as certification credentials that are meant for the general public does not need to have privacy protection mechanisms.

Figure 5.1: Overview of the architecture layers of the cotton supply chain blockchain protocol

5.3 Blockchain architecture

The blockchain platform consists of four main layers as shown in Fig.5.1, blockchain records, middle application server, cloud server, user interface. The blockchain records layer consist of the blockchain implementations and basic blockchain network protocol. The middle application server contains encryption modules, general uploading and downloading of data, consensus protocol, user account creation and management, and applications. The cloud server stores important documents and other information. Lastly, all the stakeholders will interact with the platform using the user interface.

Transactions Structures

The format of the transaction records in the supply chain blockchain is modified to fulfil the supply chain's needs and requirements. New sets of transaction inputs are added to facilitate the transfer of various products from one stakeholder to the other while keeping some of the existing inputs and outputs. All the transactions follow the data structure to improve performance. All transactions consisted of the following attributes:

1. Sender: The sender of product or transaction generator
2. Receiver: The receiver of the product or document
3. ProductID: The ID of the product
4. Quantity: The amount of product in the transaction.
5. : URI: The URI of any document or files shared during the transaction
6. : MetaData: The hash of the documents or file. Additional comments can also be added if needed.
7. : Signature: The signature of the sender generated from the private key of the sender.

Blocks

Each block in the chain will store the set of transactions verified in that specific epoch. The block consists of 3 sections, header, transactions and metadata. The header consists of the block ID, hash of the previous block, while the metadata consists of the date and time the block is generated, hash of the current block, the public key of the signer and signature of the signer/approval. The transaction layer contains all the transactions made and queued since the publishing of the previous block. The blocks will list and ranks the transactions according to the time when it receives the transactions. The blocks are linked by hashes based on the Merkle tree structure. In the first block, the transaction list is replaced by a secret text. Lastly, the first block will contain the public key used for the ABE in the encryption protocol.

Nodes

Each blockchain nodes will contain a copy of the distributed ledger with the same number of blocks. The network is designed to focus on Consistency and Availability of the CAP theorem. Our protocol using a round-robin proof of authority model. The protocol verifies the identity of the block miner using the algorithm 5. Nodes are added to the network using IPs and follow a publish and subscribe messaging model. Only authorised stakeholder's node will be added to the platform. Once added to the list of nodes, a secure communication channel using SSL is used. New transactions are broadcast to every node in the list of IPs whenever they are created. Our protocol assumes that the farmers do not own any nodes due to the cost involved. Thus in our platform, there

is a minimum of 4 nodes, assuming there is only one company taking each stakeholder's role. Each node will be allocated a specific time frame to create blocks according to the consensus protocol.

5.3.1 Middle application server

The middle application server contains the majority of the vital function and features of the protocol. To allow the blockchain platform to remain streamline and efficient, non operationally necessary information is stored in the cloud servers. The blockchain cannot hold and manage the vast amount of data that the platform requires. This cloud server needs to store encrypted documents and other photo-graphical evidence as well as useful information. The information in the cloud can be divided into three privacy categories or levels, public, privileged and private. Public information is available for non-registered blockchain users, while privileged information is view-able by authorised users in the supply chain protocol. Lastly, private information is only viewable if the user has access to the private key used in the cryptographic protocol.

Encryption

The middle layer implements CP-ABE to lock key documents to protect the privacy of the stakeholders. The implementation details are in section 5.4.

Data storage layer

Data storage layer or cloud storage servers has the same function as in chapter 4. However, in this section, the CP-ABE protocol has been improved to allow for multiply Attribute Authority.

Application layer

In our framework, all functions are built into the blockchain and web applications. Thus, the blockchain users can deploy the allowed APIs without joining the platform and the blockchain platform to deploy better access control measures. Some of the more important APIs is listed below

1. **Verifying transaction** : The verifying transaction protocol checks if the quantity of the product and the sender's signature is correct. The total quantity is computed by summing all the product quantity with the

same productID in the blockchain and transactions that are queue in the waiting list. Once verified, the transaction is broadcast to other nodes. See Appendix(4)

