DECOUPLES: A PRIVACY-PRESERVING SOLUTION FOR TRACEABILITY IN SUPPLY CHAINS

MOURAD EL MAOUCHI

to obtain the degree of Master of Science in Computer Science
Data Science & Technology Track
with a specialization in Cyber Security
to be defended publicly on February 8, 2018

DELFT UNIVERSITY OF TECHNOLOGY
Faculty of Electrical Engineering, Mathematics & Computer Science
Department of Intelligent Systems
Cyber Security Group

STUDENT NUMBER:
4204379

THESIS COMMITTEE:
Prof.dr.ir R.L. Lagendijk
Prof.dr.ir. G.J.P.M. Houben
Assist.Prof.Dr. Z. Erkin
O. Ersoy, MSc
A. Beije, MSc

SUPERVISORS:
Assist.Prof.Dr. Z. Erkin

ABSTRACT

Traceability is an increasingly important aspect of the supply chain with several highlights throughout the last few decades. Parties, such as consumers and government agencies, have shown an increase in demand for information regarding their products and materials. Studies throughout the last few years have proposed approaches for a traceability system for supply chains. These systems employ either a centralized or decentralized network to overcome the trust concern present in supply chains.Although there exist numerous frameworks for traceability solutions, these frameworks failed to address the concerns with regards to privacy-sensitive information, certificate verifiability, and auditability.

In this research, we aim to improve the existing solutions and address the aforementioned concerns, where we decompose the concerns into four aspects. First, we analyze the appliance of blockchain technology for a decentralized traceability system. Next, we analyze possible anonymization techniques to preserve the privacy concerning the identity and the corresponding relationships of actors. Then, cryptographic primitives are examined to prove the ownership of a certificate, in a privacy-preserving manner. Finally, a technique is required to realize product-specific auditability for supply chains.

We propose two systems to the concerns mentioned above. For the former, we propose TRADE, a fully transparent and decentralized traceability system. The system shows that blockchain technology can be successfully incorporated to achieve traceability in supply chains. Moreover, consumers and other parties can view all the data in the system and verify the claims of actors on the products. Positive brand-image is gained by the latter. The second system, DECOUPLES, is the first decentralized, unlinkable and privacy-preserving traceability system for supply chains. The system incorporates cryptographic techniques to address the privacy, certificate verifiability, and auditability concerns. In addition, we propose the PASTA protocol, which allows unique tracking keys per product, per actor. The protocol also anonymizes the receiver of a transaction. Moreover, cryptographic primitives are used for actors to prove the ownership of a certificate without revealing privacy-sensitive and linkable information. Our complexity analysis and proof-of-concept implementation results show that DECOUPLES is a feasible and practical traceability system for supply chains, that benefits both the included business as well as the end-client.

"Careful. We don't want to learn from this."

— Bill Watterson, *'Calvin and Hobbes'*

"I want chocolate cake, I am hungry!"

— Zekeriya Erkin, *'every day'*

ACKNOWLEDGEMENTS

If anyone would have told me that I would perform research on applied cryptography, and actually understand it all, I would say that they would be lying. After all, cryptography is not an easy field, and I think most friends and family can agree that I am not the smartest. However, this research would be, to put it gently, impossible without the limitless support of my family and friends. Especially at the points I wanted to give up, crawl in a fetal position and cry on the ground.

A special thanks to my supervisor Zeki Erkin. The formidable support, mentoring and supervision of him got me to this point. Your guidance, even though it was tough the first few months, were unforgettable and I will always appreciate it. You have created an environment of intelligent people that I can have meaningful discussions with. Besides discussions, I could also be myself and have fun throughout the research. I entered your research group completely clueless, and after nine months, actually still am... But it was fun. You have shown me that life should be balanced between serious work and having fun, where I excel at the latter.

Furthermore, I would like to express great appreciation to my fellow Master –and PhD-students. Throughout the last months, we had many great moments, and some dips, but mostly an enjoyable time. When we laughed, we laughed together. The tremendous amount of cake I have eaten, as well as the somewhat "colorful" jokes, have made me grow both in size and taught me more jokes. A special thanks to Zhijie, Oguzhan, Gamze, Chibuike, and Majid. Thanks for the countless coffees and the great apple-times.

I would want to close by saying "thank you" to all those who were bearing with me, my complaining and my countless jokes. I will be forever grateful for all the connections I have made and the things I have learned. This research report is a perfect example of what I have learned. I can now safely say that I actually know what cryptography is, namely magical formulas and unreadable mathematical security proofs.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF PROTOCOLS

ACRONYMS

SCM	Supply Chain Management
CA	Central Authority
BCT	Blockchain Technology
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDLP	Elliptic Curve Discrete Logarithm Problem
CO	Certificate Organization
SSCC	Serial Shipping Container Code
RP	Range Proof

1 INTRODUCTION

Supply Chain Management (SCM) is an integral part of businesses and is essential to company successes and customer satisfaction. An important aspect of SCM is traceability, to track products throughout supply chains. Traceability dates back to the 1930s where European countries wanted to prove the origin of high-quality drinks such as French champagne [66]. The importance of traceability has been highlighted over the past few decades due to various food safety-related concerns such as the bovine spongiform encephalopathy disease and the avian influenza [66]. Besides the food industry, other industries are also affected due to the problems regarding security, safety, and product quality.

Besides the participants of supply chains, consumers, NGOs, governments, suppliers, and buyers show an increase in demand for information regarding their products and materials. Concerning organic, fair trade and environmentally friendly products, certificates have been developed to increase transparency along supply chains and inform the consumer with regards to the quality, safety, and sustainability of the product [26]. However, the certificates are presented as labels, which are not verifiable for consumers and are easily faked by criminals [57]. Certificates play an essential role for consumers to identify the sustainability claims of products. Consequently, supply chains requires to provide a means of verifying the claims presented by the label or certificate, providing verifiability to consumers.

To achieve the required traceability throughout supply chains, a system that "records and follows the trail as products, parts, and materials come from suppliers and are ultimately distributed as end products" is required [30]. Currently, industries maintain their own systems, while sharing only a minimal amount of information. This limits the possibility of achieving traceability within the entire supply chain. The need for individual systems is mainly due to the presence of confidential data. Moreover, since there are multiple parties present within the competitive supply chain landscape, relationships between actors are also considered confidential. The actors in supply chains want to retain control of their data, instead of handing it over to a central authority, making a centralized solution infeasible.

This chapter gives a brief overview of supply chains in Section 1.1. Next, the practice of traceability in supply chains is discussed in Section 1.2, where the privacy, trust, and verifiability concerns of these practices are described in Section 1.3. Finally, we argue our contributions in Section 1.6 and the outline of the research in Section 1.7.

1.1 SUPPLY CHAIN

Supply chains is an ever-growing sector that spans the entire globe. Beamon et al. define supply chains as "an integrated manufacturing process wherein raw

materials are converted into final products, then delivered to customers" [7]. We differentiate supply chains into five actors: producers, processors, transporters, distributors, and retailers.

SCM is the active management of supply chain activities to achieve a sustainable competitive advantage and maximize customer value. The management of supply chainss covers all aspects of product development, sourcing, production, and logistics. SCM links the physical and information flow within in supply chains [79]. The most prominent motive behind the formation of the SCM is the competitive advantage [50, 71]. In addition, according to Giunipero et al. [23], SCM can improve the competitive advantage and profitability by improving the customer satisfaction. Globalization ensures that supply chains actors conduct business across the globe, resulting in new and stronger regulations. Traceability played a significant role for both auditability and compliance with regulations.

1.2 TRACEABILITY IN SUPPLY CHAINS

Traceability is defined by the International Standards Organization (ISO) as *"the ability to trace the history, application or location of an object"* [60]. According to Global Standards One (GS1), an international organization that develops and maintains standards for supply chain across multiple sectors, defines traceability as *"the ability to track forward the movement through specified stage(s) of the extended supply chain and trace backward the history, application or location of that which is under consideration"* [62].

The area of traceability can be distinguished in two types, depending on the direction that information is collected in the chain: backward –or forward traceability [32]. Backward traceability is the ability to find the origin and characteristics of a product. The latter, forward traceability, is the ability to find the location of products. Traceability can be further differentiated into three categories, depending on the business scope: (i) internal, (ii) external, and (ii) whole-chain traceability. In (i) each company owns its individual system and performs traceability over the processes and products in their manufacturing plant. Category (ii) contains the data exchange between trading partners. The last one, (iii), combines the internal and external traceability to gain an overview of the entire supply chain.

In order to ensure good practice and respect for people and the environment impacted by supply chains, traceability is used to verify and assure claims by the involved parties. Over the last few decades, certifications have played a significant role to indicate that a product complies with a list of requirements. The UTZ certified impact report of January 2014 shows that the implementation of certifications results in higher yields for farmers [74]. According to a report by BSR [56], parties in supply chains face stakeholder demands for product information and having the means to verify sustainability claims. The first is partially achieved through the use of labels and certifications, while the latter is opaque to consumers, NGOs, advocacy organizations, as well as other involved organizations.

1.3 TRUST, PRIVACY, VERIFIABILITY & AUDITABILITY CONCERNS

The actors in supply chains have limited trust in each other due to the competitive nature and the presence of confidential data. Each actor generates data with regards to their products, which can be vital to their business. Therefore, there is no trust in a centralized system. The actors do not wish to store their data at a CA, that controls the system and is susceptible to collusion and the alteration of data without the consent of the owner. In addition, it introduces the possibility of a single point of failure. Since a single party or organization has control of the system, data loss or data leakage at this party can impact all actors that participate within supply chains. A system is required that only allows validated actors, participating in supply chains, to gain write permissions to the system. Nevertheless, consumers should gain insight into the system to provide traceability and transparency. An emerging technology that can address the aforementioned trust concerns is blockchain technology.

Supply chains, as a complex and interconnected network, contain parties that are competitors of each other. Not all data can be shared in a system encapsulating these parties, resulting in local and private systems that are not interconnected. Consequently, whole-chain traceability systems are rarely present in current supply chains. In addition, the relationships between the actors are considered confidential, and exposure can degrade these relationships. A system that links products without the disclosure of the relationships is therefore required. In the current state, no traceability system has achieved the aforementioned privacy properties to the best of our knowledge.

Products, satisfying a set of requirements, contain labels. These labels are a result of certificates awarded to a product, such as Fair Trade, Animal Welfare, Rainforest Alliance, and the Carbon Footprint where the first two are ethical and the following two environmental certifications [26]. However, since these labels are printed on products, they are easily faked [57]. Furthermore, the misuse of trademarks and fake or inferior materials show how counterfeit goods and theft of intellectual property is hurting consumers and companies [3]. This is becoming an increasing concern which can directly influence brand-trustworthiness and directly impact the demand for manufacturing companies. The presence of fake certificates is the industry's responsibility to solve. Nevertheless, there is no easy way for consumers or other parties to verify the certificates on products. In the organic food area, producers aim to show the full life-cycle of the product. However, the main importance for consumers and other parties is to validate that the product is indeed organic, which is not possible in the current setting.

In the case of a recall or disaster, data is retrieved regarding the affected products. Currently, this data is opaque to organizations, such as government agencies. When the data is retrieved, a full transcript is provided containing other information, which is undesired. A full transcript of all products does not aid in the recall process. The opposite is achieved, it slows down the effectiveness to address the affected products. Therefore, specific information should be retrieved for auditability purposes. For auditability, only the specific information

should be retrieved. However, when product-specific information is requested, it should not leak any information regarding the other products.

1.4 BLOCKCHAIN TECHNOLOGY

In general, blockchain is a distributed database for transaction processing. All transactions in a blockchain are stored in a distributed ledger, in possession of all parties in the network. The blockchain single-handedly determines the present state of the system [33, 83]. The blockchain is the underlying system of Bitcoin, a peer-to-peer electronic payment system created by Satoshi Nakamoto in 2008 [52]. Blockchain technology enables a decentralized system without the requirement of a CA, which introduces trustless transactions. The ownership of a transaction, a record on the blockchain, is proven through cryptographic primitives rather than a CA. The core aspects of blockchain technology are: decentralization, immutability and not requiring trust [33]. Moreover, blockchains can be differentiated in two dimensions: public/private and permissioned/permissionless. The first indicates the visibility and access to create transactions, while the latter indicates the permission to access the blockchain.

1.5 RESEARCH STATEMENT

The trust, privacy, verifiability and auditability concerns imposed by supply chains require a new approach to achieve traceability, with the desired requirements. This research aspires to address the aforementioned concerns while achieving a system that provides traceability in a novel manner by using certificates. To achieve this, in a privacy-preserving manner, we utilize blockchain technology as a decentralized solution. The main research question for this research is formulated as follows.

How can we achieve traceability for supply chains in a privacy-preserving manner, where parties can verify certificates, and product-specific auditability is achieved in a single system by employing blockchain technology?

This research question is decomposed into the following sub-questions.

(i) How can we create a fully transparent, decentralized traceability system for supply chains?

(ii) How can we anonymize relationships between actors, while utilizing blockchain technology?

(iii) How to proof that an actor holds a certificate without revealing his identity?

(iv) How can we provide product-specific auditability?

1.6 OUR CONTRIBUTIONS

In this research, we present two systems to achieve traceability in supply chains. The first system, TRADE, achieves full traceability and provides validation of the authenticity of transactions using blockchain technology. The system shows that blockchain technology is a promising technique for traceability solutions in supply chains. Restrictions can be enforced in the validation process for a set of, or all, actors. In addition, we achieve traceability of products due to the linking property of blockchain technology.

TRADE does not take the privacy, verifiability, and auditability concerns into account, whereas DECOUPLES, on the other hand, does incorporate the requirements mentioned in Section 1.3. Anonymization techniques are incorporated into DECOUPLES to preserve the privacy of the actors and make transactions unlinkable to the sender or recipient of the transaction. Also, we introduced a novel protocol, named PASTA, allowing product-specific auditability. The protocol enables recipients to provide a single product-specific tracking key. The holders are then able to track and reveal all the product-specific transactions of that particular actor. The transaction amounts in DECOUPLES are hidden using range proofs and ring signatures. Certificates, owned by actors, are shown in the system in a privacy-preserving manner since relationships of actors and the privacy should be preserved. The appliance of non-interactive zero-knowledge proofs made it possible to prove the possession of a certificate without revealing any privacy-sensitive information. Moreover, the sharing of confidential data between actors in the network is provided through encryption methods. We believe that DECOUPLES is the first system that utilizes cryptographic primitives and a novel approach of traceability in a privacy-preserving manner for supply chains.

Based on proof-of-concept implementations of the presented systems, we show that the requirements for supply chains are achieved in a feasible and cost-efficient manner. Moreover, our implementations suggest that a cryptographic approach to preserve the privacy and yet achieve traceability can achieve the efficiency required to be deployed in a real-world scenario.

1.7 RESEARCH OUTLINE

The structure of this research is composed in the following way. The first part of this research provides background information on supply chains, the traceability aspect and previous works on this aspect with the accompanying issues. Chapter 2 gives an overview on traceability in supply chains, existing approaches for it along with the issues accompanying it. In Chapter 3 the cryptographic preliminaries are discussed that are used in this research. The second part of this research describes novel solutions to achieve traceability for supply chains by proposing two systems. The first system TRADE is presented in Chapter 4, followed by the second system DECOUPLES that improves upon the first system in Chapter 5. Then, an evaluation is given on the proposed systems in Chapter 6 Finally, we discuss the obtained results, provide an outlook for future research possibilities and conclude in Chapter 7.

2

SUPPLY CHAIN TRACEABILITY

The supply chain is a complex, interconnected network of actors. We differentiate the actors in a supply chain in five types: Producers, Processors, Transporters, Distributors, and Retailers. Several approaches have been taken in the literature that aim to achieve a traceability solution for supply chains. However, they all assume that actors in supply chains are willing to share data openly. Moreover, they do not take the concerns discussed in Section 1.3 into account. This chapter is constructed as follows. Firstly, the five types of actors are discussed. Secondly, the three traceability categories are described. Thirdly, the traceability system approaches found in the literature are examined and discussed. Lastly, the open questions and concerns on achieving a traceability solution for supply chains are discussed.

2.1 ACTOR TYPES

We identified five types of actors that are applicable in supply chains. These actors collaboratively perform a series of processes and operations on products, from raw material to the end-product. These actors are introduced below, where Figure 2.1 illustrates the actors in an exemplary situation.

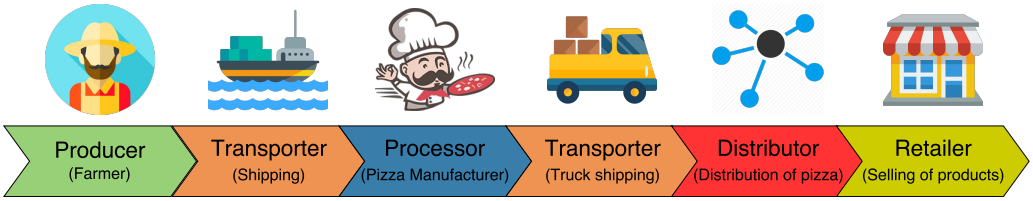


Figure 2.1: The five actors illustrated, creating pizza as a product.

PRODUCERS Supplying materials in their raw form, such as grains, fruits, vegetables, meat, fish, poultry, etc., is the task of producers. Producers are the source of products for the actors following in a supply chain. This actor is responsible for the creation and registration of new products.

