國立臺灣大學電機資訊學院資訊工程學系
碩士論文
Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University
Master Thesis

基於區塊鏈之工廠生產紀錄與產品履歷技術設計與實作

Design and Implementation of Distributed Traceability System for Smart Factories based on Blockchain Technology

楊凱崴
Kai-Wei Yang

指導教授：施吉昇 博士
Advisor: Chi-Sheng Shih, Ph.D.

中華民國 107 年 8 月
August, 2018

doi:10.6342/NTU201802852

國立臺灣大學碩士學位論文
口試委員會審定書

基於區塊鏈之工廠生產紀錄與產品履歷技術設計與實作

Design and Implementation of Distributed Traceability System for Smart Factories based on Blockchain Technology

本論文係楊凱崴君（學號 R05922138）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 107 年 7 月 31 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

（指導教授）

系 主 任

致謝

這一路走來，我非常感謝我的指導教授施吉昇，教導我如何尋找研究方向以及論述研究內容，他就像是我的燈塔，帶領我徜徉在物聯網及區塊鏈的大海中，並指引我找到明確的方向。在一次一次的討論中讓我進步，讓我從毫無頭緒的初生之犢，走向今天能成功地讓大家看到研究成果。施吉昇教授給我機會，讓我不斷的嘗試，在失敗中成長茁壯，進而完成這份作品。

謝謝我的家人，養育我成人，給我最大的鼓勵與支持，讓我能無後顧之憂地完成論文。我也要謝謝我的女朋友，在我瀕臨崩潰邊緣時給我滿滿的信心，督促我向前進，不敢鬆懈。

我更要感謝實驗室裡的每一位同學，一路上有你們的陪伴及鼓勵，是我在研究領域上精進最大的動力。

最後，我要謝謝一路上幫助過我、鼓勵過我的每一位貴人，因爲有你們的拔刀相助，讓我能更順利地完成論文，也謝謝我自己，以努力不懈、兢兢業業的態度完成這份論文，謝謝大家。

台北，夏，2018

楊凱崴

doi:10.6342/NTU201802852

摘要

在競爭激烈的市場中，如何提高生產效率、產品品質管理都是現今開發商和工廠所面臨的挑戰，由於缺乏機台資料整合及生產回溯追蹤，要找出問題根源並不容易。在現代的工廠中，生產流程上可能會有生產線合併或是分支等情形，每台生產線上的機台都會紀錄大量的傳感器數據以及機台資訊，但現今的工廠管理系統未必能有效地運用及負荷如此龐大的數據。

在這份研究中，我們設計一套基於區塊鏈的分散式生產流程追溯系統，並且在輕量級裝置的物聯網上實作，目的是提升工廠生產線的穩定性，降低生產不良率，最佳化工廠生產效能。去中心化的架構擁有部份容錯性以及可擴性，並且能夠更穩定的運作，發揮分散式系統的潛能。有了較佳的可擴性以及資料整合性，分散式帳本網路能夠在蒐集資料的同時確保資料安全並且有實力建造公開生產追溯系統，允許多家工廠在不用互相信任的前提下一同打造公開透明的生產追溯生態系統。

關鍵字 - 工業4.0，智慧工廠，生產回溯追蹤系統，區塊鏈，分散式帳本網路，機器對機器

doi:10.6342/NTU201802852

Abstract

In today's competitive business environment, manufacturing enterprises are facing the challenges of productivity improvement, product and process quality management due to the lack of data interaction and manufacturing process traceability. In modern factory, number of machineries are pipelined into assembly/production lines which may merge and branch to meet the procedure requirements. Each machine on the assembly line produce huge amount of sensing data and manufacturing data. However, many manufacturing system are not ready to manage big data.

This thesis aims on developing the distributed traceability system with Blockchain technology on IoT devices in order to improve the stability of the factory production line, to reduce defect rate and to bring the operational performance to a new level. Without depending on centralized storage, it is a robust, truly distributed peer-to-peer system and capable of node failure tolerance. With better scalability and data interaction, this distributed ledger network enables the use of data collection with security and potential of public traceability system protocol which allows multiple factories to participate, build an ecosystem of traceability together.

Keywords - Industry 4.0, Smart factory, Traceability system, Blockchain, Distributed ledger network, Machine-To-Machine (M2M)

doi:10.6342/NTU201802852

Contents

致謝 ii

摘要 iii

Abstract iv

1 Introduction 1

1.1 Motivation 1

1.2 Contribution 3

1.3 Thesis Organization 3

2 Background and Related Work 5

2.1 Background 5

2.1.1 Blockchain Technology 5

2.1.2 IOTA 5

2.1.3 Masked Authenticated Messaging 7

2.2 Related works 8

3 System Architecture and Problem Definition 10

3.1 System Architecture 10

3.1.1 Communication protocol 10

3.2 Problem Definition 12

3.3 Challenges 13

4 Design and Implementation 15

4.1 Message creation flow 15

4.2 Channel management 17

doi:10.6342/NTU201802852

4.3 Back trace path 19

5 Performance Evaluation **22**

5.1 Traceability Evaluation 22

5.2 Storage Overhead 23

5.3 Time Overhead 25

5.4 Security Analysis 27

5.5 Compatibility 28

6 Conclusion **29**

Bibliography **30**

doi:10.6342/NTU201802852

List of Figures

1.1 2017 Q2 top targeted industries based on attack volume. 2
1.2 Critical Aspects of Scaling comparison 3

2.1 Data structure of Blockchain and Tangle. 6
2.2 Data structure of Masked Authenticated Message. 7
2.3 An illustration of Merkle tree usage in MAM. 8

3.1 The system architecture of our traceability system. 11
3.2 The message publish process of MAM. 12
3.3 An illustration of targeted problem in our work. 13

4.1 The message publish process of our traceability system. 16
4.2 An illustration of Channel switching process. 18
4.3 An illustration of Channel merging process. 18
4.4 Back trace path for single source. 20
4.5 Back trace path for multiple source. 20

5.1 Time comparison of different approaches. 26
5.2 Time overhead of our proposed method. 26

doi:10.6342/NTU201802852

List of Tables

5.1 Parameters and size contained in the signatureMessageFragment of MAM bundle. 24
5.2 MAM storage comparison. 24
5.4 The relation between number of source, merge process and extra storage cost. 25
5.5 Security requirement evaluation. 27

doi:10.6342/NTU201802852

Chapter 1

Introduction

1.1 Motivation

In modern factory, number of machineries are pipelined into assembly/production lines which may merge and branch to meet the manufacture workflow. Each machine on the assembly line produces huge amount of sensing data and actuation data. In order to improve the stability of the factory production line and reduce defect rate, feedbacks collection is one of the main feature of traceability system.

In traditional approaches, data collected from machines and sensors are lack of interchanging, which brings difficulty on traceability. For example, data are labeled with machine ID, time and date or product serial number but are not cross-related with the data collected on other machines. Moreover, data are collected in trusted factory only. If we liked to track the whole supply chain, additional effort on access control and authentication are required in order to allow multiple non-trust parties participate. Traceability requires authenticated production records of every step of a final product which brings another challenge. A trusted institute for verifying the production record which will cost extra budget. Furthermore, centralized server system cannot tolerate single point failure and security threads on data integrity and privacy.