2. **Verifying Blocks :** Verify the signature and identity of the user that publish the block. See Appendix(5)
3. **Create User :** Only admin accounts can create or add a new account to the database. New accounts are given the lowest level of access control by default. Two sets of private key $pK_{private1}, pK_{private2}$ and public key $pK_{public1}, pK_{public2}$ are generated using Elliptic curve cryptography. $pK_{private1}$ and $pK_{public1}$ are used by the user to make and sign transactions. $pK_{private2}$ and $pK_{public2}$ are shared among the administrator of the company as well as the user. The API also generate a new secret key SK_a tied to the $pK_{public1}$. Thus the API returns the following tuples $pK_{private2}, pK_{public2}$ to administrator $pK_{private1}, pK_{public1}, pK_{private2}, pK_{public2}, SK_a$.
4. **Promote User :** Only admin accounts can promote accounts in the database. Node administrator cannot grant access right higher or equal to themselves.
5. **Make Transactions :** Users have to log in make transaction in the blockchain. User has to have the more or equal quantity of the product in their wallet to trade. The wallet will be consist of open transactions and transactions in the blockchain to prevent double-spending. For instance, if the user has 10 of 'product001' in the wallet, and he made a transaction, sending Qty 7 of 'product001' to another wallet, he cannot send any Qty more then 3 of 'product001', even though the previous transaction might not have been mined into a block. The sender can also upload URI and document hashes as metadata in the transaction as a record.

5.3.2 Concrete Supply chain Framework

1. The farmer registers his/her farm on the blockchain protocol and details of his farm. This includes the location and size of his farm, number of workers and maximum production capability. This information will be stored in the Cloud storage and open to the public to increase transparency and reduce fraud. Only authorised parties will be allowed to view the information in the cloud. The data is sent to the certifying agency.

2. The farmer plants the cotton crops and records the production date and amount on the blockchain. In addition, he/she has to take photographic evidence of the activity and upload them using the application layer. The URL and hash of this documented evidence will be inputted into the transaction.
3. After harvesting, farmers will record the end production date and quantity on the blockchain. Each batch of cotton production will have a unique product code and signature. The mapping of the hash of the product will be mapped to the farmer in the cloud storage. This is to allow for ease of origin tracing by the end consumer.
4. When cotton merchants or yarn spinner purchase cotton from cotton farmers using the platform, they will have to provide a copy of purchase order while the farmer provides a copy of the invoice. These documents will be encrypted and stored in the cloud. The hashes and URI of the document will be recorded in the blockchain. The blockchain will record the transfer of cotton from the inventory of the farmer to the yarn spinners.
5. During the production process, the yarn spinner will deplete his/her quantity of cotton in the inventory and create a new product transaction with a new set of product ID and signature. This new product transaction will contain the ID and hash of its cotton components, i.e. cotton from different farmers. Product information can be uploaded to the cloud to provide end-consumer with more information with the product's characteristics.
6. Fabric mills that purchase yarns from yarn spinner using the platform will similarly have to provide a copy of purchase order while the yarn spinner provide a copy of the invoice. These documents will also be encrypted and stored in the cloud. The hashes and URI of the document will be recorded in the blockchain. The yarns will be transferred to the inventory of the yarn spinner to the fabric mill.
7. The fabric mill will deplete his/her quantity of cotton and create a new product transaction. This new product transaction will contain the ID and hash of its yarn components. Product information can be uploaded to the cloud to provide end-consumer with more information with the fabric's characteristics.
8. When garment manufacturer purchases yarns from fabric mills using the platform, they will similarly have to provide a copy of purchase order while the fabric mills provide a copy of the invoice. These documents

will also be encrypted and stored in the cloud. The hashes and URI of the document will be recorded in the blockchain. The fabric bolts will be transferred to the inventory of the fabric mills to the manufacturer.

9. The garment manufacturer will record the hashes of the fabric bolts used to make their garments. Each piece of garment will have a unique QR code to prevent counterfeiting.
10. Once any retailer purchase garments from the manufacturer, the end product will be transfer to the inventory of the fabric mills to the manufacturer.
11. Any end consumer that wishes to purchase the end product can scan the QR code to retrieve information regarding the product. After the purchase, the end-product will be removed from the retailer's inventory.