PROCESSORS The processors have the general task to transform the raw materials, which are received from one or multiple producers, into products that meet the consumer requirements. This process is also referred to as *manufacturing*. Each processor has its procedures in place and often in a different order. These procedures are often considered confidential.

TRANSPORTERS The primary task of transporters is to efficiently transport the products and ensure that the product is not damaged during this process. Furthermore, this actor provides insight into the location of the transported

products, since transportation companies are increasingly equipped with GPS trackers.

DISTRIBUTORS Distributors act as the link between the producers, processors, and retailers. They achieve products in a significant amount and leverage the infrastructure of warehouses and distribution centers to distribute the products. Due to the distribution across (international) boundaries, they have to comply with several local regulations. These regulations include but are not limited to, the reduction of freight transport intensity, improvement of vehicle usage [47] and regulations regarding human rights, labor, environment, and anti-corruption by the UN Global Compact.

RETAILERS The final actor in the supply chain is the retailer, who functions as a point where consumers can buy the product. The role of the retailer in SCM is to manage their inventory in such a manner that a vendor can efficiently respond to their need and to create flexible capacity to adjust and support the supply chain infrastructure in demand.

2.2 TRACEABILITY CATEGORIES

Recall from Section 1.2 that traceability can be differentiated into three main categories. The traceability category achieved in the supply chain is categorized based on the categories discussed in this section. The three categories are shown in Figure 2.2. To effectively discuss the three categories, we consider the creation of deep freeze pizzas as a use-case.

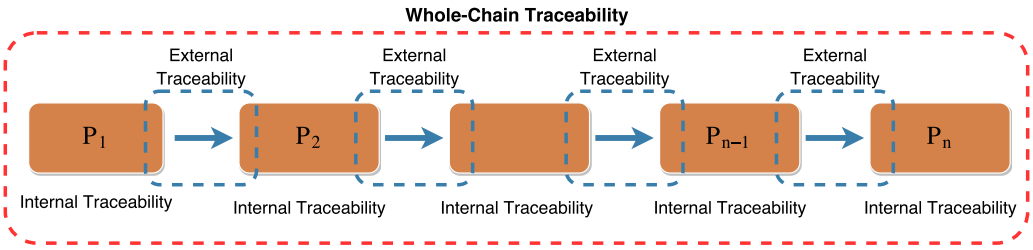


Figure 2.2: The three traceability categories.

2.2.1 *Internal Traceability*

Internal traceability occurs within a company in the supply chain and features the identification and tracking of what is made from what, when and how. It is also called *process traceability*. We speak of internal traceability when an actor receives instances of traceable items as input, and new traceable items are outputted [72].

Products are subjected to internal processes, one or more sub-processes performed by the same party, which are tracked according to a product. An internal process is required to consist of the following four sub-processes: movement, transformation, storage, and destruction. For the creation of a pizza, this con-

sists of the movement of the ingredients of it, the placement of the ingredients on a pizza, the storage in freezers at a particular location and the destruction of failed pizzas. Internal traceability aims to reduce costs and improve productivity, which is essential for keeping track of inventory, purchasing, and other in-house accounting.

EXTERNAL TRACEABILITY External traceability focuses on the tracking of physically transferring items between actors. In this system, each actor can track back an item to the direct source, and the direct recipient of the item is retrieved. This is called the "one step up, one step down" (OUOB) principle [72], and is used in a variety of supply chains.

In this category, the transfer of the pizzas is stored and tracked. It compromises the capability to achieve forward and backward traceability between actors. For example, the Dutch supermarket Jumbo can track where the ordered frozen pizzas are. Each actor in a supply chain is responsible for recording input and output data of himself, but not information that may be several steps ahead or behind to provide adequate tracking of the products. This results in the possibility to understand the custody chain of the pizzas. Data regarding the deep freeze pizzas are exchanged between actors, such as the Jumbo, to gain insight on where the products originated from, such as the producer of the ingredients on top of the ordered pizzas.

WHOLE-CHAIN TRACEABILITY The combination of internal and external traceability provides the whole-chain traceability solution, which encompasses the entire supply chain. In whole-chain traceability, both the internal processes and the physical transfer of items are encapsulated. Everyone along the supply chain receives visibility into the journey of a product. From the initial raw material at the producer to the final product being sold at the retailer. This type of traceability is an often sough option.

Insight into the full life-cycle of products provides actors, government agencies, retailers and consumers more information into the entire supply chain. Full insight improves multiple elements in the supply chain such as, but not limited to, increased recall times, better planning and scheduling and provide tracking insights to customers. For example, if the aforementioned deep freeze pizzas contain a bacteria, all the affected pizzas can be traced back and can be destroyed promptly. Besides, it provides several advantages in the area to save paper and intermediate systems which translate or match information from several actors. Insight into the sustainability of products is also achieved, due to the visibility of the life-cycle of the products.

2.3 PRIOR ART

In the last decades, researchers have paid attention to develop several approaches to achieve traceability in supply chains. Each approach applies different methods to design a traceability system for supply chains. In literature, we can differentiate the research in two main areas: methodologies to achieve traceability and the shift from a centralized to a distributed/decentralized system for trace-

ability. The first discusses the appliance of technologies to address traceability and the effect of them, while the latter aims to distribute a traceability system due to the complex nature and trust issues present in supply chains. This section discusses the approaches taken by researchers in literature and existing anonymization techniques for decentralized systems.

2.3.1 *Existing Traceability System Approaches*

Studies have attempted to achieve traceability without the creation of a new system, but rather use generated data by the actors as in [75]. However, due to the absence of standardization and the inefficiency of these systems, others have introduced new information structures or entire frameworks with traceability at its core [8, 9]. Besides, with new emerging technologies over the years, these have been applied to the field for cost-reduction and efficiency reasons [38].

A popular method that has been applied is the use of analytical methods. In [75] by Van Der Vorst et al. in 2006, a simulation environment called ALADIN is introduced that embeds food quality change and sustainability indicators in discrete event simulation models. The system has actors of the supply chain modeled and the flow of products to track them. However, at the core of ALADIN, quality change models are contained. By performing various simulations on use cases, the quality changes of products can be viewed. Their approach assumes the presence of the required data and to have a standardized form of the data to use it for the models. Besides, it does not take the distribution and manufacturing of multiple products into account, which is inevitable in the current supply chains.

In 2007 Kelepouris et al. [38] examined the appliance of radio frequency identification (RFID) technology for traceability and proposed an information infrastructure. The employment of RFID-technology reduces the investment costs while aiding traceability for supply chains. Their system utilizes a centralized information model to cut costs for the participants in the system. Their work has been purely theoretical, the application of their information infrastructure has not been tested through practical implementation. Also, they merely discuss the data that is stored on an RFID-tag. The data might be considered confidential, while this has not been addressed in their work. Therefore, their study emphasizes the cost-reduction, rather than efficiency and validity of RFID-tags for traceability.

Bechini et al., in 2008 [8], analyzed the main issues emerging at different abstraction levels for a new information structure for supply chains. From these issues, a set of suitable patterns are made that encode general traceability semantics. Bechini et al. discuss an essential aspect for traceability, namely the adoption of a generic data model to support collaboration. Generic data has been facilitated by the appliance of XML and SOAP as data structures. Bechini et al. have also created a prototype of their proposed system. The advantages are noted as the reduction of time to execute every-day tasks, a significant decrease in the error-rate caused by replicated data entries and reduced cost of the adoption of e-business processes. These advantages do not discuss the efficiency and performance on traceability. With their solution all the data is stored

in a centralized location which is responsible for the traceability, introducing a significant risk regarding trust due to the substantial amount of data it holds.

Similar to the work of Kelepouris et al., in 2013 Kang et al. introduced a set of services called traceability services (TS) and accompanying algorithms to aggregate product information, [36]. These services are defined on five query types based on a traceability requirement analysis. The movements of products have been modeled, and algorithms are created to satisfy the five query types. The tracking of products, or other business assets, is done based on RFID-technology. In their solution, different kinds of data are collected and stored in their centralized system. However, confidentiality of the data has not been taken into account, and no conclusion can be made on the real-world feasibility. In comparison to the work of Kelepouris et al., this work provides a more detailed and sophisticated solution.

Another approach to traceability is the usage of the Internet of Things (IoT), a network of interconnected physical devices and sensors. In a case study by Zhang et al. [82], a smart sensor data collection strategy for IoT is proposed. The collected data is used in proposed algorithms to trace contamination sources and to backtrack potentially infected food. Their approach was to model the IoT infrastructure for food supply chains to provide provenance. Rather than tracing products, their goal was to discover contaminations and detect infected food. In their research, the assumption was made that all information of food products is hosted by a centralized system and organized uniformly. However, as aforementioned in the previous works, standardization is yet to be achieved.

2.3.2 *Decentralized Traceability System Approaches*

The supply chain consists of a large, complex network of actors that each own confidential and privacy-sensitive data. Most approaches, as shown in the previous section, take a centralized approach for traceability. Centralization requires a single entity, or organization, to control and manage the system. Nonetheless, centralization introduces issues such as trust, fraud, corruption, tampering, and falsification of information [73].

Blockchain technology originating from its first application in Bitcoin, a peer-to-peer electronic cash system [52], has drawn the attention of many researchers. A blockchain is, in its essence, a distributed database for transaction processing. It removes the presence of a central authority and lays trust in the cryptographic protocols it utilizes. Blockchain technology has been applied to several industries, such as the energy sector [35], finance [69] and identity management [41]. The supply chain and its traceability aspect has also gotten attention throughout the last years.

An ontology-based smart contract design of a proof-of-concept blockchain system has been created by Kim et al. to enable traceability in supply chains in 2016 [39]. Ontologies are used in combination with blockchain technology to achieve provenance tracking. Due to the requirement of a common interpretation of data across organizations, ontologies are used to enforce common data standards informally. The work shows that ontologies can be used to develop blockchain applications for traceability. The assumptions made in this research

are mainly focused on the aspects to trace products, rather than the semantics and importance of the data in question. It shows the appliance and feasibility to apply ontologies in this setting.

Furthermore, Feng Tian has combined RFID-tags and blockchain technology to create a traceability system for the agri-food supply chain in China [73]. Feng discussed that a decentralized approach for traceability could solve the issues in a centralized approach, namely: trust, fraud, corruption, tampering and falsifying information. The appliance of RFID-technology is used to reduce costs and to store the data on a blockchain, which provides several features: decentralization, trustless, reliable database with anonymity. While the last feature, anonymity, is mentioned in the research, no appliance of it is shown. Moreover, no analysis is done on the combination of anonymity and achieving traceability. Nonetheless, the proposed system shows advantages in tracking possibilities, the enhancement of the credibility of safety information and in combatting fake products.

Abeyratne et al. discuss the need for transparency and traceability by providing a broader view of the issues in supply chains [1]. The need for transparency is argued based on the child labor scandal of Nike in 1996 [13]. Furthermore, sustainability and the presence of certificates show the importance of the understanding of a product life-cycle for consumers [5, 20]. However, as stated in their work, the result is merely a physical logo that is not verifiable. To address this, their system lets certifiers digitally sign the profile of actors that hold a certificate. To achieve this, the identity of the actors is disclosed to the entire network, discarding the privacy aspect. Besides the verifiability of certificates, the visibility into supply chains is another business challenge. Actors rarely have insight into the other actors that are involved in their products. The limited visibility between actors in results in reduced visibility for consumers. The system does not take privacy into account, as well as the weaknesses of their certification solution.

2.3.3 *Existing Anonymization Techniques for Blockchains*

The rise of blockchain and cryptocurrencies has resulted in the community developing new protocols to anonymize users in a blockchain system. Currently, there are three primary methods to achieve anonymization, namely the mixing protocol by DASH [18], zk-SNARKs by ZCash [64] and the RingCT-protocol by Monero [55, 63]. First off, we provide two properties that are important for this research and the anonymization techniques. The properties are satisfied by all the aforementioned techniques, where we denote the properties as follows.

- **Untraceability:** for each incoming transaction, all possible senders are equiprobable.
- **Unlinkability:** for any two outgoing transactions it is impossible to prove they were sent to the same person.

2.3.3.1 *DASH: Protocol-level Mixing*

In the protocol used by DASH, the aim is to anonymize the sender and receiver of transactions and therefore the accompanying account balance of users. The anonymization is achieved through a mixing protocol using a decentralized network of servers called Masternodes. Through the Masternodes, it introduces a two-tier network that allows the network to provide anonymization while avoiding the need for a trusted third party that could compromise the integrity of the system.

MASTERNODE NETWORK The Masternode network provides high availability and a required level of service to serve the network. A requirement to become a Masternode in the network is to obtain 1,000DASH. The Masternodes function as the mixing service in this system, which requires the execution of multiple sessions to anonymize transactions thoroughly. Each session increases the anonymity of a user's transaction.

PRIVATESEND PrivateSend is the core protocol of DASH to provide anonymity to the network. It is an improved version of the CoinJoin protocol [45]. The protocol mixes the user's transaction with the inputs of (at least) two other people. The transaction amount is split into denominations of 10^x, where $x \in \{-2, -1, 0, 1, 2\}$. The mixing service is provided by the Masternodes. A request is made to these Masternodes, which are randomly picked, to mix your transaction inputs with others. In this process, no identifiable information is sent to the Masternodes, so it cannot be linked to users. The denominated inputs are mixed in a new transaction in which the users can transfer the mixed input back to themselves. The transaction directed to the user himself is sent to a new address, called a "change address".

A malicious Masternode can follow the funds of users with a probability of $\frac{n}{t}^r$, where n is the total number of nodes controlled by the malicious Masternode, t is the total number of Masternodes in the network, and r is the depth of the chain. Due to the requirement of owning 1,000DASH, it becomes hard for an adversary to achieve a high probability of success.

2.3.3.2 *ZCash: zkSNARKs*

ZCash utilizes zero-knowledge proofs to guarantee the validity of transactions and to provide anonymization. The zero-knowledge proof construction used is called *zk-SNARK*, it stands for "Zero-Knowledge Succinct Non-Interactive Argument of Knowledge". The succinct part of zk-SNARK provides the ability that a proof of only a few hundred bytes can be verified within a few milliseconds.

The general idea of zk-SNARKs is to turn what you want to prove into an equivalent form of knowing a solution to some algebraic equations. These equations can then be evaluated on a candidate solution without revealing any sensitive information to the verifying parties. The generation of zk-SNARKs can be decomposed into four steps as follows.

$$\text{Computation} \xrightarrow{(1)} \text{Arithmetic Circuit} \xrightarrow{(2)} \text{R1CS} \xrightarrow{(3)} \text{QAP} \xrightarrow{(4)} \text{zk-SNARK}$$

CONSTRUCTION OF ZK-SNARKS The transaction validity function is first represented in a mathematical representation of the smallest possible logical operations, called an arithmetic circuit. From the arithmetic circuit, a *Rank 1 Constraint System (R1CS)* is built and checked if the values are in the correct order. The R1CS is then represented in the form of a *Quadratic Arithmetic Program (QAP)*, as discussed in [22], which is a polynomial. The verifier checks the constraints between polynomials by checking the polynomials at a randomly chosen point. To overcome the chance that the prover crafts invalid polynomials that satisfy the identity at a point, the homomorphic properties and pairings of elliptic curves are used to evaluate polynomials blindly – i.e., without knowledge on which point it is being evaluated.

APPLIANCE OF ZK-SNARKS For a shielded transaction in ZCash, the sender is required to create proof that shows that

1. the sum of the input is equal to the sum of the output,
2. knowledge of the private spending keys and
3. that the private sending keys are cryptographically linked to a signature over the whole transaction.

Unspent transactions are represented by commitments, where the spending equals revealing a nullifier. A commitment consists of the hash of (i) the recipient's address, (ii) amount being sent, (iii) a random value unique to the commitment and (iv) a random nonce. When a commitment is spent, the nullifier of it is published, which is the hash of the spending key and the unique value. If the nullifier has been found in the set of nullifiers that are being tracked by the network, it is considered a double spending attempt and thus invalid.

PUBLIC PARAMETER CEREMONY ZCash utilizes a so-called *public parameter ceremony* at the creation of the system. During this ceremony, the proving and verifying keys are generated that are being used to create and check proofs as discussed before. The generated proving and verifying keys are spread across the entire network to be used. The prover is required to perform significantly more computational power to create a proof, due to the constructions of a zk-SNARK as aforementioned, whereas the verification is simplified. The time to create a proof costs roughly 44 seconds, while the verification only takes approximately 27 milliseconds[1].

2.3.3.3 *Monero: RingCT Protocol*

Monero utilizes a protocol derived from CryptoNote [63]. This protocol has been improved upon, and an additional protocol has been created, namely the *RingCT*-protocol. The protocol provides the ability to hide the identity of the sender, receiver and the amount that is being transferred. The basic idea is to hide the sender in a group of actors using *MLSAG ring signatures*, the recipient's identity is hidden through *stealth addresses*, and the amount is hidden through Pedersen commitments. In the RingCT-protocol, each actor holds an address,

1 See: https://speed.z.cash/changes/

denoted as two public ec-keys (A, B) and the private corresponding ec-keys (a, b), where $A = aG$ and $B = bG$.