Manufacturers are found to be particularly vulnerable to hackers, according to NTT Security. Motivations for the attacks are often criminal in nature, including extortion via ransomware, industrial espionage and theft of data. Figure 1.1 shows that in Global Threat

doi:10.6342/NTU201802852

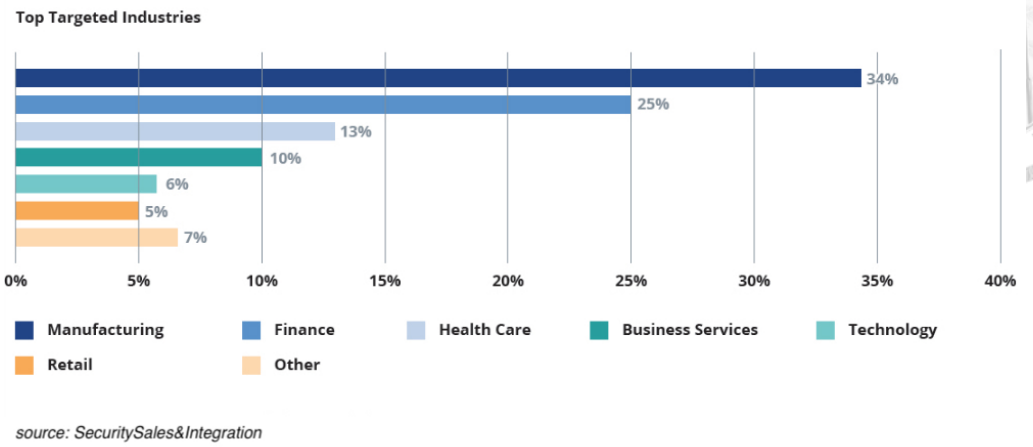


Figure 1.1: 2017 Q2 top targeted industries based on attack volume.

Intelligence Center (GTIC) 2017 Q2 Threat Intelligence Report [1], NTT Security found that 34% of all documented attacks targeted manufacturers. Therefore, we need a high security level system in order to overcome security threads.

Blockchain technology provides peer-to-peer distributed ledger network which first came out with virtual currency application and mainly active on Fintech domain. The integration of blockchain technology in traceability system can brings the benefit of immutable data storage and tolerance of node failures. Although Blockchain has the potential to overcome aforementioned challenges, we barely see related work on industry use case. The reason is the high cost for Blockchain maintenance.

IOTA, a distributed ledger network whose data structure is based on Directed-Acyclic-Graph and focuses on Machine-to-Machine (M2M) communication, is developed to bring back all the possibility in this field. Fujitsu [2] claims that there are five advantages of IOTA over the Blockchain, including transaction rate, scalability, verifiable manipulation security, etc. Figure 1.2 shows that IOTA has a better scalability compared to Blockchain due to the paralleled validation of consensus mechanism.

The functionality of data collection, information tracking, and process optimization has gone beyond the conventional observation or monitoring process. Nowadays, data collection has not been sufficiently addressed. A well designed traceability system can not only be used to identify the individual items in a final product but also be capable of

doi:10.6342/NTU201802852

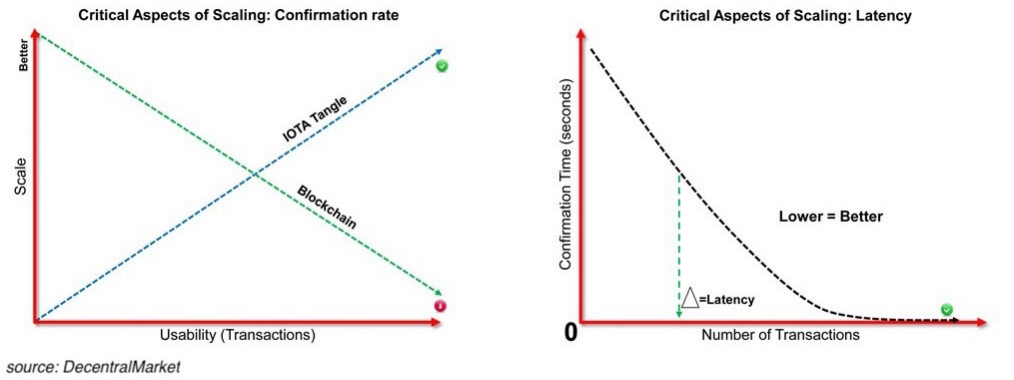


Figure 1.2: Critical Aspects of Scaling comparison

presenting manufacturing process step-by-step. Consequently, the consumers can check the product they purchased and the manufacturer can trace information for quality control, data analysis and other purposes.

1.2 Contribution

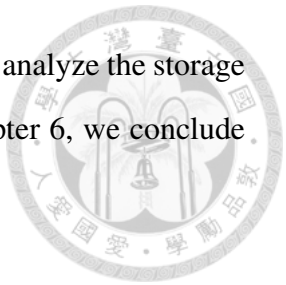
In this thesis, we design and implement a distributed traceability system on IOTA. With the benefit of tamper proof storage, our system provides a public protocol which allows multiple non-trust parties to participate and connect related data in order to build supply chain tracking for both customers and suppliers. Also, our proposed system supports complex tree-like workflow with fully traceability. Meanwhile, the proposal can overcome security threads. With this distributed traceability system, we can build an ecosystem of traceability for the industry.

1.3 Thesis Organization

The remainder of the thesis is organized as follow. In Chapter 2, we present the background of Blockchain technology, digital signature scheme and related work. In Chapter 3, we present our system architecture and define the problems and challenges we are facing in this work. In Chapter 4, we present our design and implementation of distributed traceability system on IOTA to overcome limitation of the original protocol. In

doi:10.6342/NTU201802852

Chapter 5, we evaluate the performance by measuring the traceability, analyze the storage overhead, time overhead, security threads and compatibility. In Chapter 6, we conclude the work.



doi:10.6342/NTU201802852

Chapter 2

Background and Related Work

2.1 Background

2.1.1 Blockchain Technology

Blockchain as known as public distributed ledger used to record all transactions across the network. A blockchain is a single linked list of blocks which each block contains a number of transactions and each transaction can be queried by every participators on the network. This allows anyone to verify transaction inexpensively which also brings greater transparency and trust to all participators.

The most well known blockchain technology implementation is Bitcoin which was created by Satoshi Nakamoto in 2008 [3]. Even though Bitcoin brings tremendous potential on blockchain, there are fundamental issues that cannot be solved, including low transaction throughput which leads to scaling problem. High transaction fee that makes it unfeasible to do small amount of payment is another constraint. That's why more and more cryptocurrencies are invented and aim to use Blockchain technology in different scenarios.

2.1.2 IOTA

IOTA is a distributed ledger protocol focused on Machine-to-Machine (M2M) communication of Internet-of-Things (IoT). The core technology behind IOTA is the Tangle

doi:10.6342/NTU201802852

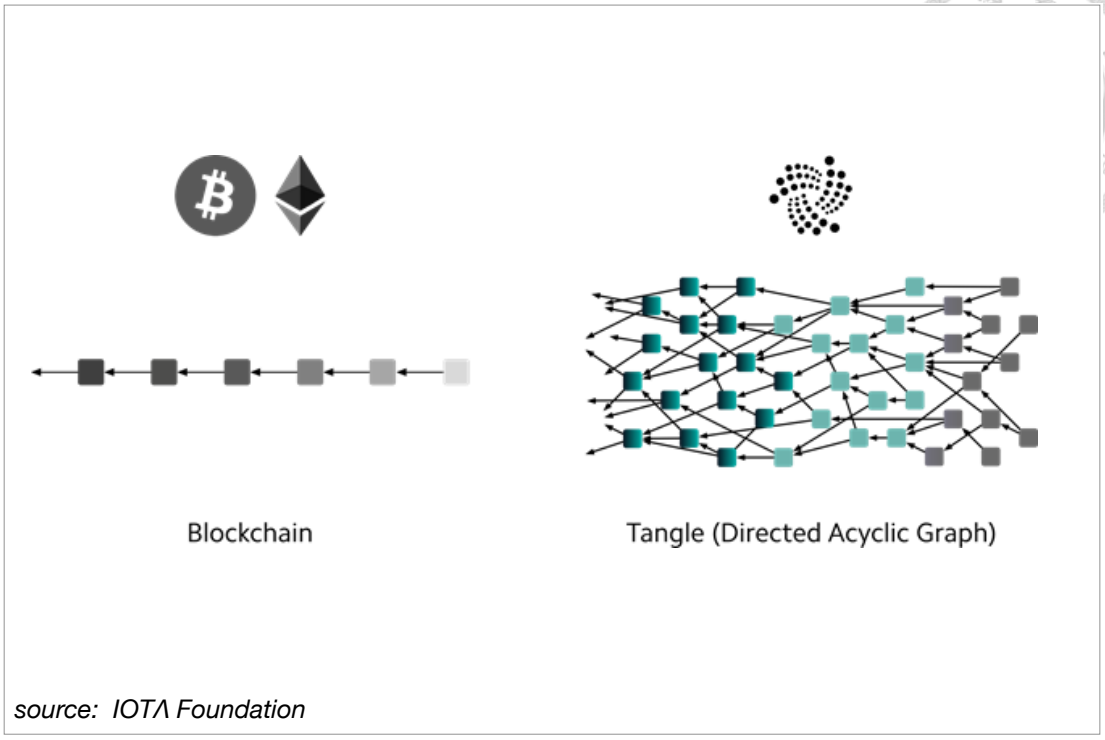


Figure 2.1: Data structure of Blockchain and Tangle.