5.4 CP-ABE Implementation

5.4.1 Attributes and access policy

The platform uses the new CP-ABE [32] scheme to encrypt vital information that is to be shared on the platform. The new scheme allows multiparty stakeholders to be the attribute authority (AA), and issue new attributes to its staff. Also, the attribute authority can revoke its staff's access rights when the staff leaves the company without the need to encrypt the data again. The primary objective is to allow each stakeholder to share information with the relevant parties securely without multiple encryptions or encrypt with multiple public keys. In addition, the same plain-text does not need to be encrypted again when a new user joins into the platform. The attributes of each stakeholder (Fig.5.2) in the blockchain are defined, and the access policies are then set for each encryption.

5.4.2 Roles definition

1. Blockchain (BC): Blockchain is the platform where the users interact and share information.
2. Root authority(RA): The root authority setups the platform and generates the master key. It is assumed that the root authority can be trusted.

Attribute tree for cotton supply chain

Figure 5.2: Overview of stakeholders' attribute of the cotton supply chain blockchain protocol

3. Attribute Authority(AA): The Attribute Authority creates the attributes keys of all the users under its domain. Each stakeholder is only allowed to issue attribute keys to its employees.
4. Data owner(DO): The data owner is the user who generates or owns the data. He/she sets the access policy of his data and encrypt his data before sharing with the relevant party.
5. Data requestor(DR): Each user has his/her own set of attributes provided by the attribute managers. The Data receiver will only be able to decrypt the data if his attributes matched the data owner's access policy.

5.4.3 Integration of CP-ABE with blockchain

In the original implemented protocol [32], the author mainly focused on the protocol design on access right distribution and relied on RSA public key schemes to revoke users. The author's protocol does not support any transaction signing. In our implementation, we have the CP-ABE protocol is integrated with existing elliptic curve cryptography(ECC) of the blockchain. This allows the platform to be more streamline and secure. The new implementation is presented below.

1. Setup: In this stage, the RA generates the master key mK and public key pK . The public key is registered in the blockchain's genesis block and is accessible to every user in the blockchain. The blockchain administrator keeps the master key.
2. Create Authority (mK): The Create Authority Algorithm takes in the master key mK as an input and generates a secret key sK_{AA} to every AA in the platform. AA is the system administrator of the company's stakeholders and is in charge of all the DR and DO in its domain.
3. Create user: new user accounts can be created using the create User API.
4. Create Attributes: The function allows the Attribute Authority to create new attributes and attribute keys. This attribute mainly comes in the form of company details and employee positions. This allows the stakeholder to control the data's access based on specific departments in the company if required.
5. IssueAttribute: The Attributes are issued internally within the organisation to the DR by the AA in the company.

6. RevokeAttribute: The Attribute Authority can revoke its employee's attribute using $pK_{private2}, pK_{public2}$. Thus the stakeholder is in charge of the behaviour of its employees.

5.5 Experimental results

The blockchain platform is written in Python 3.7, and the blockchain server is implemented using Python-Flask. We use the following package to The experiment is conducted with an Intel(R)Core(TM) i7-4770 CPU with 3.4GHZ CPU and 32GB memory of RAM. The blockchain network is simulated with three nodes connected locally via LAN using TCP stack. All APIs and simulations are performed using POSTMAN HTTPS.

5.5.1 Database size

In the experiment, we generated blocks of ten transactions each and investigated how does the database size increase with the number of blocks. The transaction consists of product creation algorithm and product transfer algorithms. All the blocks are saved in JSON files to allow for faster access and more efficient storage and loading. The main benefit of such a method is that it is fast and human-readable. The files can be further encrypted to improve the security of the blockchain.

The size of the database scaling linearly with the number of blocks (Fig.5.3). After 10000 blocks of data, the total size of the database is 98.6MB. This includes the hashes of the blocks and the transactions public keys which are 32 bytes per signature and hash.

5.5.2 Speed of transactions

The next area explored was the speed of the transaction generated. To determine the time require for the transactions, the transactions are divided into two categories, blockchain transactions and non-blockchain transactions. For non-blockchain transactions, the platform does not need to search through the records in the blockchain. The time requires to perform these transactions are constant at 3ms. This is expected since the blockchain does not need to reference back to its database and the only computation required is signature verification.