STEALTH ADDRESSES Stealth addresses is a technique, based on ECC, where a user publishes a single address and receives unconditional unlinkable payments [63]. Stealth addresses create a pair P, R where the first denotes the destination key and the latter the transaction public key. The protocol uses $\mathcal{H}_s$, which is a cryptographic hash function $\{0,1\}^* \rightarrow E(\mathbb{F}_q)$. The recipient is required to check every transaction if it is meant for him by using the private key a and public key B. The spending key is derived by using both private keys a, b of the recipient, denoted as $x = \mathcal{H}_s(aR) + b$.

MLSAG: MULTI-LAYERED LINKABLE SPONTANEOUS ANONYMOUS GROUP The anonymization of the sender of a transaction is achieved by employing MLSAG ring signatures, similar to the LSAG described in [42]. The main difference is that in MLSAG a ring signature is created on a set of n key-vectors, rather than on a set of n keys as in LSAG. In addition, MLSAG uses so-called key-images. These key-images are unique per MLSAG ring signature and are used to prevent double-spending attempts. The general idea and intent of the MLSAG ring signature is

- to prove that one of the n signers knows the secret keys to their entire key vector and
- to enforce that any of the m signing keys of the signer is linked if it is used in another MLSAG signature.

HIDING TRANSACTION AMOUNT Transaction amounts are hidden through Pedersen commitments. For every transaction amount v a random scalar α, further noted as a mask, is chosen. Furthermore, a second generator H is used which is required to be public and unrelated to G such that no apparent relation should exist (e.g., $aG = H$). The amount v is split into a binary representation, where for each binary representation a Pedersen commitment is created. A Borromean ring signature [46] is then created on the set of commitments, to decrease the space complexity.

2.3.4 *Discussion*

The previously discussed works on traceability all take the same assumption, namely that actors in supply chains are willing to share their data openly to achieve traceability. This shared data can contain business secrets, limiting the feasibility. The appliance of blockchain technology, as discussed in [73], provides anonymity. However, it has been shown that the technology as applied by Bitcoin does not provide full anonymity and is susceptible to attacks, breaking the anonymity [61]. Therefore, this property does not inherently hold.

The previous works on decentralized traceability approaches [1, 39, 73] clearly show that trust is an important concern in supply chains. Therefore, the works apply blockchain technology to distribute the trust. Moreover, the previous works on both centralized and decentralized traceability approaches have failed

to address the presence of privacy-sensitive information. Privacy-sensitive information encompasses both confidential data and the relationships between the actors. Since supply chains may contain competitors, confidential data is not able to be shared openly. Furthermore, exposing all relationships might provide competitors with an advantage. In the current state, most actors do not wish to disclose their relationships such as Adidas does [2].

In addition, the actors in a supply chain often obtain certificates to prove their compliance with a list of standards and requirements. In [1] certificates are taken into account in the form of digital signatures. For this, the identity of each actor is shown publicly without anonymity. Since the actor's profile is signed, he has no reason to remove the signature even though the certificate is later considered invalid or expired. This introduces an issue regarding the credibility and verifiability of the certificates for consumers. In [1] a solution has been proposed for certificate verifiability. For this, the identity of each actor is shown publicly without anonymity. Since the actor's profile is signed, he has no reason to remove the signature even though the certificate is later considered invalid or expired. This introduces an issue regarding the credibility and verifiability of the certificates for consumers.

The existing anonymization techniques for blockchains show three different techniques to achieve the untraceability and unlinkability properties. The cryptographic mixing of DASH in [18] uses monetization to prevent malicious behavior of Masternodes. However, this is infeasible for a traceability system due to the absence of monetization. Also, the mixing protocol contains the possibilities of side-channel attacks and bad mixes/peers [54]. In zk-SNARKs [64], anonymity is provided by using a novel zero-knowledge technique. However, the computational cost for the creation of the proof is significantly high resulting in poor performance. The RingCT protocol by Monero [55] provides plausible deniability regarding anonymity. While the RingCT protocol does not provide perfect anonymity, the computational costs are lower than zk-SNARKs and have no known attacks against it, to the best of our knowledge. Accordingly, the RingCT protocol is the most suitable solution for our proposed system and is thus used.

In conclusion, the existing works do not take privacy into account for the creation of a traceability system. The presence of privacy-sensitive data, confidential relationships and the need of verifiability show that existing works are insufficient for the current state. In the next section, we discuss these aspects in more detail.

2.4 OPEN QUESTIONS

Although previous works address the trust concern in supply chains, the works are theoretical and do not discuss feasibility. Besides, the works discussed in the previous section, fail to address the presence of privacy-sensitive information, such as manufacturing processes and product test results. Also, confidential relationships are not taken into account. Furthermore, the verifiability of certificates is forgotten [17] and approaches for product-specific auditability. These aspects are required to be taken into account to design a system that applies

to supply chains. In this section, we discuss each aspect and their importance within supply chains.

2.4.1 *Lack of Trust*

Besides the previous issues, trust is also an inevitable issue. The large group of actors in a supply chain currently provide a large amount of trust in each other. Rather, they provide full trust in their previous actor in the chain. By doing so, full trust is put in the entire chain before it reaches the current actor.

An actor might have an incentive to alter their data in the past. For example, in the case of a recall, or when problems are discovered with products. By altering the past, they can cover their tracks and remove the liability from their side. Besides, they can commit fraud on the data that they provide. There is no validation of their data, and the integrity of the data is not assured, resulting in malicious activities. Food safety and thus health safety might be at stake if an actor alters previous data. With improper management and a lack of traceability, it becomes difficult to pinpoint the cause of the problem. The result is an adverse effect on the brand/reputation of actors throughout the supply chains and even financial damage.

Moreover, the presence of privacy-sensitive data limits the possibility to achieve whole-chain traceability through a single system. The system is managed by a CA, which in turn requires the trust of all actors utilizing the system. The data submitted by the actors is, when submitted, in possession of the CA and thus can be altered. However, a significant risk is present with such a system if the CA participates in corruption or conspire with others. Even without the presence of a colluding or corrupt CA, there exists the risk that the CA might be compromised by an external (or internal) party. With this risk and the possibilities that the CA can take, changes can be made to the system with a significant effect on the network. These are all undesirable outcomes, in which trust lays at the base. Accordingly, a system is desired that minimizes the amount of trust and yet relies on the collaboration of the entire network.

2.4.2 *Privacy-Sensitive Information*

The data produced by actors in a supply chain contains information that is considered confidential to their business. The order of processes, or even the processes themselves, can reveal crucial data that gives a company their competitive advantage. Confidentiality is often regarded as a major difficulty to achieve collaboration in supply chains [53, 81]. Parts of the data are required to be shared to achieve the desired traceability. In the current setting, with separate systems, it is infeasible to a achieve a more efficient and whole-chain traceability system.

The privacy concerns regarding the shared data drive the resistance of many companies to participate. Data breaches could result in the release of proprietary information, the disclosure of business secrets or the loss of competitive advantage [28]. To ensure data confidentiality, a balance between the informa-

tion needs, the protection of intellectual property (IP) and vital business information is required.

A naive approach to solving the sharing of privacy-sensitive data could be the appliance of encryption, which hides the data for actors that should not be able to view it. The result is a system with encrypted data that still is opaque and takes more storage than storing data in plaintext. The approach, however, introduces new disadvantages, namely opaqueness and increased storage-complexity. Besides, to provide proof of authenticity and to show who the recipient is, the actor publishing the data should be revealed.

Besides the presence of confidential data, the relationships of actors are also considered confidential. These relationships are an essential aspect of the competitive advantage and exposing these might result in actors attempting to overtake relationships. For traceability, it is crucial to link products together along the supply chain, which also links the actors. However, with confidential relationships, this counterfeits the purpose. Therefore, the linkage between two actors is required to be broken for other actors to see, yet visible to the two actors. Furthermore, there is the requirement to follow a product along the supply chain and achieve a form of traceability. Due to the aforementioned contradiction, a novel approach is needed to achieve both requirements.

2.4.3 *Verifiability of Certificates*

Products purchased by consumers often contain labels or some proof of certification that indicates that the product complies with certain requirements. Examples of certificates are shown in Figure 2.3. These labels are often in the form of a label on the product. These labels are easily faked as shown in [3, 57]. The presence of fake product labels is increasing, resulting in consumers to lose trust in the original brand. The loss of trust directly affects the companies that produced the product. To our knowledge, no solution verifies the label or certificate on a product against the party that initially issued the label/certificate and considers the expiration date.

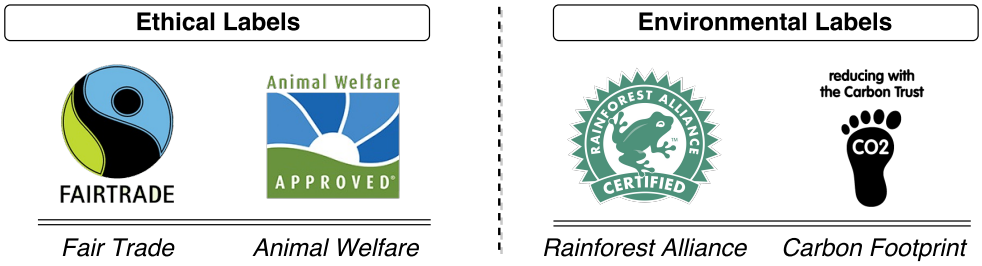


Figure 2.3: Examples of certificates, in the form of labels.

The focus on traceability has often been on tracking the entire life-cycle of a product, rather than aiming to provide the essential information. There exist a variety of certificate organizations that need a set of requirements to be fulfilled before obtaining a certificate. Companies are increasingly aiming to achieve certificates since this improves the image of the company as well as their possibility to achieve new business opportunities. The missing aspect is to utilize the cer-

tificates on products in a verifiable manner for consumers to check and (re)gain trust in the brand.

Some organizations, such as HarvestMark Traceability[2], provide the possibility to trace a product's life-cycle by scanning a barcode with the HarvestMark logo on it. The solution proposed by HarvestMark is also prone to faking since barcodes can be copied such as the HarvestMark logo. Also, the underlying system providing the information to consumers are often opaque and not open to consumers. Moreover, it relies on the assumption that the actors in supply chains are willing to share their data to trace it. However, the data produced by the actors contain privacy-sensitive information that is undesired to be shared.

2.4.4 *Product-specific Auditability*

In the case of a recall or disaster, data is retrieved regarding the affected products. Currently, this data is opaque to organizations, such as government agencies. When the data is retrieved, a full transcript is provided containing other information, which is undesired. From the previous works, discussed in Section 2.3, none address the auditability for their traceability systems. However, the size of the supply chain and the number of products it processes indicate that auditability should be taken into account.

During a recall, for example, government agencies and health organizations require insight into affected products. A full transcript of all products does not aid in the recall process. The opposite is achieved, it slows down the effectiveness to address the affected products. Therefore, specific information should be retrieved for auditability purposes.

2 http://www.harvestmark.com/solutions/item-level.aspx

3

PRELIMINARIES

The privacy and trust issues imposed by supply chains make it difficult to create a single system. In this chapter, we discuss previous attempt in literature to tackle these issues. Frameworks have been designed to achieve traceability in a supply chain [38, 39, 73, 82], but do not take the privacy issues into account. To answer the questions posed in Section 1.5 it is required to provide preliminary knowledge to propose and design a new system that takes the privacy and trust issues into account. This chapter provides that preliminary knowledge and discusses the techniques used in this research. We discuss the cryptographic primitives and protocols as well as blockchain technology as a decentralized, distributed ledger for the communication network. Finally, anonymization techniques are discussed that serve to provide privacy in supply chains.

3.1 CRYPTOGRAPHIC PRIMITIVES

In this section, we describe the cryptographic primitives and their specifications as used in this research. First, the two types of cryptographic schemes are discussed. Next, one-way functions and their appliance in digital signatures are discussed. Finally, Schnorr signatures are discussed and their usage.

3.1.1 *Cryptographic Encryption Schemes*

There exist two types of cryptographic schemes: symmetric and asymmetric (public key) cryptographic schemes. In the first, the same key is used for both the encryption and decryption. The latter utilizes two different keys for encryption and decryption, also respectively known as a *public key* and a *private key*. Besides, asymmetric cryptography provides the possibility for the creation of digital signatures.

SYMMETRIC CRYPTOGRAPHIC SCHEMES

In symmetric encryption, two parties, Alice and Bob, agree on a shared key k, which is further noted as the shared secret key. For the encryption Alice generates a ciphertext c by means of using an encryption function E_k, using the shared secret key k, on a message m. This can be noted as $c = E_k(m)$. For the decryption of c, Bob uses the decryption function D_k using the same shared secret key k. This results in recovering the plaintext message m. This can be noted as $m = D_k(c)$.

AES is widely adopted and has been chosen by NIST as the standard symmetric encryption algorithm [58]. It handles 128 bit block sizes with key sizes of 128, 192 or 256 bits. There has not been a practical cryptanalytic attack discovered against AES. In this research, we use AES as symmetric cryptographic scheme, with a key size of 256 bits.

ASYMMETRIC CRYPTOGRAPHIC SCHEMES

Asymmetric cryptographic schemes are also referred to as *public-key encryption schemes*. Each party is in possession of a public key pk_i with a corresponding private key sk_i. For Bob (B) to send a message m to Alice (A), he requires the knowledge of the public key of Alice, noted as pk_A. He uses the encryption function to obtain the ciphertext, $c = E_{pk_A}(m)$. For the decryption, Alice is the only one able to decrypt the ciphertext since she holds the private key sk_A. To decrypt c she computes the message m as $m = D_{sk_A}(c)$.

In asymmetric schemes, the private keys are required to be kept secret, while the public key can be made widely available for parties to send encrypted messages to the recipient. Figure 3.1 depicts the process of encryption and decryption in a simple manner.

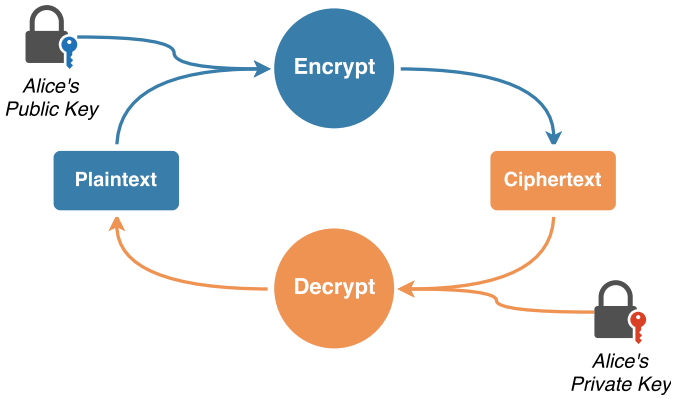


Figure 3.1: A public-key encryption and decryption process.

In this research, we utilize Elliptic Curve Cryptography (ECC) as asymmetric scheme. ECC requires smaller keys compared to non-ECC cryptographic schemes to provide an equal level of security. To achieve 128 bits security, ECC only requires a key length of $256 - 383$ in comparison with 3072 for the widely adopted RSA [37]. In the next section, we discuss ECC in more detail.

3.1.2 *Elliptic Curve Cryptography (ECC)*

ECC was introduced in 1985 as an alternative to other public-key cryptosystems [49] and is based on the algebraic structure of elliptic curves over finite fields. The security of ECC is primarily based on the difficulty of solving the discrete logarithm problem, which is considered a hard problem [76]. An elliptic curve can be defined as follows.

Definition 3.1.1. An *elliptic curve* is the set of points described by the following equation:

$$\{(x, y) \in \mathbb{Z}_p \mid y^2 = x^3 + ax + b, 4a^3 + 27b^2 \neq 0\} \cup \{0\}. \tag{3.1}$$

ECC considers a set of values which compromise the domain parameters, denoted as $D = (p, a, b, G, n, h)$. The domain parameters consist of an integer p specifying the field of the curve $\mathbb{F}_p$, two elements $a, b \in \mathbb{F}_p$ specifying an elliptic curve $E(\mathbb{F}_p)$ defined by the equation in Definition 3.1.1, a base point

$G = (x_G, y_G)$ on $E(\mathbb{F}_p)$, a prime n which is the order of G and an integer h which defines the cofactor of the curve. The domain parameters in D are available to all participants discussed further in this research. In the rest of this research, we use the curve $secp256k1$, equal to the one used in Bitcoin [59]. Below we define the operations for point operations on an elliptic curve and the generation of a key-pair. Next, we discuss the Elliptic Curve Integrated Encryption Scheme *(ECIES)* scheme for encryption and decryption of data.

POINT OPERATIONS

The basic operations for an elliptic curve is point addition and point doubling. In point addition, two points P and Q are added to result in the point $R = P + Q$. This can be denoted as $(x_p, y_p) + (x_q, y_q) = (x_r, y_r)$. The pre-requisite is that $P \neq Q$. In case this holds, the addition is performed as follows.

$$\begin{aligned} \lambda &= \frac{y_q - y_p}{x_q - x_p}, \\ x_r &= \lambda^2 - x_p - x_q, \\ y_r &= \lambda(x_p - x_r) - y_p. \end{aligned} \tag{3.2}$$

In the case of point doubling the calculations of x_r and y_r stay equal, except for the first calculation that is now as follows.

$$\lambda = \frac{3x_p^2 + a}{2y_p}. \tag{3.3}$$

KEY-PAIR GENERATION

For a party to create a public and private key-pair, a value $d \in_R [1, n-1]$ is chosen, which is considered the private key. Next, the public key Q is computed as $Q \leftarrow dG$, by using the base point G. The result is a key-pair (Q, d).