[4]. It is a data structure based on Directed-Acyclic-Graph (DAG). Figure 2.1 shows that compared with other Blockchain implementations, IOTA has better scalability due to paralleled validation of transactions. In addition, IOTA's consensus mechanism does not require miners to maintain. According to the IOTA official introduction [5]:

> Each participant in the network that wants to make a transaction has to actively participate in the consensus of the network by approving 2 past transactions (IOTA Foundation, 2017).

This feature enables feeless micro-transactions which adapt to M2M's requirement.

Seed and Private keys

In IOTA, the seed is the master private key of an account used to generate private keys. In other words, a seed can be treated as an ID in IOTA although this seed should be kept secret.

Private keys are used to generate the address for sending/receiving payments and signature for signing transactions. IOTA claims their signature scheme is quantum-secure

doi:10.6342/NTU201802852

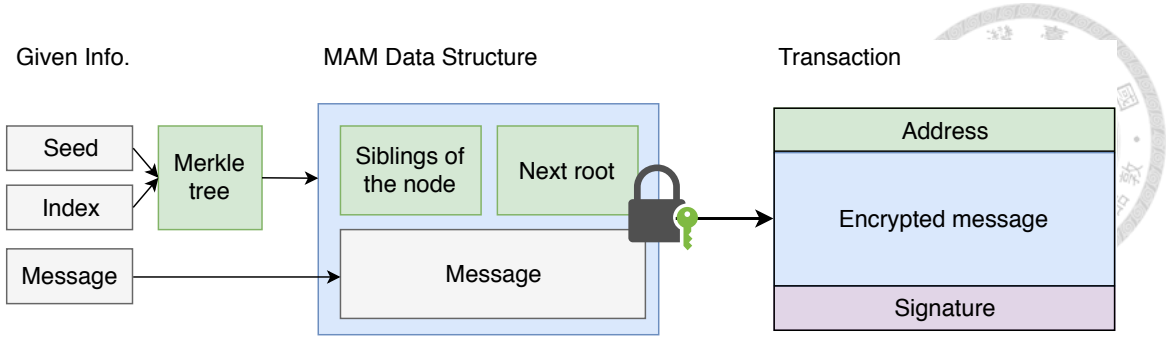


Figure 2.2: Data structure of Masked Authenticated Message.

by using Winternitz One-Time signatures [6]. Based on this scheme, private keys should never be reused but to generate new private key from the seed.

2.1.3 Masked Authenticated Messaging

Masked Authenticated Messaging (MAM) is a second layer data communication protocol which provides secure, encrypted and authenticated data stream for M2M over the Tangle. Each message is encrypted and shipped by transactions in IOTA. MAM provides 3 modes including Public, Private, Restricted modes. For simplicity, we used public mode in the following work. Public mode means that everyone with the root are able to decrypt the message. Figure 2.2 shows the data structure of MAM. Compound of Sibling of the node, Next root and Message are encrypted with the root of merkle tree, further description for merkle tree will be covered on the following subsection.

Merkle tree based signature scheme

MAM uses a Merkle tree based signature scheme to sign the encrypted message. The main purpose of this signature is to authenticate the identity of the message publisher. For example, Figure 2.3 shows that Merkle tree is a tree within which leafs PK_0, PK_1 are the private keys generated by a seed with index 0 and 1. Both keys will be hashed once and the root of merkle tree is the result of Hash(Hash(PK_0), Hash(PK_1)). When the users receive a MAM, they will do a signature validation and receive Hash(PK_0). Given a node and the siblings of the node, the message publisher can be easily verified by hashing them and reproduce the root.

doi:10.6342/NTU201802852

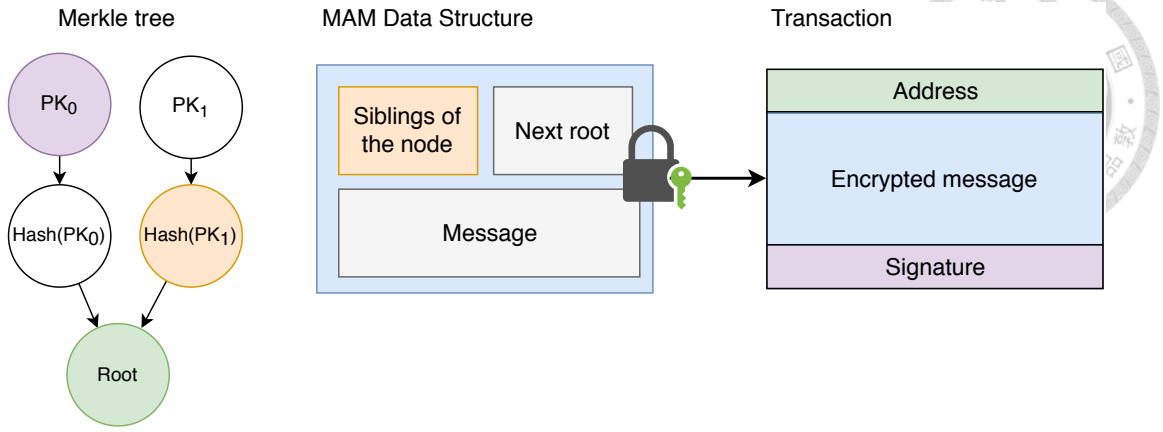


Figure 2.3: An illustration of Merkle tree usage in MAM.

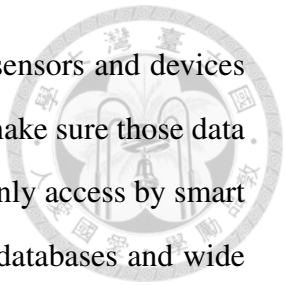
Channel and Message chain

The concept of channel on MAM looks after to FM radio channel, where the broadcaster and subscribers meet. In order to subscribe a channel, one should have the addresses of the messages. According to the MAM protocol, every message contains next message's address which brings out what we called as message chain. Combining this architecture with Merkle tree based signature scheme, every MAM contains the root of the next Merkle tree that leads to the next message. Given the root of a masked message, one is able to access the message chain start from the message with the root one is holding. Previous message will not be referenced to make sure message chain is forward secrecy. This enables data marketplace for any sensor device that brings lots of potential in machine economy.

2.2 Related works

In [7], they implemented two blockchains in a smart home scenario and focused on security and privacy of IoT devices. Public blockchain is used to store the access permission to data from the service provider. Private blockchain records the hash value of the data in database. Hash values are used as an index for data query and data integrity validation. Due to the heavy loading of Proof of Work (PoW) and expensive transaction fees, they did not store their data on the chain which means that the data security solely depends on

doi:10.6342/NTU201802852



database. The propose of using Blockchain in this work is because sensors and devices in smart house needs to send their data back to service provider and make sure those data are not corrupted. Furthermore, data should be confidential and can only access by smart home owners and service providers. Most of the job is achieved by databases and wide area network which Blockchain is only an assistive tool for permission control.