Figure 5.3: Plot of the linear relationship between the size of the database with the number of blocks

Figure 5.4: The time required to create transactions scale linearly with the number of transactions

The next experiment consists of determining how does the number of blockchain transactions affects the speed of transactions. Each transaction generated requires the blockchain to verify the content of the transaction against its ledger. Specifically, the blockchain will verify the product ID, product quantity and the public keys of the sender. Our results show that the time required for each transaction increase linearly with the number of transactions in the blockchain ledger (Fig.5.4). The time required for the 3001st transaction is 732 ms.

5.6 Security analysis

In the protocol, we based the threat model on the following threat model:

1. **Trusted permissioned network**

In the protocol, we assume most users are trusted, and insider attacks are not likely. However, external adversaries may attempt to sniff communications, impersonating as staff of the stakeholders or perform Denial of services

2. **Access control and key management**

In the protocol, we assume that each stakeholder has a secure key management system to prevent private keys' leakage. In addition, all the private keys of the stakeholders are only issued to verified employees.

3. **Secure communication channels**

The protocol relies on the stakeholders to have secure communication channels and distribute information and private keys via these channels.

5.6.1 Security threats

Double spending attacks Double-spending attack refers to the form of attack where adversaries send conflicting transactions to spend the same funds. The goal of the adversary is to use his/her inventory twice. In the case of our permissioned blockchain, this double-spending attack may occur when stakeholders generate multiple transactions with conflicting information and break the consistency of blockchain. For example, the fabric mill may send the same batch of product to multiple garment producers. This typically occurs when the transaction is still pending, i.e. not added to the blockchain. While there is no financial transaction involved, this leads to data inconsistencies and disputes. In our platform, the product quantity is locked by the system when

a transaction is made. This prevents double spending from occurring. If the stakeholder made any transactions with the same product batch and invalid product quantity, Algorithm 4 will reject the transaction.

Sybil attacks Sybil attacks are not likely to occur in the platform, as the consensus protocol does not work based on majority voting. In the platform, proof of Authority will validate the identity of the block generator. After verification, the block will be 'mined'. Secondly, all the transactions have metadata such as document checksum as well as documentation evidence. No voting is required in any transaction. These make the platform Sybil attacks proof.

Flood attacks and DoS Flood attacks can occur when malicious adversary intercept any communication channel and flood the platform with transactions. This leads to many transactions in the memory pool of the nodes and larger block size. However, the huge block size will lead to problems in relaying. Moreover, adversaries can also perform Denial of services attacks(DoS) on the platform by overloading the nodes with transactions and bring down the network. In our platform, all transactions are made in a separated web interface. Once adversaries attack the interface, it can decouple itself from the network. If any node bought down by any DoS, the system administrator could decouple fault nodes efficiently.

Chapter 6

Planning and Future research

6.1 Conclusion

The globalised supply chains have connected companies and small players around the world in a complex network. This complex network which involves the movement and management of information and goods requires data integration and transparency to improve its efficiency and effectiveness. Each supply chain has its specific security and privacy requirements due to different operational or regulatory needs and processes. Thus, every blockchain protocol and platform have to be specifically designed and optimised for its purpose.

This thesis presented a permissioned blockchain framework for apparel logistics supply chain based on the requirement studies performed. The framework improves the supply chain's transparency and data traceability through immutable ledgers' records in the blockchain.

The new permissioned blockchain architecture consists of a new communication protocol, privacy protection mechanisms, and a series of functions to support the apparel supply chain's operations. Under the new blockchain protocol, business owners can adopt access controls to the database and APIs for its employees, similar to existing supply chain management platforms.

The thesis introduces a novel approach to perform privacy-protected data sharing using blockchain. In this approach, users can perform fine grained authorisation of the access rights to their data. Data owners can issue and revoke new access rights to their data without the need to perform multiple

encryptions. In particular, a new blockchain protocol is designed to support the fair trade of data between data owners and requesters. Experimental data shows that the number of functions required is lesser than traditional methods. The security analysis has shown that the new solution can help improve data trades' correctness and fairness. Simultaneously, the new protocol can also resist illegal data trades and effectively withstand Single-Point failure, DDos, and Sybil attacks. Our implementation results prove the viability of using CP-ABE on a public blockchain.

In conclusion, the main contribution of the paper can be summarised below:

1. A new permissioned blockchain protocol is designed to cater to the apparel supply chains' requirements.
2. Design new Fine-grain authorisation mechanisms in blockchain platforms using CP-ABE.
3. Implemented the new permissioned blockchain with CP-ABE for proof of concepts.