ELLIPTIC CURVE INTEGRATED ENCRYPTION SCHEME (ECIES)

ECIES is a hybrid encryption scheme to encrypt data between two parties, Alice and Bob [67]. The security of the scheme is based on the computational Diffie-Helman problem. A key derivation function (KDF), message authentication code (MAC) algorithm and a symmetric encryption scheme are required in addition to the domain parameters of the used elliptic curve and the recipients public key.

To encrypt a message M, Alice performs the algorithm as shown in the first procedure of Algorithm 3.1, resulting in a tuple $(c, R, hmac)$. Based on that tuple, Bob can decrypt it and retrieve the original message M as shown in the second procedure of Algorithm 3.1.

3.1.3 *One-Way Hash Functions*

Cryptographic hash functions map input of arbitrary length to a short fixed length output string. The output, generated by such functions, are called hash

Algorithm 3.1. ECIES

procedure ENCRYPTION(D, M, pk_B)
  $r \in [1, n-1]$, $R = rG$
  $S = P_x$, where $P = (P_x, P_y) = rpk_B$
  Derive symmetric encryption key and MAC key: $k_E \,||\, k_M = KDF(S)$
  $c \leftarrow E(k_E; M)$ ▷ E is the symmetric encryption method
  $d \leftarrow MAC(k_M; c)$
  return $(c, R, hmac)$
end procedure
procedure DECRYPTION($D, pk_B, (c, R, hmac)$)
  $S = P_x$, where $P = (P_x, P_y) = pk_B R$
  $k_E \,||\, k_M = KDF(S)$
  if $hmac \neq MAC(k_M; c)$ **then**
    return false
  end if
  $M \leftarrow E^{-1}(k_E; c)$
  return M
end procedure

values or simply hashes. Practical appliances of hash functions include *message integrity checks (MAC)*, *authentication* and *digital signatures*. Hash functions possess three properties, namely pre-image resistance, second pre-image resistance and collision resistance [15]. In this research, we use $SHA3$ as a one-way function, which is approved as a standard hashing function by NIST [19]. $SHA3$ is not susceptible to the length extension attack since it is not based on the Merkle-Damgård construction.

3.1.4 *Digital Signature Schemes*

Digital signature schemes are mathematical schemes for demonstrating the authenticity of digital data or documents. Digital signatures provide *authentication*, *non-repudiation* and *integrity* as properties and are made possible by public-key cryptographic schemes. To reduce the size, a message m is first hashed using a secure one-way function, resulting in h. h is then used in the signing algorithm, together with the signer's private key (sk_a) to create a digital signature. For the verification, m is hashed again using the same hashing algorithm to obtain h'. The verification algorithm is performed using the signer's public key pk_a, to obtain the original file hash and succeeds if h' is equal to h. In Algorithm 3.2 the algorithm for the generation and verification of a signature using elliptic curves (ECDSA) is shown.

3.1.5 *Zero-Knowledge Proofs*

Zero-knowledge proofs (ZKPs) are encryption schemes used to prove the knowledge of certain information without revealing the information [25]. Zero-knowledge proof schemes consist of two parties, the prover, and verifier. There exist two

Algorithm 3.2. ECDSA

procedure GENERATION(D, m, sk_a)
  $z \leftarrow H(m)$
  $k \in_R \mathbb{Z}_p$
  $(x_1, y_1) = kG$
  $r = x_1 \pmod n$
  $s = k^{-1}(z + rsk_a) \pmod n$
  return (r, s)
end procedure
procedure VERIFICATION($D, m, pk_a, (r, s)$)
  $z \leftarrow H(m)$
  $w = s^{-1} \pmod n$
  $u_1 = zw \pmod n$
  $u_2 = rw \pmod n$
  $(x_1, y_1) = u_1 G + u_2 pk_a$
  return $r \stackrel{?}{=} x_1 \pmod n$
end procedure

types of ZKPs, namely interactive and non-interactive ZKPs. The first requires interaction between the prover and verifier, whereas the second requires no interaction. Interactive ZKPs come with additional communication and computational costs. We now describe *Schnorr signatures*, a known non-interactive zero-knowledge proof scheme.

SCHNORR SIGNATURES The Schnorr signature scheme, using the Fiat-Shamir transform [21], provides a publicly verifiable non-interactive zero-knowledge proof. Given the domain parameters D of an elliptic curve E, a prover can prove the knowledge of x, for a public key $P = xG$, without revealing anything about it. However, everyone can validate the proof on x without the knowledge of x. We denote the procedures for the creation and validation of a Schnorr signature in Algorithm 3.3.

Algorithm 3.3. Schnorr Signature with Fiat-Shamir Transform

procedure CREATION(D, x)
  $k \in_R \mathbb{Z}_p$
  $e = H(kG)$
  $s \leftarrow k + xe$
  return (s, e)
end procedure
procedure VALIDATION($D, (s, e)$)
  $kG = sG - eP$
  return $e \stackrel{?}{=} H(kG)$
end procedure

Since H, a one-way function, returns a different e for different inputs, it is possible to add a message to the input of the Schnorr signature algorithm; for

example, computing $H(m \,||\, kG)$ for a message m. The result is a signature containing m, for which m cannot be altered without the knowledge of x, providing a *signature of knowledge* on m.

3.2 BLOCKCHAIN TECHNOLOGY

In 2008, Satoshi Nakamoto published a paper on the creation of Bitcoin [52]. Nakamoto is a person, or entity, of which the real identity is still unknown. Nakamoto created the world's first peer-to-peer (P2P) electronic payment system, namely Bitcoin. The idea for the popular Bitcoin was to create a payment system without the need of central minting authorities. The absence of central minting authorities allows for transparency, flexibility and the spread of trust. The underlying technology, blockchain, provides a system without the need of trust in a CA or TTP. The ownership of a transaction is proven cryptographically, rather than by an entity, like a bank in the traditional system [68].

In general, blockchain is a distributed database for transaction processing. All transactions in a blockchain are stored in a distributed ledger. This ledger is in possession of all parties in the network. The present state of the system in the blockchain technology is uniquely determined by the ledger [33, 83]. The architecture, for a new system of decentralized trustless transactions, is the key innovation of blockchain [68]. The blockchain technology can be split up in three core aspects: (i) decentralized, (ii) immutability and (iii) no trust [33]. Respectively, aspect (i) represents the absence of a single point of failure in the system. Point (ii) describes the disability to change previous transactions in the system. Every made transaction in the system is persistent, and therefore the state can not be changed. The last aspect (iii) provides reliance on algorithmically enforced rules to process transactions with no human interaction required.

This section discusses the concepts of blockchain in Section 3.2.1, the different modes possible in Section 3.2.2 and the different consensus models in Section 3.2.3.

3.2.1 *Building Blocks*

Blockchain contains several elements that make up the entire system as introduced by Nakamato in the original Bitcoin paper. These elements are discussed here, where the notions mentioned are taken from [52].

TRANSACTIONS Transactions are the transfer of information from one party to the other. Each transaction contains a source, a transaction message, and a destination. A valid transaction transfers the ownership of the asset from the source to the destination. Afterwards, the recipient is in control of the asset and can spend it. The spending of a transaction is possible by using the hash-value of the received transaction as input (source).

DIGITAL SIGNATURES Digital signatures are the cryptographic mechanisms used to chain the transactions together. They are used to verify the integrity and authenticity of a transaction and the corresponding source. It shows that

the transaction has not been altered and that is has been issued by the owner of the private key.

BLOCKS Blocks contain the aggregation of transactions that are timestamped. Blocks are hashed and linked together into the blockchain. Due to this, any small change in the content of a block is detectable since it would make the hash invalid and thus completely change all subsequent blocks. A block is made immutable after enough consecutive blocks have been validated and added into the blockchain. The block creation and the linkage are shown in Figure 3.2. S. Nakamoto solved the double spending problem by the usage of blocks connected to each other in the blockchain.

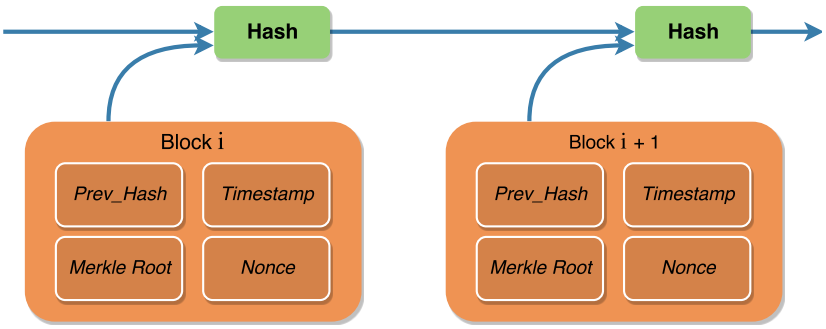


Figure 3.2: The content and linkage of blocks.

MERKLE TREES Merkle trees is a technique to reduce disk space and is used within the blockchain technology [48]. To verify a transaction, only the root hash and the transaction hash is required [40]. The verification is done by concatenating hashes to the top of the tree. The resulting root can be examined against the provided root hash.

CONSENSUS Each blockchain solution holds a consensus algorithm which is composed of the process of deciding, as a network, on one single correct blockchain in a decentralized network. There exist a variety of consensus algorithms, of which two are discussed in Section 3.2.3.

3.2.2 *Two-Dimensional Classification*

Current blockchain technology can be characterized as a two-dimensional classification, which is in the area of data access and validation privilege. In the first case, we have public and private blockchains. The latter case encompasses permissionless and permissioned blockchains. The two dimensions are combinable. The possible combinations are depicted in Table 3.1.

PUBLIC VS. PRIVATE BLOCKCHAINS Public blockchains, like Bitcoin, grant full access to the data stored in the blockchain which still may be encrypted. There also exists no restriction regarding who is allowed to participate and add new transactions to the network. Private blockchains restrict which parties are

Table 3.1: Four types of blockchain categories and their combinations.

Access to Transactions	Accesss to Transaction Processing	
	Permissioned	**Permissionless**
Public	Colored Coins protocols	Cryptocurrencies (e.g., Bitcoin)
Private	Limited access to transaction processors (i.e., opaque for clients)	Not applicable

allowed to submit transactions and access the data in the blockchain. Private blockchains are applicable within companies as the facilitator for a series of business flows [51].

PERMISSIONLESS VS. PERMISSIONED BLOCKCHAINS Permissionless blockchains are blockchains where users do not require permission to join. An example of such a blockchain is Bitcoin, where everyone can join the network and create or validate transactions. For users to join permissioned blockchains, permission is needed. These blockchains could form a more controlled and predictable environment than permissionless blockchains. Most of these blockchains are not controlled by one entity but rather a consortium of entities who permit others to use it.

3.2.3 *Consensus Models*

In this section, we discuss two consensus models, namely (i) Proof-of-Work (PoW) and (ii) Byzantine Fault Tolerance (BFT). Besides these, there exist a various amount of consensus models to achieve the goal of agreement in a distributed system such as, but not limited to, Proof-of-Stake (PoS), Proof of Elapsed Time and Federated Byzantine Agreement [6].

PROOF-OF-WORK (POW) PoW is the traditional consensus model as introduced and described by S. Nakamato in [52]. For the creation of a new block, containing any number of transactions, it is required that a *miner* creates a block header. In the case of Bitcoin, a block header H contains the previous block's header H_{n-1}, the Merkle root hash M_i, the timestamp t and a nonce i, see Figure 3.2. It is the goal for the *miner* to find a nonce i which results in the hash of the block header to meet a certain difficulty:

$$\begin{aligned} h(H_n) &= h(H_{n-1} \mid\mid M_n \mid\mid t_n \mid\mid i), \\ h(H_n) &< 2^{256-m}. \end{aligned} \tag{3.4}$$

The entire network of miners performs random guesses for the nonce to meet the difficulty. This results in the chance of a correct guess to be higher. PoW requires miners to invest a significant amount of CPU/GPU power to be able to create a new block. Due to this, it is called proof-of-work. Nonetheless, the

verification if a nonce is correct, is fast. PoW comes with two important downsides: (i) it results in a high energy consumption [14, 43] and (ii) the *Tragedy of the Commons* problem [29].

BYZANTINE FAULT TOLERANCE (BFT) The BFT consensus model is derived from the Byzantine Generals Problem (BGP), first introduced in 1982 by Lamport et al. [78]. BFT is a state-machine replication protocol which promises consensus despite the participation of malicious (Byzantine) nodes. It aims to achieve a consensus for the distributed network. In this network, potentially faulty nodes are present. In this consensus protocol, Byzantine faults are tolerated, and the network can withstand f number of Byzantine nodes, where N is the total amount of nodes in the network. Now, the following holds:

$$f = \frac{N-1}{3}. \tag{3.5}$$

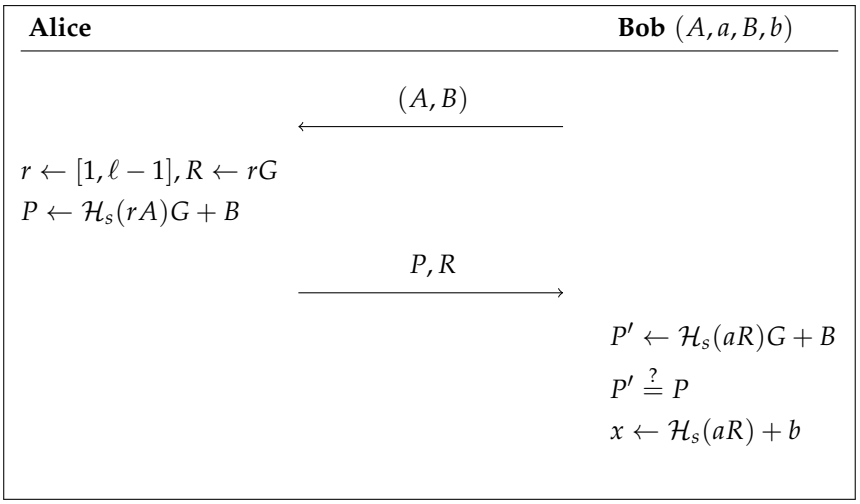
Due to this, BFT provides a minimum of $2f + 1$ nodes to reach consensus. Practical Byzantine Fault Tolerance (PBFT) [11], an often implemented variant of BFT provides *safety* and *liveness* as properties. The first ensures that the system maintains state and looks to the node like a non-replicated remote service. Safety includes a total ordering of requests. The latter property ensures that nodes will eventually receive a reply to every request sent, provided network is functioning.

3.3 THE RINGCT-PROTOCOL

The RingCT-protocol provides the ability to hide the identity of the sender, receiver and the amount that is being transferred. The basic idea is to hide the sender in a group of actors, the recipient's identity is hidden through *Stealth Addresses*, and the amount is hidden through Pedersen commitments [63]. In the RingCT-protocol, each actor holds an address, denoted as two public ec-keys (A, B) and the private corresponding ec-keys (a, b) where $A = aG$ and $B = bG$.

STEALTH ADDRESSES Stealth addresses is a technique, based on ECC, where a user publishes a single address and receives unconditional unlinkable payments [63]. Stealth addresses create a pair P, R where the first denotes the destination key and the latter the transaction public key. The protocol uses $\mathcal{H}_s$, which is a cryptographic hash function $\{0,1\}^* \to \mathbb{F}_q$. The recipient is required to check every transaction if it is meant for him by using the private key a and public key B. The spending key is derived by using both private keys a, b of the recipient, denoted as $x = \mathcal{H}_s(aR) + b$. The complete protocol is shown in Protocol 3.1.

HIDING TRANSACTION AMOUNTS Transaction amounts are hidden through Pedersen commitments. For every transaction amount a a random scalar α, further noted as a mask, is chosen. Furthermore, a second generator H is used which is required to be public and unrelated to G. So, no obvious relation should exist ($aG = H$). The Pedersen commitment is now generated as follows,



Protocol 3.1: Stealth Addresses Protocol

$$C = \alpha G + aH. \tag{3.6}$$

A range proof is applied on the Pedersen commitment in order to prevent an actor to cheat the system, by wrapping the value around the modulus, i.e. $1 - 9 - 5 = -13 \equiv 0 \pmod{13}$. The amount a is split into denominations of 2^ℓ, where ℓ varies from 1 to k. It ensures that a is in the range $2^0 \leq a \leq 2^{k-1}$. Next, k Pedersen commitments are created as shown in Algorithm 3.4.