In [8], they proposes a IoT-based techniques for itemized data registration and information traceability in a digital manufacturing system. The designed system in there work itemize product item with universal unique identifiers (UUID). The information traceability is performed by integrating related data and information with UUID. Machines on the production line synchronized the current status of manufacturing process in order to categorize data correctly. The traceability of this work will be compared with our work.

doi:10.6342/NTU201802852

Chapter 3

System Architecture and Problem Definition

3.1 System Architecture

The system architecture of our work targets smart factories which require real-time data collection on manufacturing process. The blockchain framework we used here to deploy our distributed traceability system is IOTA and we will utilize MAM communication protocol to accomplish the traceability. The system architecture of our traceability system is shown in Figure 3.1. Each block in manufacturing process line represents a machine and each machine connects each other on IOTA network. In this distributed system, every device runs a Full Node client of IOTA which means they all participate in the Peer-to-Peer network and contribute their resources to maintain the consensus of the distributed ledger which is also known as Tangle in IOTA. Furthermore, each device is capable of publishing their own messages and broadcast to other machines.

3.1.1 Communication protocol

To publish a message with MAM protocol on IOTA require several steps, Figure 3.2 shows the process of publishing a message on Tangle. As mentioned in Section 2.1.3, merkle tree creation is required for message address allocation and signature validation.

doi:10.6342/NTU201802852

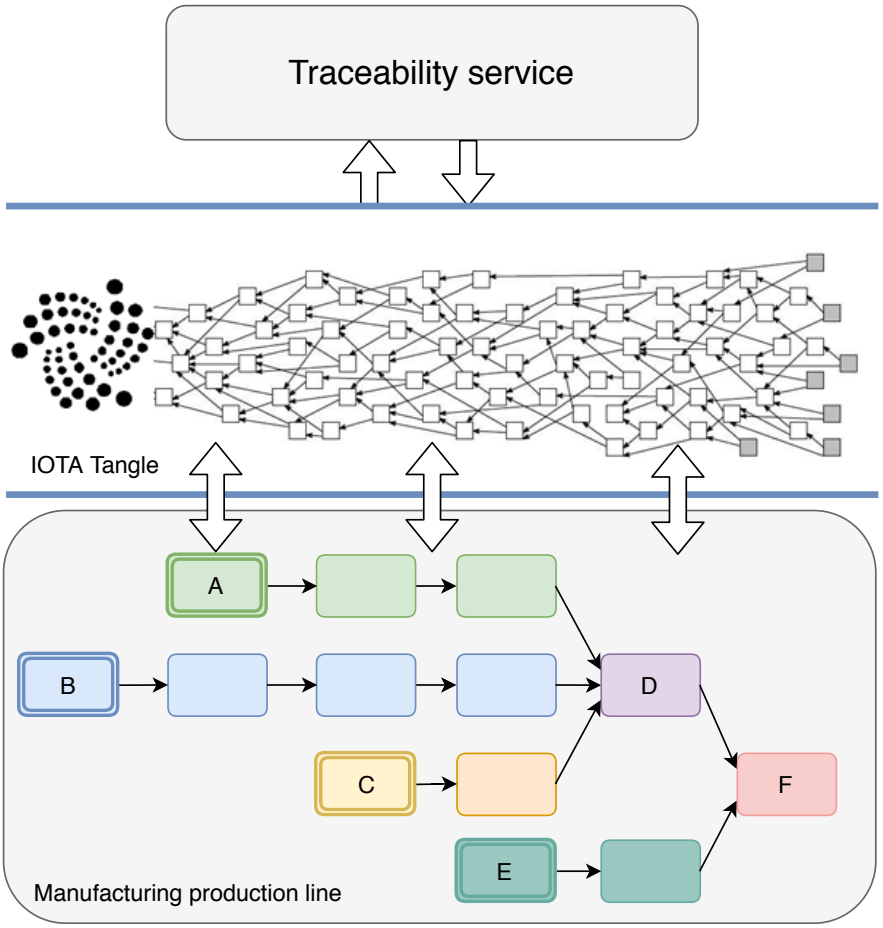


Figure 3.1: The system architecture of our traceability system.

doi:10.6342/NTU201802852

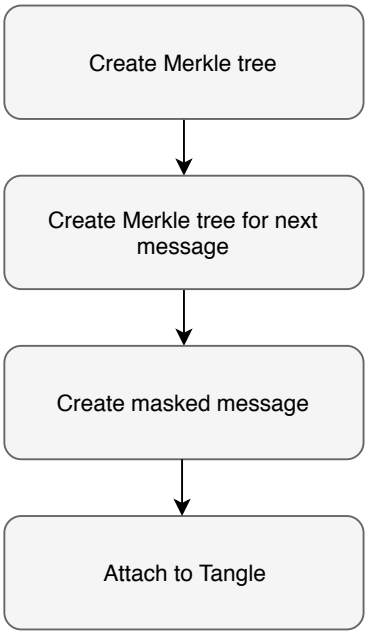


Figure 3.2: The message publish process of MAM.

To create a merkle tree, a seed is required to generate the leafs and also to identify the message owner. Next, the merkle tree for next message is created and this should be done because the next root is determined before its message has been published. As shown in Figure 2.2, a masked message contains several objects. Besides the plain message to be delivered, next root and the siblings of the node are also required. The detail process of masked message creation is explained as follow. First, we encrypt all the information mentioned above with the root of merkle tree. Then, the masked message will be signed by a private key. This private key is the first leaf of the merkle tree. Finally, the signature and masked message will be wrapped into IOTA's transaction format and attach to Tangle.

3.2 Problem Definition

Our goal is to design a distributed traceability system that supports complex tree-like workflow of manufacturing process shown as Figure 3.3. In this figure, there are three types of workflows. Message chains are able to be built with original MAM protocol.

doi:10.6342/NTU201802852

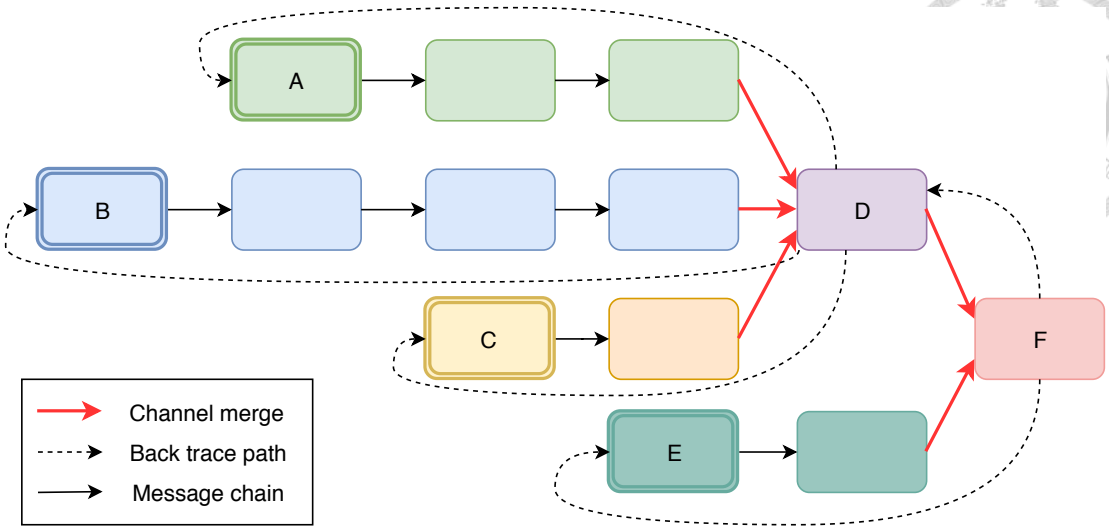


Figure 3.3: An illustration of targeted problem in our work.