6.2 Future works

For the master thesis goal, future works will be mainly focused on improvements in the logistics supply chain blockchain platform. Namely, the improvements will come in three main areas.

1. The platform should include a hash-table to track and apply pointers to the transactions. There is currently no bound on time required to generate the transactions. In the ideal case, the blockchain should keep track of the relevant transactions such that it does not need to search through the entire blockchain for transactions with the same product ID or code.
2. The blockchain should have more secure communication channels for key distribution. As the platform is proof of concern, many security standards are not implemented.
3. The master key used in CP-ABE should be generated by the stakeholder using multi-party computation. Using MPC, entities in the platform reduces the need to trust the administrator of the platform.

Appendix A

Appendix

Algorithm 4: Verify transactions

Input: Sender's Address Add_{sender} , Sender's signature Sig_{sender} , Open transactions list $TXNs_{open}$, Blockchain BC_{list} , product quantity Qty_{send}

Output: True, False

Inventory retrieval:

Totalasset = 0 ;

```
for each transaction Txn in  $TXNs_{open}$  do
    if transaction.sender = sender's address then
        Totalasset = Totalasset.append(Txn);
    end
end
```

```
for each Block Blk in  $BC_{list}$  do
    for each transaction TXN in Blk do
        Totalasset = Totalasset.append(TXN);
    end
end
```

Quantity verification:

bool quantity check;

Totalproductamount = 0;

```
for each transaction Txn in Totalasset do
    if transaction.productID = productID then
        Totalproductamount = Totalproductamount +
            transaction.quantity
    end
end
```

```
if Totalproductamount =  $Qty_{send}$  then
    return true;
end
```

sig verification:

Algorithm 5: Algorithm to Verify Blocks

Input: Publisher's Address $Add_{Publisher}$, Publisher's signature $Sig_{Publisher}$, Current Hash $Hash_{current}$

Output: True, False

```
    bool hash check;  
    Previoushash = Hash(blockchain);  
    if  $Hash_{current} = Previoushash$  then  
         $\_$  return true;  
    bool signature check;  
    if  $signature = Sig_{sender}$  then  
         $\_$  return true;  
    if hash check  $\&\&$  sig verification then  
         $\_$  return true;
```

Table A.1: Table of information access rights

List of Information and ownership			
Type of information	Data Generator	Data receiver	Allowed viewership
Farm Location, Production Details	Farmer		Rest of the users in the blockchain
Purchase order for cotton	Yarn spinner	Farmer	
Sales Order of Cotton	Farmer	Yarn Spinner	Certification Agencies
Inventory transfer from Farmer to Yarn Spinner	Farmer, Yarn spinner	Certification Agencies	Rest of the users in the blockchain
Yarn production details	Yarn Spinner	Certification Agencies	
Purchase order for Yarn	Fabric Mill	Yarn spinner	
Sales Order of Yarn	Yarn Spinner	Fabric Mill	Certification Agencies
Inventory transfer from Yarn Spinner to fabric mill	Fabric Mill, Yarn spinner	Certification Agencies	Rest of the users in the blockchain
Fabric production details	Yarn Spinner	Certification Agencies	
Purchase order for Fabric	Garment manufacturers	Fabric mills	
Sales Order of Fabric	Fabric mills	Garment manufacturers	Certification Agencies
Inventory transfer from Fabric mill to Garment manufacturers	Fabric Mill, Garment manufacturers	Certification Agencies	Rest of the users in the blockchain
Garment production details	Garment manufacturers	Certification Agencies	
Product certifications	Certification Agencies	Relevant parties	Rest of the users in the blockchain