Algorithm 3.4. Range Proof

procedure CREATION(G, H, a) ▷ Transaction amount and generators
  $\alpha \in_R \mathbb{Z}_p$
  $C = \alpha G + aH$
  Binary representation of a, $a = b_0 + 2b_1 + \cdots + 2^{k-1}b_{k-1}$
  $\forall 0 \leq i < k - 1$ pick $\alpha_i \in_R \mathbb{Z}_p$, let $\alpha_{k-1} = \alpha - \sum_{i=0}^{k-2} \alpha_i$
  For all i, commit $C_i = \alpha_i G + 2^i b_i H$
  return all C_i's
end procedure

The set $A_i = \{C_i, C_i - 2^i H\}$ is created and used as the public keys for a ring signature. The binary expansion of a is used, and this gives us the knowledge of the private key to precisely one of the public keys in A_i. This is because

$$\begin{aligned} b_i = 0 \Rightarrow C_i &= \alpha_i G + 0H = \alpha_i G, \\ b_i = 1 \Rightarrow C_i - 2^i H &= \alpha_i G + 2^i H - 2^i H = \alpha_i G. \end{aligned} \tag{3.7}$$

To prove that $b_i = 0$ or 1, we construct a ring signature over A_i. A verifier cannot determine which key has been used for the signing process since ring signatures are signer-ambiguous This results in the ability to hide all the bits while proving that they are indeed bits. The final range proof for a is denoted as follows, where s_0 and $\overline{s_0}$ are the s-values of the i^{th} ring signature,

$$R_a = (C_0, ..., C_k, e_0, s_0, \overline{s_0}, ..., s_k, \overline{s_k}). \tag{3.8}$$

To gain space savings, Borromean ring signatures are applied on a set of commitments $C_i \in R_a$ [46]. We denote the set of commitments as $\mathcal{A}_\rangle = \{C_i, C_i - 2^i H\}$, where i as described above. We further denote the result, the Borromean ring signature on a, as BS_a. The algorithm for the creation and validation of Borromean ring signatures is provided in the research by Maxwell et al. in [46].

MLSAG: MULTI-LAYERED LINKABLE SPONTANEOUS ANONYMOUS GROUP

The anonymization of the sender of a transaction is achieved by employing MLSAG ring signatures, which is similar to the LSAG described in [42]. The main difference is that in MLSAG a ring signature is created on a set of n key-vectors, rather than on a set of n keys as in LSAG. The intent of the MLSAG ring signature is

- to prove that one of the n signers knows the secret keys to their entire key vector and
- to enforce that any of the m signing keys of the signer is linked if it is used in another MLSAG signature.

The protocol for the generation and validation of ring signatures consist of two parties: the signer and verifier. MLSAG utilizes a function $\mathcal{H}_p$ that takes as input a point and hashes it onto another point on the curve. There exist a variety of methods to achieve this, discussed in [31]. Each signer of a ring containing n members has exactly m keys $\{P_i^j\}_{j=1,...,m}^{i=1,...,n}$. We describe two phases: **SIGN** and **VER**. Below the phases are described with their accompanying algorithms.

SIGN: Let m be a given message and π a secret index corresponding to index of the signer of the generalized ring. For $j = 1, \ldots, m$ let $I_j = x_j \mathcal{H}_p(P_\pi^j)$ and for $j = 1, \ldots, m$, $i = 1, \ldots, \hat{\pi}, \ldots, n$ let s_i^j be random scalars ($\in_R \mathbb{Z}_q$). Here, $\hat{\pi}$ means to omit the index π. Now, define for random scalars α_j and $j = 1, \ldots, m$,

$$\begin{aligned} L_\pi^j &= \alpha_j G, \\ R_\pi^j &= \alpha_j \mathcal{H}_p(P_\pi^j), \end{aligned} \tag{3.9}$$

followed by the computation,

$$c_{\pi+1} = \mathcal{H}_p(m, L_\pi^j, R_\pi^j, \ldots, L_\pi^m, R_\pi^m). \tag{3.10}$$

Then, we compute the following values while working successively in i modulo n for each $j = 1, \ldots, m$.

$$\begin{aligned} c_i^j &= \mathcal{H}_p(m, L_i^j, R_i^j, \ldots, L_i^m, R_i^m), \\ L_i^j &= s_i^j G + c_i P_i^j, \\ R_i^j &= s_i^j \mathcal{H}_p(P_i^j) + c_i I. \end{aligned} \tag{3.11}$$

Afterwards, the last c-value is computed as follows.

$$c_\pi = \mathcal{H}_p(m, L_{\pi-1}^1, R_{\pi-1}^1, \ldots, L_{\pi-1}^m, R_{\pi-1}^m). \tag{3.12}$$

Finally, solve for each s_π^j by using $\alpha_j = s_\pi^j + c_\pi x_j \bmod \ell$. The signature is then given as:

$$(I_1, \ldots, I_m, c_1, s_1^1, \ldots, s_1^m, \ldots, s_n^1, \ldots, s_n^m). \tag{3.13}$$

VER: In the verification phase, further denoted as VER_MLSAG, the verifier computes L_i^j, R_i^j and c_i for all i and checks that $c_{n+1} \stackrel{?}{=} c_1$. The final check is performed by validating c_{i+1} as follows, for all $i \bmod n$,

$$c_{i+1} \stackrel{?}{=} \mathcal{H}_s(m, L_i, R_i). \tag{3.14}$$

If these equalities hold, the verifier checks if the key images I_j have been used in past signatures ($I \in \mathcal{I}$). Here, $\mathcal{I}$ denotes a list of all key images, held by the actors in the system. If a duplicate has been found, this is considered an attempt of double spending, and therefore the signature is rejected.

4

TRADE: A TRANSPARENT, DECENTRALIZED TRACEABILITY SYSTEM[1]

Traceability has become an increasingly important aspect of supply chains in the last few years, due to customer awareness as well as better planning and problem identification. Unfortunately, technological, legal, and organizational concerns limit the possibility to achieve traceability. Trust is one of the most important factors that prevent involved parties to build such a system.

In this chapter, we address the aforementioned trust concern. We propose a traceability system, called TRADE, that achieves traceability in a transparent and decentralized manner. TRADE is, to the best of our knowledge, the first feasible solution, as a fully transparent traceability system, for supply chains. We assume minimal trust between the actors, requiring a system that ensures that no single actor is in control of the system. Only authorized actors can participate, view and add information to the system. The authorization to the system is handled by a central authority (CA).

This chapter is constructed as follows. Firstly, the system model is introduced and its accompanying components. Secondly, we introduce the TRADE system. Thirdly, we discuss the validation of the transactions and blocks. Fourthly, we analyze the system regarding security and privacy, computational and communication complexity. Lastly, we provide a discussion of our proposed system. In the continuation of this chapter, we denote $x[y]$ as the element y in the set x and `INRF` as "if not, return false".

4.1 SYSTEM MODEL

We assume there exist a network of five types of actors, namely Producers, Processors, Transporters, Distributors, and Retailers. We assume that a Producer creates a product and then transports it via a Transporter to a Processor. A Processor performs internal processes on the product, which is further transported to a Distributor via a Transporter. The Distributor then distributes the end product to its final destination: a Retailer. Figure 4.1 depicts the relation among the actors.

Figure 4.1: Relationship view of actors.

The actors create transactions that are then broadcasted directly to the other actors in the network. The transactions are considered valid if they fulfill a

1 Submitted as a scientific paper to the 2^{nd} ACM workshop on Blockchains, Cryptocurrencies and Contracts (BCC'18), held in conjunction with ACM AsiaCCS 2018.

set of requirements. We further denote the checking of these requirements as validation. The entire network validates the broadcasted transactions. We use a blockchain, denoted as BC, as a decentralized solution for TRADE. The genesis block of BC is denoted as BC_{gb}. Figure 4.2 depicts a schematic flow diagram of data between actors in TRADE.

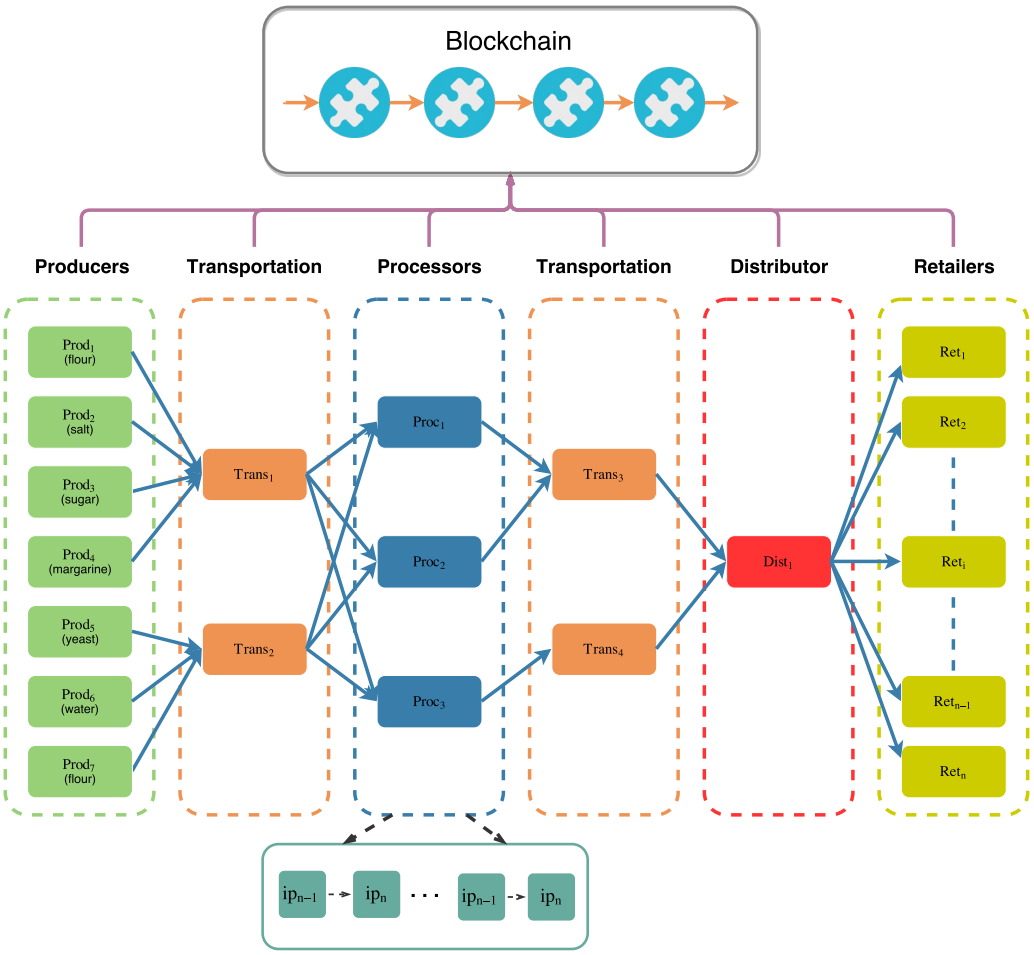


Figure 4.2: A schematic flow of our proposed system.

Each transaction is denoted as tx_h, where h denotes the hash-value of the transaction. The transaction tx_h is in the form of a tuple $tx_h = \langle a, p_{id}, in, out, info, t, Sig(tx_h)\rangle$. The transaction structure and its description is shown in Table 4.1.

4.2 TRADE

INITIALIZATION Each actor in the network performs the key-pair generation algorithm for ECDSA, as discussed in Section 3.1.2. The key-pair is denoted as (pk_a, sk_a), where a is the actor. The public key of each actor is shared with the CA. In case an actor does not hold pk_a of the actor that signed the transaction, the CA is consulted. Furthermore, each actor holds a list PID, which contains all the p_{id}'s. This list is used to check that any new registered p_{id} is unique.

Table 4.1: Transaction Structure

Field	Description
a	Actor issuing the transaction.
p_{ID}	Unique ID for a product.
k	Number of products.
in	Hash of the previous transaction.
out	Receiver of the transaction.
$info$	List of additional information.
t	Date and time of the transaction.
$Sig(tx_h)$	Signature of a on the transaction.

PRODUCTION A Producer, denoted as $PD_i \in PD$, creates a product with a unique ID p_{id}. Afterwards, the Producer creates the additional information $info = \{dest\}$, where $dest$ is the Processor $PS_i \in PS$. Since the Producer creates a new product for the supply chain, he has no previous transaction to link it to and thus links it to the genesis block BC_{gb}. The final transaction is created as $tx_h = \langle PD_i, p_{id}, BC_{gb}, T_j, info, t, Sig_{PD_i}(tx_h)\rangle$, where out is set as the Transporter $T_j \in T$. The validation of a transaction by a Producer is shown in Algorithm 4.1.

Algorithm 4.1. Transaction Validation: Producer

```
1: procedure VALIDATION_PRODUCER(tx_h)
2:     Check in = BC_gb; INRF.
3:     Check p_ID ∉ PID; INRF.
4:     Check info[dest] ∈ (PS_i ∈ PS); INRF.
5:     return true
6: end procedure
```

TRANSPORTATION The Transporter, denoted as $T_i \in T$, creates a transaction tx_h upon placing the product in a means of transportation. He receives a product from a Producer or Processor and transfers it to a Processor or Distributor respectively. The additional information is set to $info = \{src, dest, V_{ID}, SSCC\}$, where src is the actor that provided the product, $dest$ is the destination actor, V_{ID} is the vehicle ID for transportation and SSCC is the Serial Shipping Container Code in which the product is placed, defined by *GS1* [65]. The complete transactions is denoted as $tx_h = \langle T_i, p_{id}, k, in, out, info, t, Sig_{T_i}(tx_h)\rangle$, where out is either a Processor or Distributor, based on the received transaction. The validation of a transaction by a Transporter is shown in Algorithm 4.2.

PROCESSING A Processor, denoted as $PS_i \in PS$, performs internal processes on p_{id}, such as combining materials, testing or sanitizing the product. The Processor sets the additional information to $info = \{dest, IP\}$, where $dest$ is the recipient, which is a Distributor. The complete transaction is denoted as $tx_h = \langle PS_i, p_{id}, k, in, T_j, info, t, Sig_{PS_i}(tx_h)\rangle$, where in corresponds to the hash

Algorithm 4.2. Transaction Validation: Transporter

1: **procedure** VALIDATION_TRANSPORTER(tx_h)
2: **if** $in \in PD$ **then** Check $info[dest] \in PS$; `INRF`. **end if**
3: **if** $in \in PS$ **then** Check $info[dest] \in D$; `INRF`. **end if**
4: Check $out = info[dest]$; `INRF`.
5: **return** true
6: **end procedure**

value of the previous received transaction and out is set to the recipient Transporter $T_j \in T$. The validation of a transaction by a Processor is shown in Algorithm 4.3.

Algorithm 4.3. Transaction Validation: Processor

1: **procedure** VALIDATION_PROCESSOR(tx)
2: Check $info[IP] \neq \emptyset$; INRF.
3: Check $info[dest] \in D$; INRF.
4: Check $out = (T_j \in T)$; INRF.
5: **return** true
6: **end procedure**

DISTRIBUTION A Distributor, denoted as $D_i \in D$, creates a transaction upon distribution of p_{id}. The additional information is set to $info = \{src, V_{ID}, SSCC\}$, where src is the Processor that sent the product to D_i and recall the definition of V_{ID} and SSCC as mentioned before. The complete transaction is then set up as $tx_h = \langle D_i, p_{id}, k, in, out, info, t, Sig_{D_i}(tx_h)\rangle$, where out is set to a Retailer $R_j \in R$. The validation of a transaction by a Distributor is shown in Algorithm 4.4.

Algorithm 4.4. Transaction Validation: Distributor

1: **procedure** VALIDATION_DISTRIBUTOR(tx)
2: Check $info[src] \in PS$; INRF.
3: Check $out \in (R_j inR)$; INRF
4: **return** true
5: **end procedure**

RETAILER The Retailer, denoted as $R_i \in R$, is the end-actor that eventually sells the received products. This actor does not create a transaction. Therefore, retailers do not actively participate in the system, but rather function as an end-station for the products throughout the supply chain.

4.3 VALIDATION

4.3.1 *Validation of Transaction Authenticity*

Digital signatures are applied to prevent forgery or false transactions in TRADE. Each transaction tx_h is signed by the creator of the transaction using the private key sk_a, where a is the actor. Anyone with the corresponding public key pk_a can validate the signature, providing the ability for anyone to confirm that a has signed the transaction and no-one else. We assume that a is the only party that is capable of signing tx_h since he is the only one in possession of sk_a. The integrity of a transaction is held since an altered transaction results in an invalid digital signature. An invalid signature results in an invalid transaction.

4.3.2 *Validation of Transactions*

The validation of a transaction is dependent on the actor that created the transaction. Recall that a transaction is a tuple containing multiple fields, as shown in Table 4.1. Actors, upon receiving a transaction, need to check each field of the transaction. In Algorithm 4.5, we combine our previous proposed algorithms in a single algorithm to validate a transaction.

Algorithm 4.5. Transaction Validation

1: **procedure** VALIDATION_TX(tx_h)
2: Check $\forall x \in tx_h, x \neq null$; INRF.
3: Check that the $tx_h[t]$ < current timestamp; INRF.
4: Validate digital signature of tx_h.
5: Check Validation_Producer(tx_h) = true; INRF.
6: Check Validation_Transporter(tx_h)) = true; INRF.
7: Check Validation_Processor(tx_h) = true; INRF.
8: Check Validation_Distributor(tx_h) = true; INRF.
9: **return** *ret*.
10: **end procedure**

4.3.3 *Validation of Blocks*

A number of transactions are collected and aggregated in a block, which is broadcasted to the network and requires validation. Note that the validation of a block is different than the validation of a transaction. The block structure is similar to the one described in Bitcoin[2]. Let b be a block and $b[TX]$ be the transaction list in b. We propose an algorithm, described in Algorithm 4.6, that validates a block.