The targeted issue is to design and implement Channel Merge and Back trace path in order to reach the goal, defined as follow:

- **Channel branch and merge management**: Multiple assembly lines merge leads to channel switching issue. Message chain should support workflow branch and merge process despite the tree-like workflow of manufacturing process. Otherwise, the message chain will be disconnected and lost the ability of traceability.
- **Back Trace Path**: The message chain of MAM does not refer previous message due to the forward secrecy concern. In order to enable the consumers to trace back the whole manufacturing process, the mechanism should support an efficient and secure back trace path to trace back to the sources with low computation time.

3.3 Challenges

IOTA has some limitations on address usage [9]. Any outgoing address which contains signature are not able to reuse again due to the risk of private key exposure. Each time one signs the message, half of the bits in the signature will be revealed [10]. In other words, to identify transaction sender's or message publisher's identification is very challenging. The reason is due to the quantum-computing protection one-time signature scheme design

doi:10.6342/NTU201802852

of IOTA.

In MAM, the next root is determined before next message is published, which brings a challenge in our case. While making a channel switch, the destination of next root is still unknown. Moreover, performing such action require seed sharing which leads to a security breach.

Message chain with forward secrecy brings an issue for traceability system. Without the address of the source one cannot trace back the whole manufacturing process. The challenge of designing a back trace path is to be compatible with IOTA's MAM protocol and does not violate MAM's security mechanism

doi:10.6342/NTU201802852

Chapter 4

Design and Implementation

In this chapter, we present the design of distributed traceability system with MAM protocol. In Section 4.1, we discuss the message creation process of our traceability system. In Section 4.2, we propose the solution for seed sharing issue in channel merging process. In Section 4.3 we describe the design of our back trace method on MAM protocol.

4.1 Message creation flow

In our design, we modify the MAM client to accomplish our system use case. Figure 4.1 shows the process of message publishing in our traceability system. Compared with the original process shown in Figure 3.2, two statements are added.

The first statement, switch channel, decides whether the next message will be published by a new owner. This will happen when workflow merge occurs in manufacturing process. The second statement, last message, decides if current message is the last message in the channel, checkpoint will be generated and added in the masked message. This will happen when the product is dispatched from the factory or merge process in assembly line. In summary, the message creation flow in our design splits into three scenarios. A normal message creation, a channel merge process and the last message of current channel. The implementation of these two statements are called as “channel management” and “back trace path” which will be discussed in the following sections.

doi:10.6342/NTU201802852

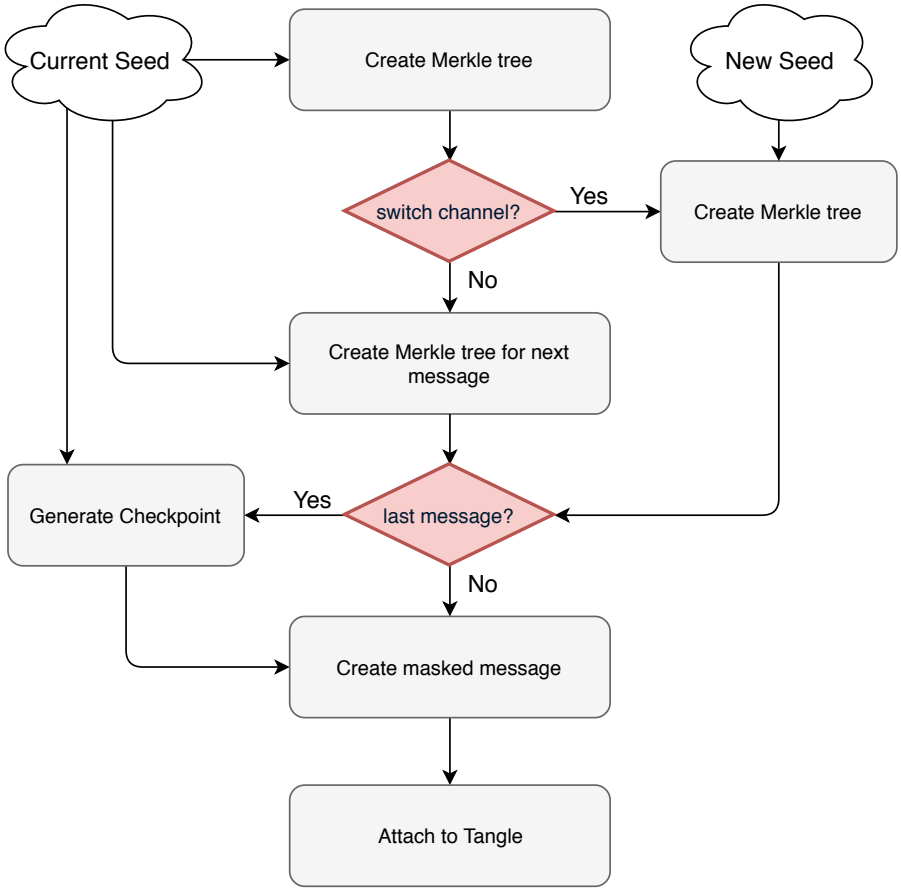


Figure 4.1: The message publish process of our traceability system.

doi:10.6342/NTU201802852

4.2 Channel management

In order to keep the message chain link properly, channel management should be handled gracefully. In our design, the seed is carried by component or product on the assembly line. When the merge operation occurs, a new seed will be generated. The previous channel of the components will be closed and linked the last message to the new channel created by the new seed.

The reason why this should be implemented is because our traceability system is aiming for cross factory usage. While components C_A and C_B are manufactured in different factories and end up assembling into product P_C, factory F_A will like to follow-up how C_A performed in order to optimize their product quality.

However, performing such process requires seed sharing which leads to a security breach. We solve this by using different seeds to create the last message. The function of masked message creation can be expressed as,

$$\text{MAM}(S, k)$$

where S denotes the seed of message publisher and k denotes the index of message chain.

Figure 4.2 shows an example of channel switching. The next root of $\text{MAM}(S_A, n)$ is assigned to a temporary channel $\text{MAM}(S_{A'}, 0)$ which belongs to $S_{A'}$. During this process, S_A that C_A is carrying has been replaced to $S_{A'}$ and it will be physically delivered to the subscriber. Figure 4.3 shows that after the components has successfully delivered, $\text{MAM}(S_{A'}, 0)$ and $\text{MAM}(S_{B'}, 0)$ are able to complete the merging process since channel H_C's first root is determined at this point. This makes sure the original seed will not be exposed to non-trust parties. Thus, it is guaranteed that all messages are published by the channel owner. The pseudo code of channel switching and merging process is shown in Algorithm 1.

doi:10.6342/NTU201802852

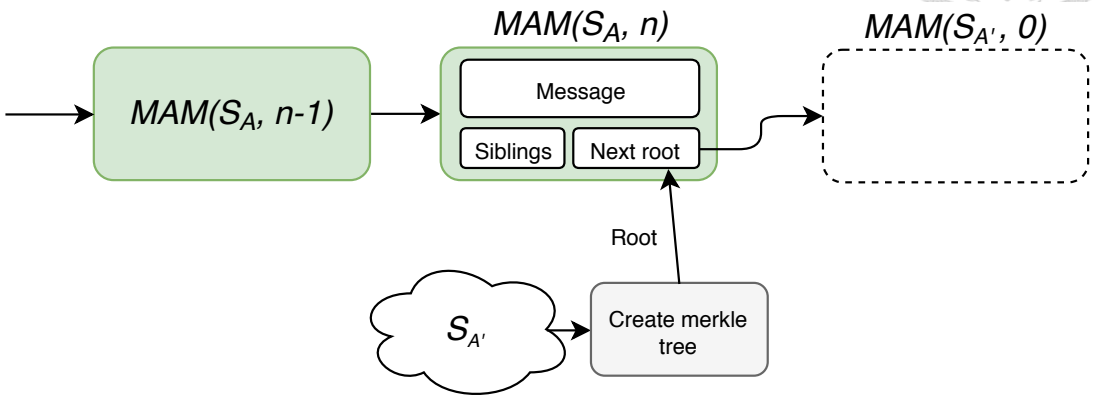


Figure 4.2: An illustration of Channel switching process.