Bibliography

- [1] Jan Falkowski, C Gaudreau Ménard, Richard J. Sexton, Johan F.M. Swinnen, and Senne Vandevelde. Unfair trading practices in the food supply chain: A literature review on methodologies, impacts and regulatory aspects. 2017.
- [2] The Irish Times. Unfair trade keeps developing farmers in poverty, Dec 2005.
- [3] Kate Hodal. Abuse is daily reality for female garment workers for gap and h&m, says report, Jun 2018.
- [4] Mahmoud Al-Odeh and Safwan Altarazi. Supply chain management information systems survey for jordanian companies. *International Journal of Mathematical, Engineering and Management Sciences*, 4:567–579, 06 2019.
- [5] Kate Taylorc. Egyptian cotton sheets might not be what you think they are. August 29, 2016.
- [6] Annie Kelly. Lawyers challenge uk imports of 'slavery-tainted' uzbek cotton. May 21, 2019.
- [7] Bitcoin: A peer-to-peer electronic cash system, Aug 2008.
- [8] Nancy Lynch Seth Gilbert. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, June 2002.
- [9] Di Matteo Caldarelli G. Pappalardo, G. Blockchain inefficiency in the bitcoin peers network. *EPJ Data Science*, 7, September 2018.
- [10] Roeschlin M. Scherer T. Capkun S. Androulaki E., Karame G.O. Evaluating user privacy in bitcoin. *Financial Cryptography and Data Security*, 7859:34–51, FC 2013.

- [11] Michael Lesk and Brian Kernighan. Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, page 173–186, New Orleans, Louisiana, USA, 1999.
- [12] Chris Baraniuk. Bitcoin’s energy consumption ’equals that of switzerland’, Jul 2019.
- [13] Nic Niedermowwe. Scaling bitcoin to millions of transactions per second, May 2019.
- [14] The zilliqa technical whitepaper, Aug 2017.
- [15] Introduction to sawtooth pbft, Feb 2019.
- [16] A. Sahai and B. Waters. Fuzzy identity-based encryption. page 457–473, May 2005.
- [17] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy (SP ’07)*, pages 321–334, 2007.
- [18] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS ’06*, page 89–98, New York, NY, USA, 2006. Association for Computing Machinery.
- [19] Eran Toch. Crowdsourcing privacy preferences in context-aware applications. *Personal Ubiquitous Comput.*, 18(1):129–141, January 2014.
- [20] Md. Nahian Shaikat. Supply chain management in apparel industry. 10 2018.
- [21] Md. Rifaul Islam Avizit Basak, M.M.Israfil Shahin Seddiqe and Md. Omar Faruk Akanda. Supply chain management in garments industry. *Global Journal of Management and Business Research: A Administration and Management*, 14, 2014.
- [22] W. Douglas Cooper. Textile and apparel supply chains for 21st century. *Journal of textile and apparel Technology and Management*, 6, Fall 2010.
- [23] Himanshu Aggarwal SAshish Seth, Ashim Raj Singla. Service oriented architecture adoption trends: A critical survey. pages 164–175, IC3 2012.

- [24] Horst Treiblmaier. The impact of the blockchain on the supply chain: a theory-based research framework and a call for action. *Supply Chain Management*, 23, 10 September 2018.
- [25] DHL Trend Research. Blockchain in logistics. 2018.
- [26] Accenture plc. Editing the uneditable blockchain: Why distributed ledger technology must adapt to an imperfect world. 2017.
- [27] Jing Linbo Kaijun Leng, Bi Ya, Fu Han-Chi, and Inneke Van Nieuwenhuyse. Research on agricultural supply chain system with double chain architecture based on blockchain technology. *Future Generation Computer Systems*, 86:641–649, September 2018.
- [28] A. Kumar, D. Choudhary, M. S. Raju, D. K. Chaudhary, and R. K. Sagar. Combating counterfeit drugs: A quantitative analysis on cracking down the fake drug industry by using blockchain technology. pages 174–178, 2019.
- [29] Bill Wang Ray Y Zhong Abraham Zhang V.G.Venkatesha, Kai Kang. System architecture for blockchain based transparency of supply chain social sustainability. *Robotics and Computer-Integrated Manufacturing*, 63, June 2020.
- [30] Qi Feng, Debiao He, Sherali Zeadally, Khurram Khan, and Neeraj Kumar. A survey on privacy protection in blockchain system. *Journal of Network and Computer Applications*, 126, 11 2018.
- [31] Zhu LH. Shen M. et al Zheng, BK. Scalable and privacy-preserving data sharing based on blockchain. *Comput. Sci. Technol.*, 33:557–567, 2018.
- [32] Mark Gall Georg Bramm and Julian Schutte. Blockchain-based distributed attribute based encryption. pages 99–110, ICETE 2018.