2 Bitcoin block structure: https://en.bitcoin.it/wiki/Block

Algorithm 4.6. Validation of a Block

```
1: procedure BLOCK_VALIDATION(b)
2:     Check the syntactic correctness of b.
3:     Check that no duplicate of b exists.
4:     Check length of b[TX] > 1; INRF.
5:     Validate Merkle root.
6:     for each tx_i ∈ b do
7:         Check Validation_TX(tx_i) = true; INRF.
8:     end for
9:     Relay block to nodes.
10:    return true.
11: end procedure
```

4.4 EVALUATION

In this section, TRADE is evaluated in three dimensions: security, performance, and scalability. First, we discuss the security imposed by the system. Next, we provide a theoretical analysis of the performance of the computational and communication complexities. Finally, we discuss the measurements obtained from a proof-of-concept implementation to show the actual performance of the system.

4.4.1 *Security Analysis*

TRADE does not allow any unauthorized participation since it uses a public permissioned blockchain. The consensus model provides the integrity of the block structure, and the signature algorithm secures the transactions. For TRADE, the consensus model preserves the integrity of a propagated block. TRADE does not enforce a specific consensus model. There are several models available that can be used for our system [44, 84]. The security of the blocks is thus dependent on the chosen consensus model.

TRADE uses digital signatures to provide authenticity and integrity of each transaction, where ECDSA is used as the digital signature scheme. The security of ECDSA relies on the elliptic curve discrete logarithm problem (ECDLP), which is considered to be computationally hard [34]. Therefore, the security of a digital signature, and thus the transaction, is kept under the ECDLP assumption.

4.4.2 *Computational Complexity*

For the analysis of the computational complexity, we list the number of operations performed by each actor in three aspects: (i) the creation of transactions, (ii) validation of transactions and (iii) the validation of blocks. The amount of performed operations depends on a number of variables, listed in Table 4.2.

Recall that a transaction consists of a set of values. The only computed value is the digital signature. Therefore, we focus on the computation complexity of

the digital signature scheme. In Table 4.3, the amortized number of operations for the aforementioned aspects are listed.

Table 4.2: Parameters used in the computational analysis.

SYMBOL	DESCRIPTION
$\mathcal{N}$	Number of actors in the network.
γ	Number of transactions per minute, by an actor.
ℓ	Number of transactions in a block.
s	Key-size in bits for the elliptic curve.

TRANSACTION CREATION For the creation of a transaction, a digital signature is created. The digital signature procedure is dependent on the key-size s for the chosen elliptic curve. The computational complexity, per transaction, is thus linear in s.

TRANSACTION VALIDATION The computational complexity of the validation of a transaction depends on the digital signature. The validation procedure of a digital signature is, equal to the creation, dependent on the key-size s.

BLOCK VALIDATION The validation of a block has the highest computational complexity. Firstly, the Merkle root is required to be validated, which requires multiple hashing operations and is computed in $\log(\ell)$ [70]. Then, each transaction is validated inside the block. The verification of ℓ digital signatures requires $s\ell$ verifications per block. Since $s\ell \gg \log(\ell)$ for $\ell > 1$, the block validation procedure is dominated by the validation procedure of digital signatures. Consequently, the block validation has a computational complexity of $\mathcal{O}(s\ell)$.

Table 4.3: Computational complexity in TRADE.

PROTOCOL	ACTOR
Transaction Creation	$\mathcal{O}(s)$
Transaction Validation	$\mathcal{O}(s)$
Block Validation	$\mathcal{O}(s\ell)$

4.4.3 *Communication Complexity*

To analyze the communication complexity of TRADE, we list the number of communications required on the network for the broadcast of a transaction and a block. The required communication depends on a number of variables in Table 4.2.

In the initialization phase, each actor sends their public key to the CA and requires $\mathcal{N}$ communication rounds. This procedure only re-occurs if an actor updates their key-pair. The public keys of the actors are stored locally by each actor to reduce the communication rounds necessary. Next, each transaction is broadcasted to the network, which requires $\mathcal{N} - 1$ rounds with the assumption that each actor knows each other and their addresses on the network allowing a direct connection. The same applies to the broadcast of blocks. Since the initialization phase only occurs at the beginning of the system, the communication complexity is dominated by the broadcast procedure for transactions and blocks. Therefore, the communication complexity of TRADE is $\mathcal{O}(\mathcal{N})$.

4.4.4 *Experimental Results*

To measure the runtime of TRADE, we created a proof-of-concept implementation of the system in `Python 2.7` by creating a simple blockchain implementation based on the work of Daniel van Flymen[3] and the `fastecdsa` package[4]. The p_{id} values are represented as 32-bit fixed-point numbers.

The measures of the runtime were executed on our commodity hardware, running macOS 10.13 on a dual-core 3^{rd} generation 2.9GHz Intel® Core i7 processor with 16GB RAM. We measured the runtime for the transaction and validation of a transaction. For accurate measurements, we executed 1000 iterations for each procedure. We use the NIST P-curves for our measurements. Figure 4.3 shows the impact of s on the runtime for transaction creation and validation. It is clear that the procedures grow quadratically based on s. Using $s = 256$ for an elliptic curve, each actor is able to create approximately $\frac{1}{2.84 \cdot 10^{-3}} = 351$ transactions per second and validate transactions at a speed of $\frac{1}{2.28 \cdot 10^{-3}} = 437$ transactions per second. For the latter, an actor can validate $437/\ell$ blocks per second, depending on ℓ.

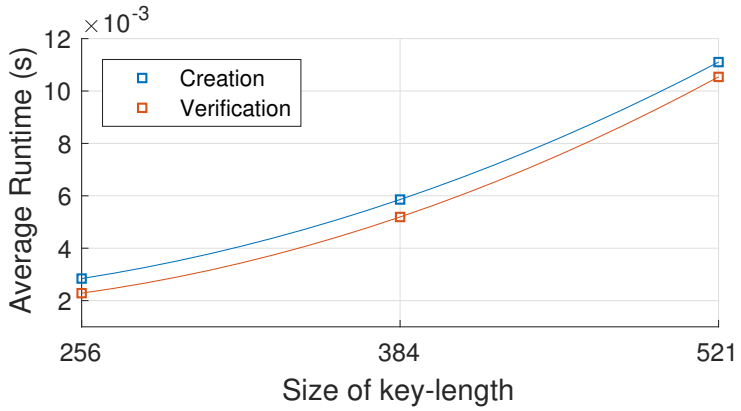


Figure 4.3: Average computation time for the creation and validation of a single transaction, based on s.

3 A simple Blockchain Implementation, https://github.com/dvf/blockchain
4 fastecds: https://pypi.python.org/pypi/fastecdsa

There are approximately 32.9 million shipping containers globally as of 2013 [80]. Based on the assumption that a container changes possessor up to 100 times per year, and for each time a transaction is made, approximately 317 transactions are made per second. The supply chain requires fewer transactions per second than all containers globally. Given our experimental results, it is clear that our system achieves the required performance to be applied in a real-world setting.

4.5 DISCUSSION

In previous works [1, 39, 73], researchers proposed several frameworks to achieve a decentralized, traceability system. The previous works are purely theoretical, and thus do not provide any implementation. Furthermore, no complexity of the approaches is given, and thus the feasibility of the previous works is missing.

In this chapter, we proposed TRADE, a single, traceability system for actors in the supply chain to share data built upon blockchain technology. Each actor creates a transaction regarding a product p_{id} containing the full information on the product. The stored data inside a transaction is fully transparent allowing each actor in the network to view the data. Each transaction is signed by the issuing actor using a digital signature, providing a proof of authenticity. The valid transactions are aggregated in a block and broadcasted to the network. Each transaction regarding a product p_{id} is linked throughout the supply chain on the blockchain, providing full traceability and insight for each actor. The insight on the data can be used to improve planning and scheduling, and faster recalls for the supply chain. In addition, consumers can view this data and gain insight into the full life-cycle of products.

TRADE achieves a significant performance to create and validate transactions, as well as the validation of blocks. We show that it is feasible to apply blockchain technology for the supply chain to achieve traceability. Moreover, consumers and other parties can view the data to gain knowledge on the procedures performed on their product as well as information on the sustainability, if the actors provide it. Actors are in control to share such information, which is recommended since it aids the company brand and increases the trust of consumers in the company. In case actors are willing to share data in a single system and achieve full traceability, blockchain technology is shown feasible to accomplish this in a real-world setting for the supply chain.

Although TRADE provides full transparency for the supply chain to achieve traceability, the viewable data can be used by competitors to gain a competitive advantage since the supply chain contains multiple competitors of actors. Any information that is considered confidential for the business of an actor can be extracted and used by other actors. Also, the throughput of actors can be extracted by viewing the timestamp of transactions and the transaction amount within a transaction. The throughput can be used by actors to approach clients of actors and attempt to take over their business. The clients of actors, thus the relationships of actors, can be easily derived by viewing the recipients and destined actors of every transaction in the system. These relationships are confi-

dential for businesses in the supply chain and are wished to remain confidential, whereas in TRADE these are fully exposed. Moreover, the system does not provide the ability to store confidential data.

5

DECOUPLES: A DECENTRALIZED, UNLINKABLE AND PRIVACY-PRESERVING SYSTEM[1]

Besides the trust concern, privacy is an important aspect of achieving a traceability system for supply chains. The previous chapter introduced TRADE, a fully transparent system. However, this system does not address the presence of privacy-sensitive information, achieve certificate verifiability and product-specific auditability. Besides TRADE, previous works [1, 39, 73] also did not take these aspects into account.

In this chapter, we introduce DECOUPLES, a decentralized, unlinkable and privacy-preserving traceability system. Besides, we propose PASTA, a Product-Auditable STealth Addresses protocol. The protocol allows anonymization of receivers while allowing for auditability. The system is, to the best of our knowledge, the first, feasible traceability system that takes privacy into account for supply chains. The goal of the system is to preserve the privacy requirements for the actors and providing a means for traceability. Each product is tracked in the system, enabling the possibility for batches, while actors in the system are anonymized. We use non-interactive zero-knowledge proofs to prove that an actor holds a certificate, allowing for certificate verifiability. In our proposed system only authorized actors can participate in the network, yet everyone can view the transactions stored on the blockchain.

This chapter is constructed as follows. Firstly, we introduce DECOUPLES and our proposed product-specific, auditable protocol. Secondly, we discuss the creation and validation of transactions and blocks in our proposed system. Thirdly, we state our contributions achieved by DECOUPLES. Lastly, the evaluation of security, privacy, and performance of the system is provided in Chapter 6.

5.1 DECOUPLES

We assume that there exist a network of five types of actors, as in TRADE. These actors create transactions for every phase, where the transactions are eventually added to the blockchain. DECOUPLES uses a public permissioned blockchain as the communication network. We assume that the technology to accept and connect to the blockchain is in place. However, in contrast to TRADE, we have set up a set of requirements. The requirements are as follows.

1. Untraceability,
2. unlinkability,
3. hidden relationships,
4. anonymous certificates to hide the owner and
5. hide privacy-sensitive information.

1 Submitted as a scientific paper to the 16^{th} International Conference on Applied Cryptography and Network Security (ACNS) 2018

We achieve the requirements mentioned above as follows. We achieve (1) by applying MLSAG ring signatures [55], (2) by the appliance of our proposed PASTA protocol, (3) is a result of the combination of MLSAG ring signatures and the PASTA protocol, (4) due to Schnorr signatures [21] and (5) by using ECIES as an encryption scheme [67].

In this chapter, we propose a complete design for a traceability system for supply chains and a product-specific auditable protocol, namely PASTA. The proposed system, DECOUPLES, is the first privacy-preserving traceability system for supply chains, to the best of our knowledge. Now, we design DECOUPLES that achieves the aforementioned requirements. For the continuation of this chapter, we denote $x[y]$ as the element y in a set x and `INRF` as "if not, return false".

5.1.1 *Initialization*

Each actor in the network holds a unique identifier a_{id}, the actor type a_{type} and two elliptic curve key-pairs (A_a, B_a), where a denotes the actor. The key-pairs are used as an address and made publicly available to all the actors in the system. Furthermore, each actor holds a set of certificates. Certificates are issued by a certificate organization (CO), whereas the actor publishes a set of information regarding the certificate. We now discuss the certificate processes.

CERTIFICATE ISSUING We propose a data structure for certificates containing primary information, derived from the following certificates: *Fair-Trade, Rainforest Alliance* and the *Child Labor Free* certificate. The data structure is shown in Table 5.1. The creation of the public key P is discussed in Section 5.1.2. The value h, of a certificate $Cert_j$, is computed as:

$$\begin{aligned} Cert_j[h] &\leftarrow \mathcal{H}_s(Cert_j[name] \mid\mid Cert_j[\lambda]), \text{ where} \\ \lambda &\leftarrow Cert_j[aid] \mid\mid Cert_j[sn]. \end{aligned} \tag{5.15}$$

Here, $\mathcal{H}_s$ denotes a cryptographically secure hash function equal to the one used in the RingCT-protocol. The actor is linked to the certificate by λ. Each CO stores the data of a certificate in their database that we assume is queryable by any party.

Table 5.1: Certificate Structure.

FIELD	DESCRIPTION	FIELD	DESCRIPTION
$name$	Name	cid	Issue date
co	Organization	ed	Expiration date
sn	Serial number	P	Public key
a_{id}	Actor ID	$info$	Extra information
cat	Category	h	Hash

INITIAL CERTIFICATE TRANSACTION For each certificate that an actor holds, a set of information is stored in a transaction. The transaction contains the name of the CO, the certificate name, a hash of the certificate, the public key of the certificate and the digital signature by the CO. The transaction is checked before being stored on the blockchain. We propose a validation algorithm for the certificate transaction in Algorithm 5.1, where *Query* denotes a query to the database of the CO.

Algorithm 5.1. Initial Transaction

procedure VALIDATE_CERTIFICATETRANSACITON($CO_{name}, Cert_j$)
  $CO_{pk} \leftarrow$ public key of CO_{name}
  Check if $Sig_{CO_{pk}}(Cert_j)$ is valid; `INRF`.
  $s, name, ed \leftarrow Query(Cert_j[h])$
  if $s \neq Cert_j[P]$ **or** $name \neq Cert_j[name]$ **then**
    return false
  end if
  if $ed <$ current date & time **then**
    return false
  end if
  return true
end procedure

5.1.2 *Hiding the Owner of a Certificate*

To prevent that the CO can create proofs on behalf of the recipient, we create a stealth address for a certificate $Cert_j$. The stealth address is generated according to the original protocol in Protocol 3.1, resulting in a key-pair $(Cert_j[R], Cert_j[P])$. The key-pair is sent to the recipient, where he derives the *spendkey* x_j. The value x_j is used by the recipient, for future transactions, to create a proof that he holds the certificate. The value $Cert_j[R]$ is discarded by both parties after x_j is obtained by the recipient since this value can be recomputed at any time.

For each new transaction, an actor creates a proof, indicating knowledge of x_j for each certificate he holds. In addition to the signature, the hash of the certificate $Cert_j[h]$ is published resulting in a tuple $(s_j, e_j, Cert_j[h])$. Algorithm 5.2 shows the procedure to create a certificate proof, using Schnorr signatures.

Algorithm 5.2. Certificate Proof Creation

procedure CREATE_CERTIFICATEPROOF($D, x_j, Cert_j[h]$)
  $k \in_R \mathbb{Z}_p$ and compute $K \leftarrow kG$
  $e_j \leftarrow H(K)$
  $s_j \leftarrow k + x_j e$
  return $(s_j, e_j, Cert_j[h])$
end procedure

HIDING PRIVACY-SENSITIVE INFORMATION In order to hide and safeguard the privacy-sensitive information ($info$), we propose the encryption of $info$. Let r be the receiving parties of the information, where $0 \leq r \leq \mathcal{N}$ and $\mathcal{N}$ the number of actors in the network. The information is encrypted using the Elliptic Curve Integrated Encryption Scheme *(ECIES)*. For the scheme, the first element of the tuple of an actor, $A \in (A, B)$, is taken as the public key. We refer the reader to Algorithm 3.1 for the encryption and decryption procedures for the ECIES-scheme.

5.1.3 *The PASTA Protocol*

We now introduce the PASTA protocol. The protocol is based on the original stealth addresses protocol in [63]. In the original protocol, the recipient can publish a *tracking key*. The tracking key allows the holders of it to link **all** the incoming transactions of the recipient. While this allows a form of auditability, it is not possible to only reveal the incoming transactions of one particular p_{id}. To overcome this shortcoming, and achieve product-specific auditability, we propose the PASTA protocol. The protocol allows for a single tracking key, specific for a p_{id} per actor. Therefore, the tracking key is unique per actor and accompanying p_{id}.