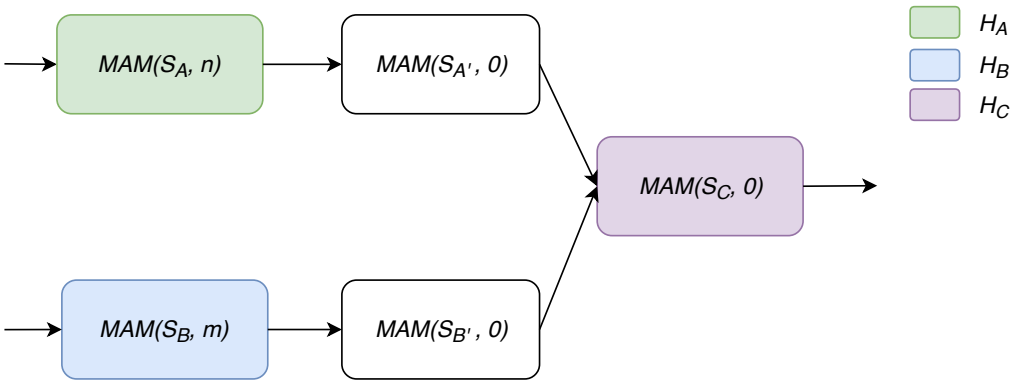


Figure 4.3: An illustration of Channel merging process.

doi:10.6342/NTU201802852

Algorithm 1 Channel switching and merging process

Input: The seed of message publisher S_A; The index of message chain n;
Output: Merged channel with the new publisher F_C;

μ_A = MAM(S_A, n);
if channel switch **then**
    μ_A.next_root = MAM($S_{A'}, 0$).root;
else
    μ_A.next_root = MAM($S_A, n+1$).root;
end if
μ_A.mask(msg); /* *encrypt message* μ_A */
attach(μ_A); /* *attach to Tangle* */

/* C_A *arrived* F_C */
$\mu_{A'}$ = MAM($S_{A'}, 0$);
$\mu_{A'}$.next_root = MAM($S_C, 0$).root;
$\mu_{A'}$.mask();
attach($\mu_{A'}$); /* *attach* $\mu_{A'}$ *and channel switch is done.* */

4.3 Back trace path

In this section, we describe our design of back trace method. Our goal is to make the subscriber being able to review the manufacturing process of the product they purchased with one simple address. On the other hand, the manufacturer can check the cause of problem if it was an individual case or would apply to the complete production branch. With the design of back trace path and channel management, our traceability system provides tracing both ways to ensure full traceability for each product.

We implement a back trace method by leaving checkpoints in the end of each channel. Checkpoint is created when the current channel is closing up such as merge case or dispatched from the factory. Figure 4.4 shows that Checkpoint contains the first root of this channel. For more detail, the first root can be reproduced with the same seed. Therefore, one can receive the source address and review the manufacturing process on an IOTA full-node. Single source is easy to implement, let's take a look on a multiple source complex graph. Figure 4.5 shows an example of multiple source manufacturing flow. According to previous section, we know that during merging process the seed will be physically delivered and so does the checkpoint of each component. These checkpoints will be stored at the first message of the merged channel. Therefore, H_D in Figure 4.5 will receive the

doi:10.6342/NTU201802852

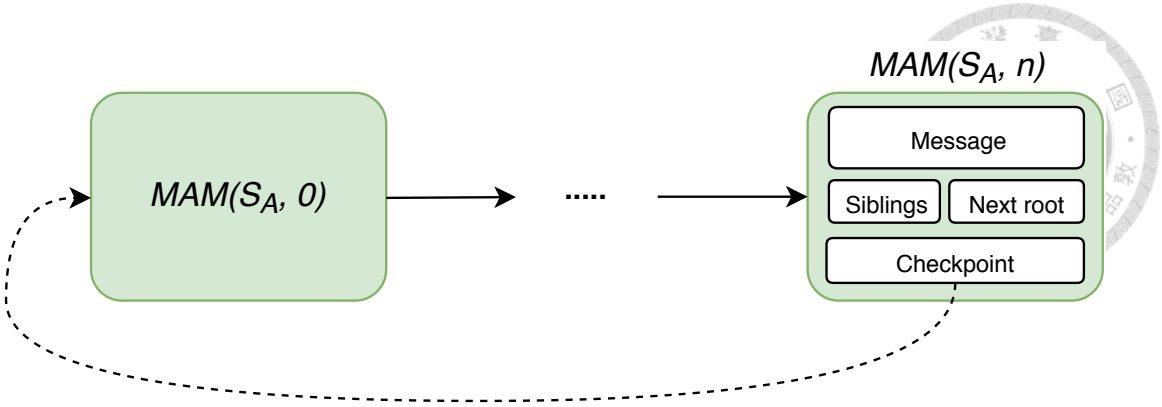


Figure 4.4: Back trace path for single source.

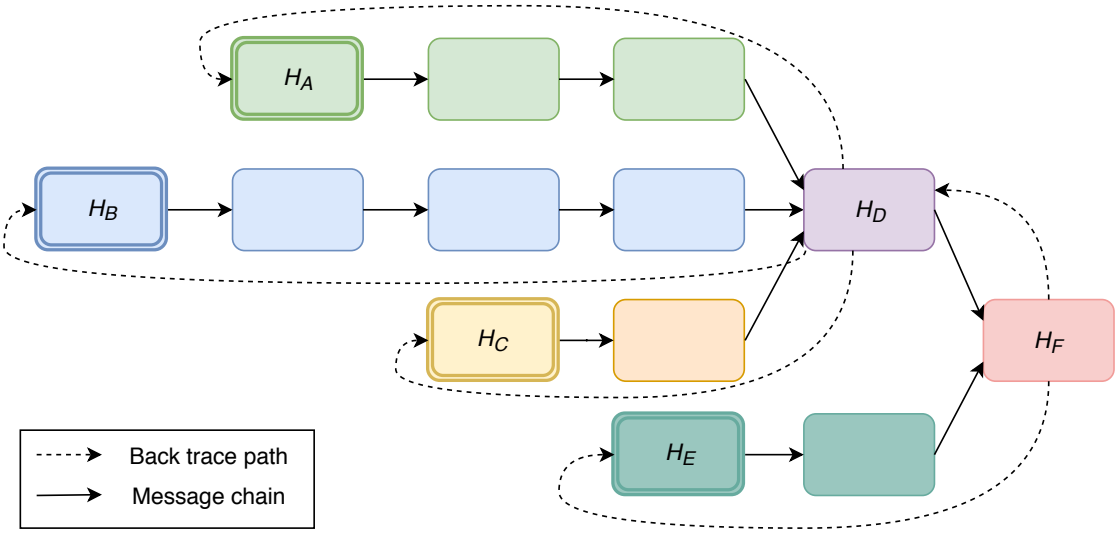


Figure 4.5: Back trace path for multiple source.

first address for H_A, H_B, H_C and publishs the first message contains checkpoints of components. By following the aforementioned rules, We are able to build back trace paths for the manufacturing process.

In order to make the subscriber able to review the manufacturing process, we implement a back-tracing function as shown in Algorithm 2 to locate the sources addresses. The input of the function is the address of last message which we assume it was printed on the production. The `fetchMAM` function is provided by IOTA's MAM client which will fetch and decrypt the masked message of the given address. After we have the sources, it's easy to walkthrough the route with `fetchMAM` function and we are able to represent the tree-like workflow graph of manufacturing process like Figure 4.5.

doi:10.6342/NTU201802852

Algorithm 2 Back tracing function

Input: The last message's address of the product λ_k;
Output: An array of source addresses $\Omega = \{\omega_0, \omega_1, \ldots, \omega_n\}$;

1: $\mu =$ fetchMAM(λ_k);
2: **if** μ.checkpoints $= \emptyset$ **then**
3: **return** $\emptyset$; //*Checkpoint not found*
4: **else**
5: $E = \mu$.checkpoints;
6: **end if**
7: **while** $E \neq \emptyset$ **do**
8: $\varepsilon = E.pop()$;
9: $\mu =$ fetchMAM(ε);
10: **if** μ.checkpoints $= \emptyset$ **then**
11: $\Omega \leftarrow \varepsilon$
12: **else**
13: $E \leftarrow \varepsilon$;
14: **end if**
15: **end while**
16: **return** Ω;

doi:10.6342/NTU201802852

Chapter 5

Performance Evaluation

In this chapter, we will first discuss the performance of traceability. Furthermore, we will discuss the overhead of our proposed method in two aspects, storage and time. Last but not the least, we will discuss the security analysis of our system and the compatibility with MAM protocol on IOTA.

5.1 Traceability Evaluation

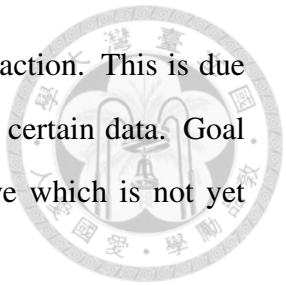
In order to justify our design, we should evaluate the traceability of our proposed method. According to [11], the evaluation methods of traceability have no consensus on what should be the benchmark. In their work, they did a research on benchmarking for traceability techniques and defined evaluation methods and metrics. We will choose some methods that are suitable as reference and evaluate our work.

Goal satisfiability, the typical goals of trace techniques are:

- Goal 1: To find all the relevant documents with high accuracy.
- Goal 2: To find relevant documents without inclusion of irrelevant documents.
- Goal 3: To accurately rank the most relevant documents near the top of the retrieved list.

Goals 1 and 2 are satisfied in our proposed method, since the back trace path design enable us to trace back to the sources and fetches the history of manufacturing process.

doi:10.6342/NTU201802852



However, the address of MAM is required in order to perform such action. This is due to the security design for keeping out unauthorized parties to access certain data. Goal 3 required further data post-process in order to satisfy this objective which is not yet implemented in this work.

Robustness, the ability to measure the essential accuracy of a technique not affected by random chance due to peculiarities in a data set. There will be the case that more than one message appeared on a same address. If this happens, our system will first validate the signature and filtered out those which were not authorized. The rest of the messages that passed the validation will be accepted.

5.2 Storage Overhead

Since our system is deployed on light-weight devices, storage overhead will be our concern. We compare the storage overhead with the related work [8]. In their work, a mechanical encoder-based line synchronization was implemented so that all modules are able to use the UUID of product items for data integration. In other words, no extra storage cost but extra database I/O is required to approach traceability.

As mentioned in Section 4.3. We left checkpoints in the last message of each channel. In order to evaluate the scalability of this design, we'll like to discuss the storage overhead.

Before we discuss the storage overhead of our work, we need to make a brief introduction on IOTA's transaction. Each transaction contains 2673 trytes of data. Including `transaction hash`, `signatureMessageFragment`, `address`, `nonce`, `trunk and branch transaction hash`, etc.

The field that we stored our message is `signatureMessageFragment` which provides 2187 trytes of size. Message with size greater than 2187 trytes will be carried by multiple transactions which known as **bundle**.

The structure of MAM bundle contains two sections, signature section and MAM section. In our experiment, security level of address is set as 1 and the number of merkle tree leafs is 2. Signature section carries 2187 trytes of signature and MAM section contains multiple objects. Table 5.1 shows the size of each object contained in MAM bundle. The

doi:10.6342/NTU201802852

size of Unmasked Message m depends on the data to be stored.

Parameters	Size (trytes)
Unmasked Message	m
Next root	81
Siblings	81
Checkpoint	81
Signature	2187

Table 5.1: Parameters and size contained in the signatureMessageFragment of MAM bundle.

Scenario	Extra Storage Usage	Total Storage (trytes)
Basic	-	2187 + 81 + 81 + m = 2349 + m
Last message of the channel	Checkpoint$*1$	2349 + 81 + m = 2430 + m
Channel Merged	Checkpoint$*n$	2349 + 81$*n$ + m

Table 5.2: MAM storage comparison.

Table 5.2 shows the storage overhead comparison on different scenarios. The Basic scenario is the original publish method MAM provided which contains all the parameters shown above at Table 5.1 without checkpoint. The other two scenarios, the last message of the channel or the first message of the merged channel, requires extra storage for the checkpoint parameter. The value n depends on the number of merged components in channel merged scenario or $n = 1$ when the message presented as the last message of the channel. The key of storage overhead is the number of source components and the count of merge operations. Table 5.4 shows the storage cost related to the number of checkpoints. A complex manufacturing process with 128 number of sources and 64 merge process will need 192 Checkpoints which requires 15,552 trytes approximate to 9 MB extra space only, In other words, storing checkpoints required additional 3% of storage in average.

doi:10.6342/NTU201802852

Number of sources	Merge process	Checkpoints	Extra Storage cost (trytes)
1	0	1	81
2	1	3	243
3	1	4	324
3	2	5	405
4	1	5	405
4	2	6	486
4	3	7	567
8	4	12	972
16	8	24	1944
32	16	48	3888
64	32	96	7776
128	64	192	15552

Table 5.4: The relation between number of source, merge process and extra storage cost.

5.3 Time Overhead

Our distributed traceability system aims for real-time data collection in smart factories scenarios. Therefore, time overhead is the cost we must concern. Compared the store and access time duration of IOTA with database approach based on related work [7] and Bitcoin as data storage were shown as Figure 5.1. The experiment environment we ran on was an Intel® Core™ i7-7500 2.7 GHz computer. In this figure, the time overhead of storing operation in IOTA and Bitcoin included message encryption, signature generation and PoW. IOTA's overhead is 3 times larger than the database approach. Although Bitcoin requires more than 8 times overhead to accomplish such work.

Next, we'll like to discuss the time overhead of our proposed method. Figure 5.2 shown that over 50% of time was caused by MAM protocol. The reason could be attributed to the additional encryption and hashing operation. The additional 45 ms time consumed in our proposed work is caused by the checkpoint generation. This process required extra Merkle tree creation which will only appear on channel merged scenario and the end of manufacturing process. In other word, the extra overhead of our proposed method depends on the complexity of the manufacturing workflow.

doi:10.6342/NTU201802852

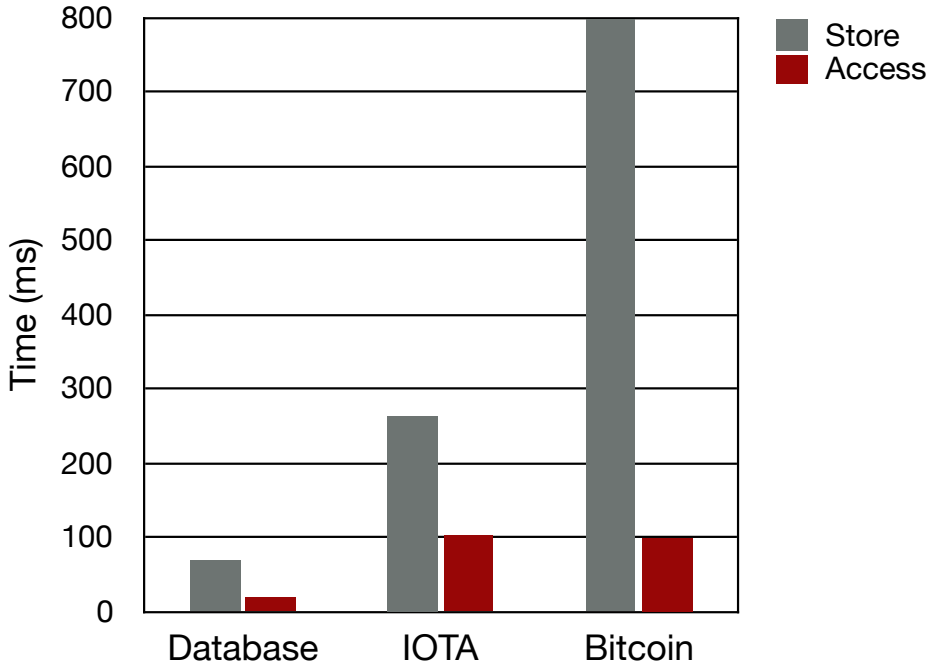


Figure 5.1: Time comparison of different approaches.