Alice (p_{id})		**Bob** (a, B, b)
	Request TK_{pid} for p_{id} $\longrightarrow$	
		$TK_{pid} \leftarrow \mathcal{H}_s(p_{id}a)G$
	$\longleftarrow$ (B, TK_{pid})	
$r \in_R \mathbb{Z}_p$		
$R \leftarrow rG$		
$P \leftarrow \mathcal{H}_s(rTK_{pid})G + B$		
	(R, P) $\longrightarrow$	
		$P' \leftarrow \mathcal{H}_s(tk_{pid}R)G + B$
		$P' \stackrel{?}{=} P$
		$x \leftarrow \mathcal{H}_s(tk_{pid}R) + b$

Protocol 5.2: PASTA Protocol

In the PASTA protocol, we introduce two types of tracking keys, namely a public and private tracking key. The first is denoted as TK_{pid} and can be requested by anyone in the network, while the latter is denoted as tk_{pid} and is private per actor. Both keys are only possible to be computed by the recipient in the protocol as follows, where a is the recipient's private key.

$$\begin{aligned} TK_{pid} &\leftarrow tk_{pid}G, \text{ where} \\ tk_{pid} &\leftarrow \mathcal{H}_s(p_{id}a) \end{aligned} \tag{5.16}$$

From Equation 5.16 it is clear that p_{id} is incorporated and the private key a of the recipient. This creates a unique combination of p_{id} and recipient. We show the complete PASTA protocol in Protocol 5.2. In the protocol, the sender requests TK_{pid} and the public ec-key B of the recipient. The recipient computes TK_{pid} and returns a tuple containing B and TK_{pid}. The sender then follows the protocol and computes a tuple (R, P), that is then stored in a transaction and broadcasted to the entire network. The recipient then computes P', using the private tracking key tk_{pid}, and the public key R. In case $P' = P$, the transaction was meant for the recipient, and he computes the spendkey x.

During a recall or a disaster, tk_{pid} is shared enabling the holders to reveal the recipient of the transaction, for a specific p_{id}. Since each p_{id} is unique, we assume that an actor will not receive multiple transactions regarding the same p_{id}. Even though if an outsider in the network is in possession of tk_{pid}, the spendkey x cannot be derived since he does not hold x and b. We show the security analysis of the PASTA protocol in Section 6.1.

5.2 TRANSACTION CREATION AND VALIDATION

We denote a transaction as tx_h, where h is the transaction hash, a tuple of the form $\langle p_{id}, mlsags, recinfo, certproofs, info, at, t \rangle$. Table 5.3 contains each field and its corresponding description. We now discuss the creation and validation of each field in tx_h, where we further refer to the fields p_{id}, at and t as *meta*-fields.

Table 5.3: Transaction Structure

FIELD	DESCRIPTION
p_{ID}	Product ID
a_{type}	Actor type
t	Date and time
$mlsags$	List of MLSAGs per received transaction
$recinfo$	List of stealth addresses and Borromean ring signatures
$certproofs$	Proof per certificate owned by the issuing actor
$info$	Additional information for the transaction

META-FIELD The value p_{id} is created by a Producer and is required to be unique. Each actor in the network holds a list of all p_{id}'s PID. First, for each new transaction by a Producer, p_{id} should not be contained in PID. Secondly, at can only be one of the actor types as aforementioned. Thirdly, t must always be in the past, thus $t <$ current time and date.

MLSAG-FIELD The MLSAG ring signature protocol achieves unlinkability by hiding the actor among public information from past transactions. Per received transaction a separate ring signature is created. Each actor holds a list of key-images $\mathcal{I}$ that contains the key images I_j, used in past signatures. $\mathcal{I}$ is used to prevent double-spending attempts in the system. We refer the reader, for the creation and validation of an MLSAG ring signature, to the paragraph on MLSAG ring signatures in Section 3.3.

RECINFO-FIELD The $recinfo$-field contains recipient information, encapsulating stealth addresses and Borromean ring signatures. The first element is constructed using our proposed PASTA protocol in Protocol 5.2. The element cannot be validated due to its unlinkability property. However, the second element can, and is, required to be validated. For the range proof, only the Borromean signature has to be checked. This check is performed for each value in $recinfo$. If all checks return true, the recipient information validation is considered valid. We refer the reader to the paper by Maxwell et al. [46] for the creation and validation algorithm for Borromean ring signatures.

CERTPROOFS-FIELD The $certproofs$-field contains k Schnorr signatures, where k is the number of certificates an actor holds. Recall the proof creation in Section 5.1.2. Each proof $i \in certproofs$ is checked according to Algorithm 5.3. In case the algorithm returns true, the $certproofs$-field is considered valid.

Algorithm 5.3. Certificate Proof Validation

1: **procedure** VALIDATE_CERTIFICATEPROOF($s_j, e_j, Cert_j[h]$)
2: $K = kG = s_iG - e_iCert_j[P]$
3: **if** $e_i \neq H(K)$ **then**
4: return false
5: **end if**
6: $ed \leftarrow Query(Cert_j[h])$
7: **if** $ed <$ current date & time **then**
8: **return** false
9: **end if**
10: **return** true
11: **end procedure**

INFO-FIELD Actors in our system can share confidential information by encrypting this for the corresponding recipient(s). The information is encrypted using ECIES, where the input is the information that needs to be encrypted and a set of public keys of the recipients. To hold the unlinkability property, the $info$-field does not contain any information linking it to a recipient. Therefore, each actor attempts to decrypt the information. If this is successful, the confidential information is retrieved. This field cannot be checked to be valid since it contains encrypted data.

5.3 BLOCK CREATION AND VALIDATION

Recall that transactions are aggregated in blocks, which are then broadcasted to the network before being permanently stored on the blockchain. The block structure we propose is similar to the one described in Bitcoin[2]. Let b be a block and $b[TX]$ be the transaction list in b. First, the syntactic correctness of b is checked and that there exist no duplicate of b. Secondly, $b[TX]$ should not be empty. Thirdly, the Merkle root of b is checked. Finally, for each transaction in $b[TX]$ it is checked according to the checks aforementioned in Section 5.2. We propose a block validation procedure for DECOUPLES as follows.

Algorithm 5.4. Validation of a Block

```
1:  procedure BLOCK_VALIDATION(b)
2:      Check the syntactic correctness of b.
3:      Check that no duplicate of b exists.
4:      Check that b[TX] ≠ ∅; INRF.
5:      Validate Merkle root.
6:      for each tx_i ∈ b do
7:          for each f ∈ tx_i do
8:              Check that f is valid; INRF.          ▷ See Section 5.2
9:          end for
10:     end for
11:     Relay b to nodes.
12:     return true.
13: end procedure
```

5.4 OUR CONTRIBUTIONS

In this chapter, we proposed DECOUPLES, which is a decentralized, unlinkable and privacy-preserving traceability system for supply chains. Our proposed system is the first traceability system for supply chains that takes privacy into account, to the best of our knowledge. Although several works have been proposed, both for a centralized and decentralized system, DECOUPLES is the only system addressing the privacy aspects. Besides, we proposed the PASTA protocol. The protocol anonymizes the receiver and achieves the unlinkability property. Furthermore, the PASTA protocol makes it possible to create product-specific tracking keys. These can be used during a recall, at a disaster or for auditability purposes.

Although the system can satisfy the requirements we have set up, the system is required to be evaluated regarding security, privacy, its performance, and scalability. The evaluation of the system is provided in Chapter 6.

2 Bitcoin block structure: https://en.bitcoin.it/wiki/Block

6 EVALUATION OF DECOUPLES

In the previous chapter, we proposed DECOUPLES and the PASTA protocol. To the best of our knowledge, DECOUPLES is the first system that provides a traceability system for supply chains in a privacy-preserving manner. Several techniques are applied to achieve the goal of this research.

We now evaluate DECOUPLES in four dimensions: security and privacy, performance, storage, and scalability. First, we discuss the security & privacy imposed by the system. Next, we provide a theoretical analysis of the performance based on the computational and communication complexities. Afterwards, the storage complexity of the system is discussed. Finally, we present the measurements obtained from a proof-of-concept implementation to show the practical performance and scalability of the system, taking all required computations into account.

6.1 SECURITY & PRIVACY ANALYSIS

In our proposed system we use MLSAG ring signatures [55], Borromean ring signatures [46] and Schnorr signatures [21], which have been proven securely in their respective papers. Therefore, we refer the reader to their corresponding research papers for their security analyses and proofs. For the certificate transaction, we use stealth addresses. The security relies on obtaining the value x, by the CO. This corresponds to 1 in the security analysis of the PASTA protocol. We now provide the security analysis for our proposed PASTA protocol.

6.1.1 *Security Analysis of the PASTA Protocol*

We provide a security analysis of the PASTA protocol, following the procedures in [55]. We assume a semi-honest security model and the Elliptic Curve Discrete Logarithm Problem (ECDLP). Furthermore, we assume that $\mathcal{H}_s$ is a cryptographically secure hash function. We investigate the anonymity and unlinkability properties of the PASTA protocol.

Recall that the ECDLP is defined as follows [27].

Definition 6.1.1. ECDLP Given two points $P, Q \in E(\mathbb{F}_p)$, where $Q \in \langle P \rangle$, it is computationally infeasible to find a k such that $Q = kP$.

Lemma 1 (Security)**.** The PASTA protocol is secure, if a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ is unable to derive the spendkey x created for the owner of secret key b.

Proof. Assume that a PPT adversary $\mathcal{A}$ obtains a stealth address (R, P) and holds tk_{pid}, provided by the recipient whose identity is known. Note that the adversary does not possess the secret key b that is needed to compute the

spendkey x. The only way for the adversary to successfully compute a valid spendkey, meant for the owner of b, is to obtain the secret key b. This is only possible when A can solve $B = bG$, meaning that he can solve the ECDLP, which is infeasible. □

Lemma 2 (Anonymity)**.** The PASTA protocol provides anonymity, if a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ is unable to derive the receiver of a stealth address.

Proof. Assume that a PPT adversary $\mathcal{A}$ obtains a stealth address (R, P) and p_{id}, he then tries to find the corresponding recipient i. For that, he needs to find a P', such that $P' = P$. Since $\mathcal{A}$ does not know the recipient, he requests the $TK_{i,pid}$ from all the actors in the network for the specific p_{id} and the accompanying B_i.

The adversary holds (R, P) and a list of tuples $(TK_{i,pid}, B_i)$. He needs to compute $P' = \mathcal{H}_s(rTK_{i,pid})G + B_i$, such that $P' = P$. To find such a P', $\mathcal{A}$ computes the difference $P - B_i = \mathcal{H}_s(rTK_{i,pid})$. However, $\mathcal{A}$ does not hold the secret value r. Based on the ECDLP, this is computationally infeasible and thus $\mathcal{A}$ is unable to find the recipient of the stealth address. □

Lemma 3 (Unlinkability)**.** The PASTA protocol provides unlinkability, if a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ is unable to distinguish the receiver of two different stealth addresses.

Proof. Assume that a PPT adversary $\mathcal{A}$ observes numerous stealth addresses and chooses two, namely (R_1, P_1) and (R_1, P_2). He tries to distinguish if the two stealth addresses belong to the same receiver. In this analysis we differentiate two cases:

1) the two stealth address belong to the same recipient or
2) the two stealth address belong to two different recipients.

The adversary computes the difference between P_1 and P_2 to distinguish if they were meant for the same person. In case 1), the difference is computed as:

$$P_1 - P_2 = \mathcal{H}_s(\widetilde{R})G = xG, \text{ for some unknown } x. \tag{6.17}$$

In case 2), where $B_x = \beta G$ and $B_y = \gamma G$, the difference is:

$$\begin{aligned} P_1 - P_2 &= \mathcal{H}_s(\overline{R})G + B_x - \mathcal{H}_s(\overline{\overline{R}})G - B_y, \\ &= (\mathcal{H}_s(\overline{R}) - \mathcal{H}_s(\overline{\overline{R}}))G + B_x - B_y, \\ &= (\mathcal{H}_s(\overline{R}) - \mathcal{H}_s(\overline{\overline{R}}) + \beta - \gamma)G, \\ &= yG, \text{ for some unknown } y. \end{aligned} \tag{6.18}$$

In both cases, $\mathcal{A}$ is unable to distinguish the two equations stated above. He does not know in what form $P_1 - P_2$ is given. Therefore, he is not able to distinguish if the difference is given in the form of case 1) or 2). Furthermore, for $\mathcal{A}$ to be certain that $P_1 - P_2$ is in case 1), he requires to compute x. Based on the ECDLP, this is infeasible to be computed. We thus conclude that, given two different stealth addresses, it is infeasible for $\mathcal{A}$ to distinguish them. □

The overall security and privacy of DECOUPLES relies on the correct appliance of the used protocols. It is of importance that the random numbers generated in the protocols are truly random and not re-used in the same or other protocols. Otherwise, the security of the protocols degrades, since an adversary might use the values to break the properties of the protocols. Although the protocols are proven secure and achieve the required anonymity, providing linkable information eradicates these properties. Since actors are in control of their data and can choose to provide information in an unencrypted form, the responsibility of this is entirely kept to their side.

6.2 PERFORMANCE ANALYSIS

We evaluate the theoretical performance of DECOUPLES in two areas: computational and communication complexity. The first takes the protocols and their cryptographic operations into account, while the latter is focused on the communication of the actors. The performance depends on a number of variables, listed in Table 6.1.

Table 6.1: Symbols used in the performance analysis of DECOUPLES.

SYMBOL	DESCRIPTION
$\mathcal{N}$	Number of actors in the network.
ρ	Number of input transactions.
P	Number of output transactions.
k	Size in bits for range proofs.
c	Number of certificates of an actor.
s	Size in bits of private key for elliptic curve.
$\mathcal{S} = 2s$	Size in bits of point on elliptic curve.
β	Key-size in bits for the symmetric encryption scheme.
e	Size in bits of encrypted data.
ζ	Number of destined actors of encrypted data.
h	Output size in bits of the hash function.
δ	Number of members in a ring signature.

6.2.1 *Computational Complexity*

For the invocation of each protocol, per transaction, the complexity grows linear in its respective variable. Note that the certificate issuing is performed at two different points in time. The first is when an actor joins the network and already holds a set of certificates. The second is when an actor gains other certificates, requiring a new process for the CO and the actor. Furthermore, the complexity for Borromean signatures is bound by the number of bits k for the correspond-

ing range proof. This variable can be reduced, depending on the maximum number of products contained in a transaction. The computational complexity of a transaction is dominated by the Borromean ring signature protocol.

We summarize the computational complexity of each protocol in Table 6.2. The computational complexity of the protocols is based on a single transaction. The Borromean ring signature protocol and the MLSAG protocol require the largest number of operations. The first creates k Pedersen commitments per recipient, whereas the latter depends on δ per input transaction of ρ. However, the computation for both protocols are trivially parallelizable and can thus be sped up significantly using multiple processors or cores on a GPU. Moreover, k can be reduced specific to the scenario. We expect $k = 22$ in a real-world setting for the Borromean ring signature protocol. For the range proof in the Borromean ring signatures protocol, an additional set of public keys is created that is used as the public keys for the creation of the ring signature. The dimension of the set is $2 \times k$, where the computational cost is only k since the first dimension contains a copy of the commitments created before. Therefore, the computational complexity of a transaction is dominated by the Borromean ring signature protocol.

6.2.2 *Communication Complexity*

The required communication bandwidth depends on a number of variables, listed in Table 6.1. We assume that each actor in the network holds a list of addresses of the actors with whom they wish to communicate. Note that the communication costs to broadcast both transactions and blocks is dependent on the consensus model used. The chosen consensus model depends on the number of transactions per second and the number of nodes in the network [77]. For the PoW consensus model the communication complexity is $\mathcal{O}(\mathcal{N})$, whereas for BFT this is $\mathcal{O}(\mathcal{N}^2)$ [77].

For each transaction a number of recipients are set, where for each recipient a separate stealth address is created. The creation of a stealth address, as in Protocol 5.2, contains a request of TK_{pid} for a specific p_{id} where the recipient replies with the TK_{pid}. Therefore, the communication costsper transaction is 2P. Since we assume that a p_{id} is unique and that a product is transferred maximally 5 times, the total computational complexity is $\mathcal{O}(5 \cdot 2PID) = \mathcal{O}(10PID)$, where PID is the list of all unique p_{id}'s.

Moreover, for each validation of a transaction an actor performs a database query to the CO's database for the contained certificate proofs. Here, the communication complexity depends on $\mathcal{N}$ and the number of certificate proofs c, resulting in a complexity of $\mathcal{O}(c\mathcal{N})$. However, this can be improved by caching the queries and reusing them for future validation procedures, eliminating the database queries afterward. In case an actor obtains a new certificate, an additional query is required by all actors to the CO's database.

Table 6.2: Computational complexity analysis for DECOUPLES.

PROTOCOL	OPERATION	SENDER	RECEIVER
Certificate Issuing	Creation	$\mathcal{O}(c)$	
Certificate Proofs	Creation	$\mathcal{O}(c)$	
	Validation		$\mathcal{O}(c)$
MLSAG	Creation	$\mathcal{O}(\rho \cdot \delta)$	
	Validation		$\mathcal{O}(\rho \cdot \delta)$
PASTA	Creation	$\mathcal{O}(\mathrm{P})$	
Borromean Signatures	Creation	$\mathcal{O}(k)$	
	Validation		$\mathcal{O}(k)$
Confidential Data Sharing	Encryption	$\mathcal{O}(\zeta)$	
	Decryption		$\mathcal{O}(\zeta)$

6.3 STORAGE ANALYSIS

We analyze the storage complexity of bits stored locally and transferred over the network, per invocation of the protocols. The amount of bits per invocation depends on a number of variables, listed in Table 6.1. The amortized number of bits stored for each protocol by each party is listed in Table 6.3. The number of bits primarily depend on the chosen security level for the elliptic curve and ECIES. We consider the security level to be constant, and thus the accompanying variables $s, \mathcal{S}$ and β are also constant. The storage complexity of transactions is dominated by Borromean ring signatures, bound by k. The complexity emphasizes the importance of k to be set to a realistic value for supply chains.