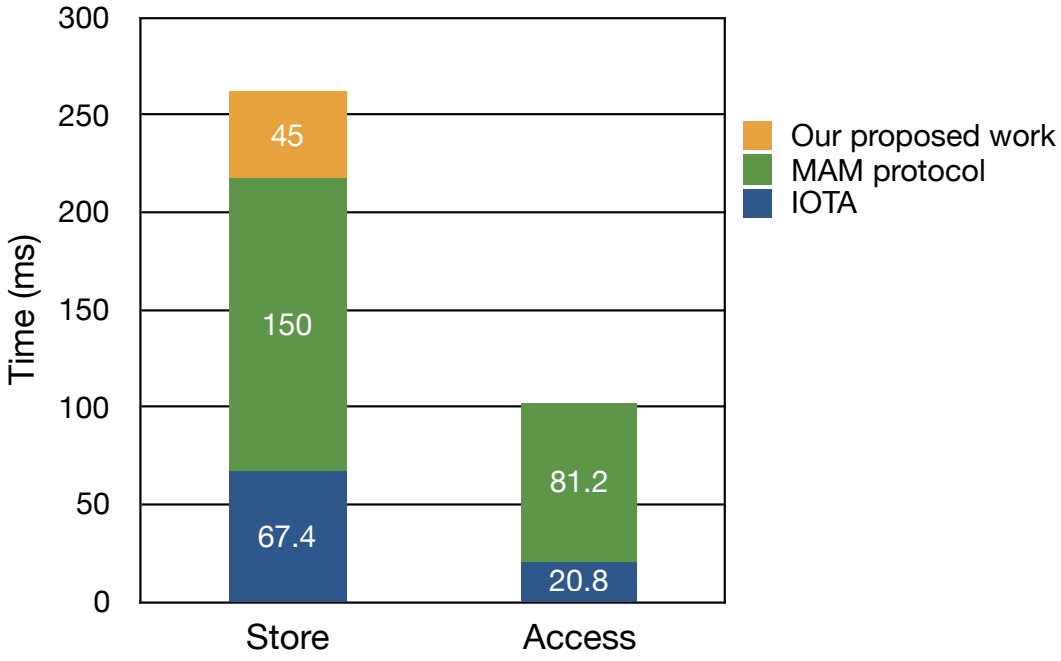


Figure 5.2: Time overhead of our proposed method.

doi:10.6342/NTU201802852

5.4 Security Analysis

We evaluated our security performance by reviewing three main security requirements, namely: Confidentiality, Integrity and Availability, known as CIA [12]. Confidentiality means that only authorized users are able to read the message. Integrity promises that sent messages are immutable and will be received at the destination without any change. Availability makes sure that each service or data are available when an authorized user needs. The second and third requirement are fulfilled by the main feature of distributed ledger network provided by IOTA. However, the first requirement depends on the channel mode one chosen for MAM which was mentioned in 2.1.3. In our work, we have chosen the public mode which used a symmetric encryption on messages. Table 5.5 summarizes how IOTA network of our traceability system achieved the aforementioned security requirement.

Requirement	**Explanation**
Confidentiality	Users without message address (root) will not be able to find (decrypt) the data
Integrity	Once the transaction was attached on Tangle and broadcasted to other nodes, there is no chance to override/modify/corrupt the data
Availability	Every synced nodes are able to fetch the data on Tangle (The more nodes one have linked, the higher availability one will get)

Table 5.5: Security requirement evaluation.

In order to justify our work is secure, we shall make sure our proposed method does not violate IOTA's security mechanism and rules which will be discussed in the following paragraphs:

The risk of a reused address was mentioned in section 3.3. Our proposed method for publishing messages required an index as the input argument. As long as we make sure each machine uses unique index, address duplication will not happen.

In our proposed method, the seeds will not leave the factory who owns the product or component. We assumed that seeds are physically printed or attached on the item during manufacturing process. Before it dispatched from the factory, no matter it was going to the next factory or ready for sell, the seed will be destroyed and replaced by a new generated seed or the public address according to different scenarios. Therefore, no one

doi:10.6342/NTU201802852

has the ownership of a finished manufacturing process and the seed will not be revealed.

Forward secrecy is one of the feature that MAM protocol provided. Since each message contained the root of next merkle tree, it was easy for users to follow the data stream. On the other hand, users are not able to gain access to data they have not been given access to. Combined this with Channel splitting feature, publishers are allow to split the message chain in order to manage the access control on specific data. MAM accomplish channel splitting by generating merkle trees with different size. Merkle tree's root generated by the same seed and same index but different number of leafs will turn out different output. This is due to the number of hash operation in mekle tree.

In our proposed method, the checkpoint of our back trace path will only linked to the first root with the same size of merkle tree. Therefore, one could only access the data streams that were allowed to read and the forward secrecy on channel splitting is still valid.

5.5 Compatibility

Our proposed method is fully compatible with the original IOTA's MAM protocol without modifying the client node. The additional parameter Checkpoint we implemented is stored in message section which will not effect the data structure of MAM. In other word, the system we proposed was able to deploy on IOTA main-net. During the message fetching process, the checkpoint parameter will be considered as a part of the message parameter and the message were able to decrypt by `fetchMAM` function provided by MAM client.

doi:10.6342/NTU201802852

Chapter 6

Conclusion

In this thesis, we design and implement a distributed traceability system with Blockchain technology by utilized and modified the MAM protocol provided by IOTA. We solve the channel switching issue with temporary seed that promised the privacy of the original seed. Furthermore, we implemented the back trace path by leaving checkpoints and the storage overhead of our design is reasonable. The system we proposed was able to deploy on IOTA main-net and fully compatible with the original MAM protocol.

doi:10.6342/NTU201802852

Bibliography

[1] "Ntt security, global threat intelligence center (gtic) 2017 q2 threat intelligence report," 2017. [Online]. Available: https://www.nttsecurity.com/en-us/gtic-2017-q2-threat-intelligence-report

[2] "Fujitsu: These are the advantages of iota over blockchain," 2018. [Online]. Available: https://iota-news.com/fujitsu-these-are-the-advantages-of-iota-over-blockchain

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[4] P. Serguei, "The tangle," 2017.

[5] "Iota introduction." [Online]. Available: https://iota.readme.io/docs/what-is-iota

[6] L. Lamport, "Constructing digital signatures from a one-way function," Technical Report CSL-98, SRI International Palo Alto, Tech. Rep., 1979.

[7] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, March 2017, pp. 618–623.

[8] Z. Wu, Z. Meng, and J. Gray, "Iot-based techniques for online m2m-interactive itemized data registration and offline information traceability in a digital manufacturing system," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2397–2405, Oct 2017.

[9] "How addresses are used in iota." [Online]. Available: https://iotasupport.com/how-addresses-are-used-in-IOTA.shtml

[10] "Very basic estimates on the risk of reusing the winternitz signatures." [Online]. Available: https://public.tangle.works/winternitz.pdf

[11] Y. Shin, J. H. Hayes, and J. Cleland-Huang, "A framework for evaluating traceability benchmark metrics," 2012.

doi:10.6342/NTU201802852

[12] N. Komninos, E. Philippou, and A. Pitsillides, "Survey in smart grid and smart home security: Issues, challenges and countermeasures," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1933–1954, Fourthquarter 2014.

doi:10.6342/NTU201802852