One way to overcome the communication complexity in the PASTA protocol is to store all TK_{pid}'s for all p_{id}'s from all $\mathcal{N}$ actors in the network. Let p_{id} be a 32-bit fixed-point integer and TK_{pid} a point on the elliptic curve of $\mathcal{S}$-bits. An actor then requires to hold a table with a p_{id} and the accompanying TK_{pid}, for each actor. The size of the table, in bits, is then $2^{32}\mathcal{N}(32 + \mathcal{S})$. The total size of the tables, where $\mathcal{S} = 512$ for a 128 bit level of security, is approximately $272\mathcal{N}$ Gibibytes, where the size grows linear in $\mathcal{N}$. Due to the significant size of the tables that are required to be kept, it is impractical to store these tables.

To denote the storage in bits for a transaction in a real-world setting we set the required values as follows. We set the security level to 128 bits for the elliptic curve, $s = 256, \mathcal{S} = 512$. AES256 is used as a symmetric encryption scheme, where $\beta = 256$ bits. The block size, IV and h are 256 bits, where SHA3-256 is used as the hash function. Next, we set $k = 22$, $m = 2$ and $\delta = 4$ for Borromean ring signatures and the MLSAG protocol. Moreover, we set $\rho = 1, \mathrm{P} = 8$ and $\zeta = 1$ with the plaintext size as 8192 bits. With these settings, the storage of a transaction is approximately 8.26 Kibibytes.

Table 6.3: Storage complexity analysis in DECOUPLES.

PROTOCOL	ACTOR
Certificate Proofs	$\mathcal{O}(3s)$
MLSAG	$\mathcal{O}(m(\mathcal{S}+\mathcal{S}\delta+s\delta)+h)$
PASTA	$\mathcal{O}(4\mathcal{S})$
Borromean Ring Signatures	$\mathcal{O}(4k\mathcal{S}+3s)$
Confidential Data Sharing	$\mathcal{O}(e+\zeta\beta+h)$

6.4 EXPERIMENTAL RESULTS ANALYSIS

To measure the runtime of DECOUPLES, we created a proof-of-concept implementation of the system in Python 2.7 using a simple blockchain implementation based on the work of Daniel van Flymen[1]. Besides, we used the elliptic curve implementation on the $secp256k1$ curve by Adam Gibson as well as a modified version of his Borromean ring signatures implementation[2]. All cryptographic operations use a key length of 256 bits to achieve a sufficient security level, where the NIST considers a key length of 256 sufficiently secure until 2030[3]. The p_{ID} values are represented as 32-bit fixed-point numbers. The measures of the runtime were executed on an Amazon® EC2 server, running Ubuntu 16.04 on a 3.0GHz Intel® Xeon Platinum processor with 16 vCPUs and 32GB RAM. For accurate measurements, we executed 1000 iterations of the procedures for each protocol.

Figure 6.1 shows the impact of the number of input and output transactions for the construction of a transaction and the validation of it. Both procedures increase linearly as the amount of input and output transactions increases. However, the increase in runtime for the validation is slower than for the creation. We now break the construction and validation of a transaction up in the protocols it encompasses.

For the MLSAG protocol, we set $\delta = 4$. The runtime of the MLSAG protocol, regarding the creation and validation, are equal due to equivalent internal cryptographic computations. The runtime is approximately 0.17ρ seconds, for ρ input transactions. Although the runtime of the MLSAG protocol can be decreased by applying parallelization, it cannot be fully parallelized since aspects of the algorithm depend on previous computations.

The performance measurements of the Borromean ring signature protocol is done in four procedures: the creation of the range proof ($Rangeproof$), signing a range proof ($Sign$), the combination of the previous two procedures ($BorSig$) and the verification of a Borromean ring signature ($Verify$). It is clear that the runtime increases linearly in k, as shown in Figure 6.2. The difference in runtime

1 A simple Blockchain Implementation, https://github.com/dvf/blockchain
2 Borromean ring signatures Code, https://github.com/AdamISZ/borring
3 See https://www.keylength.com/ for key lengths recommended by various organizations

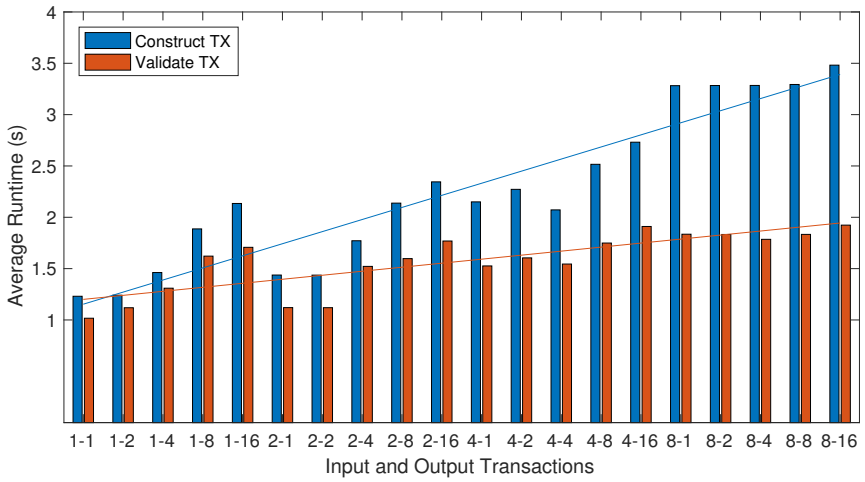


Figure 6.1: Total computation time required for the creation and validation of a single transaction, based on the amount of input and output transactions.

between $k = 22$ and $k = 32$ shows an increase of approximately 50%. The significant increase indicates that it is important to choose a k that is sufficient for supply chains to overcome this decrease in performance.

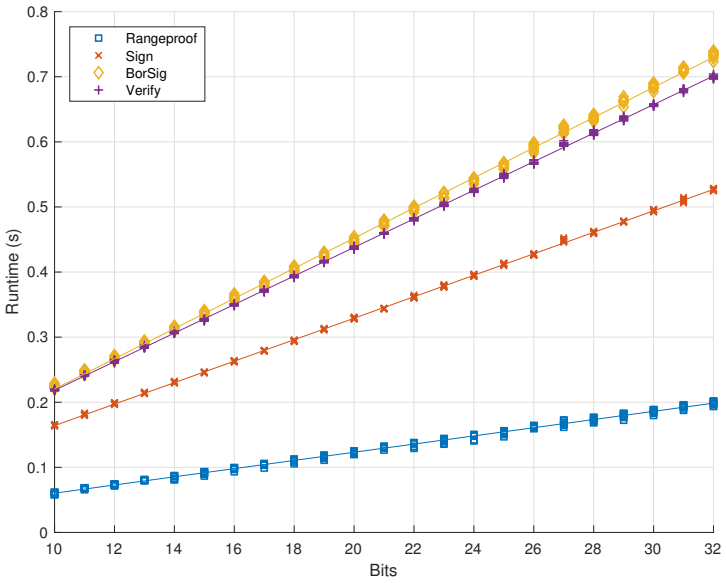


Figure 6.2: The effect of k on the Borromean ring signature protocol.

We denote three procedures for our PASTA protocol: (i) correct recipient, (ii) get spendkey and (iii) generate stealth address (SA). Besides the additional communication in the PASTA protocol, the protocol computations have not changed. In our computations, we do not take the creation of TK_{pid} into account. Therefore, the runtime for PASTA is equal to the original stealth addresses protocol in [63]. Figure 6.3 shows the runtime of the PASTA protocol according to the three procedures.

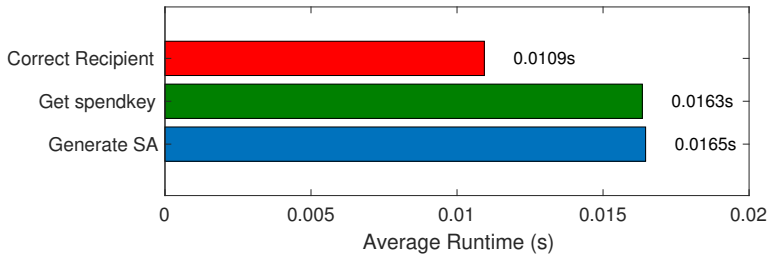


Figure 6.3: Average runtime of the PASTA protocol.

For the creation and verification, of certificate proofs, the procedures show a somewhat constant runtime. The creation takes approximately $0.5 \cdot 10^{-2}$ seconds, whereas the verification takes approximately $1.0 \cdot 10^{-2}$ seconds. It is clear that the verification takes twice the amount of time.

The obtained measurements discussed in this section can be improved in three areas. First, the appliance of another programming language that is optimized for the computation of mathematical operations such as `C` or `C++`[4]. Secondly, the operations on elliptic curves can be implemented on hardware to gain a speedup [16]. Thirdly, the used protocols can be parallelized to obtain a speedup.

6.5 SCALABILITY ANALYSIS

We use the measurements, obtained in the previous section, for our scalability analysis. For a realistic real-world setting, we assume $\mathcal{N} = 1000$ and that an actor constructs a transaction with one input transaction and a maximum of eight output transactions. The scalability depends on the creation and validation of transactions. In Section 6.2 it is shown that all protocols grow linearly in their respective variable.

Based on Figure 6.1, an actor is able to construct $\gamma = \frac{60}{1.9} \approx 31$ transactions per minute. Parallelization can increase this number concerning ℓ, where ℓ is the number of transactions that can be created in parallel. Therefore, the system is limited to 31ℓ transactions per minute (tx/min) per actor. However, in the case of an actor, specialized in a specific product, a single transaction can be created encompassing all products with an equal p_{ID}. Besides, due to the transportation requirements of products in the supply chain, which takes significantly longer than one minute, a chance of network congestion is unlikely to occur.

An actor in the system is able to validate $\frac{60}{1.6} \approx 38$ transactions per minutes, without the appliance of parallelization. Given $\mathcal{N} = 1000$, in the worst-case scenario, an actor needs to validate $\Gamma = 3.8 \cdot 10^4$ transactions per minute. However, an actor is only able to validate 38 transactions per minute. Therefore, it is infeasible to validate the same amount of transactions as there would be created transactions per minute. However, since products are transported physically, the network does not meet Γ. Furthermore, actors create transactions per product, for multiple recipients. This results in a significantly lower amount of separate

4 See, e.g., https://benchmarksgame.alioth.debian.org/ for benchmark results between Python and C/C++

transactions that have to be created. Therefore, the validation of transactions is not significantly slower than the possible amount of transactions created per minute.

For an actor to construct a new transaction, all transactions destined to him are collected. The procedure to check if a transaction is destined for him takes $\sim 1.1 \cdot 10^{-2}$s. Without the appliance of parallelization, a is able to check $\frac{1}{1.1 \cdot 10^{-2}} \approx 92$ tx/s and thus approximately $5.5 \cdot 10^3$ tx/min. Given the limitation of 31 tx/m per actor and the assumption that $\mathcal{N} = 1000$, $3.1 \cdot 10^4$ transactions are constructed per minute for all actors on the blockchain. Although this is slower than the number of transactions constructed per minute, the procedure can be easily parallelized to overcome this. On an equal system as used for the measurements in Section 6.4, with 16 CPU cores, we can increase the performance of the procedure from $5.5 \cdot 10^3$ to approximately $8.8 \cdot 10^4$. We have now achieved a performance of $8.8 \cdot 10^4$ which is higher than the number of transactions constructed per minute, overcoming the previously stated limitation.

6.6 DISCUSSION

DECOUPLES is, to the best of our knowledge, the first decentralized, unlinkable, privacy-preserving traceability system for supply chains. Trust is distributed among the actors in the network by using blockchain technology. The system takes privacy-sensitive information, certificate verifiability, and auditability into account. The result is a traceability system for supply chains, without any transactions linkable to the responsible actors. Our proposed system satisfies all the requirements set up in Chapter 5. Also, we introduced a novel protocol, namely PASTA, allowing product-auditability. The protocol allows actors to reveal their p_{ID}-specific tracking keys. These keys are used to reveal the recipient of a transaction, rather than all transactions as in the original stealth addresses protocol. Furthermore, non-interactive zero-knowledge proofs are used to prove that an actor holds a certificate, without revealing their identity. The proofs can be verified by any entity, such as consumers, achieving certificate verifiability.

Based on our theoretical and experimental analysis, we note that the performance is limited by the creation of transactions rather than the validation. The cause is mainly due to the production of range proofs and accompanying Borromean ring signatures. Although DECOUPLES only allows 31 transactions per minute, per actor, the computational performance shows promising results. Most computations in the system can be parallelized across the actors, and the operations that cannot be parallelized are at most linear in their corresponding variables. Besides, the variable k can be reduced depending on the supply chain setting. Our performance analysis and the proof-of-concept implementation results show that, although the system has the aforementioned limitations, it remains a practical system for supply chains.

7

DISCUSSION AND FUTURE WORK

The traceability aspect of supply chains has experienced several highlights throughout the last few decades, such as the bovine spongiform encephalopathy disease and the avian influenza [12, 24, 66]. These type of incidents increased the importance of traceability, both on a business level and on a regulation level. Moreover, traceability is being used to improve the performance of the business as well as the compliance with (inter)national regulations. Besides these supply chain actors, other parties such as consumers, Non-Governmental Organizations (NGOs), governments, suppliers, and buyers show an increase in demand for information regarding their products and materials [56].

Several approaches have been suggested in previous works to achieve a traceability system for supply chains. These systems either employ a centralized [8, 36, 38, 75, 82] or decentralized [1, 39, 73] network. However, the previous works do not state the feasibility of their work and the appliance for supply chains. Therefore, we proposed TRADE to achieve a fully transparent, decentralized traceability system. The system has been provided with accompanying analysis and results from a proof-of-concept implementation. Nevertheless, TRADE and the previous works take the general assumption that actors are willing to share their data openly. This assumption is undesired in the current supply chain due to the presence of privacy-sensitive information. Furthermore, the previous works failed to address the certificate verifiability and product-specific auditability, that are desired in a traceability system. For our second system, we discard the aforementioned assumption and propose a complete traceability system, namely DECOUPLES, and a product-auditable protocol called PASTA. In this chapter, we revisit our main research question:

> *How can we achieve traceability for supply chains in a privacy-preserving manner, where parties can verify certificates, and product-specific auditability is achieved in a single system by employing blockchain technology?*

In this chapter, we discuss how the research goal is achieved, by examining our proposed systems TRADE and DECOUPLES. Moreover, we provide future research possibilities by identifying the remaining open problems and improvements for our system.

7.1 DISCUSSION

The studies in [8, 36, 38, 75, 82] show methods to achieve a traceability system for supply chains in a centralized manner, whereas [1, 39, 73] show methods for a decentralized network. To provide decentralization, the latter works use blockchain technology. Kim et al. [39] and Tian [73] apply blockchain technology for traceability solutions, whereas Abeyratne et al. [1] provide a broader

view on the traceability aspect in terms of transparency and sustainability. Nevertheless, all previous works fail to address the presence of privacy-sensitive information. In the work by Abeyratne et al., the appliance of product certificates is discussed. However, their solution reveals privacy-sensitive information and does not account for certificate revocation. Moreover, the previous works do not take auditability of products into account. During a recall, a disaster or for auditability purposes, the data accompanying a specific product is not directly accessible.

In this research, two complete traceability systems for supply chains are presented. Both systems are built on top of a public permissioned blockchain. In the first system, called TRADE, we focus on the trust concern presented in supply chains and envision to address the first sub-question posed in Section 1.5. The second system, called DECOUPLES, discards this assumption that actors are willing to share information openly in a single system. DECOUPLES addresses the privacy-sensitive information, certificate verifiability, and auditability concern. Both systems have been implemented to show the performance and the feasibility for a real-world setting.

TRADE The first system, namely TRADE, is a fully transparent traceability system. The recent studies [39, 73] proposed transparent, decentralized traceability system. Their work is purely theoretical, and the feasibility is not discussed. In TRADE, each actor creates a transaction regarding a product p_{id} containing all information on the product. The stored data inside a transaction is fully transparent allowing each actor in the network to view the data. Each transaction is signed by the issuing actor using a digital signature, providing a proof of authenticity, integrity, and non-repudiation. The insight on the data can be used to improve planning and scheduling, and facilitates faster recalls. Also, consumers obtain access to this data and gain insight into the full life-cycle of products. Standardization is enforced in TRADE since each transaction, depending on the issuing actor, has a corresponding validation procedure.

The TRADE system has been implemented and shows promising results. Based on our implementation and the hardware used, 351 transactions per second can be created, whereas 471 transactions can be validated per second. An actor is thus able to validate $437/\ell$ transactions per second. TRADE achieves a significant speed for creating and validating transactions, as well as the validation of blocks. We show that it is feasible to apply blockchain technology for supply chains to achieve traceability. Moreover, consumers and other parties can view the data to gain knowledge on the procedures performed on their product, as well as information on the sustainability, if the actors provide it. Actors are in control to share such information, which is recommended since it improves the company's brand image and increases the trust of consumers in the company. In case actors are willing to share data in a single system and achieve full traceability, blockchain technology is shown to be a feasible solution to accomplish this in a real-world setting for supply chains.

Digital signatures are the only cryptographic primitive used in TRADE. The digital signatures are used to validate the authenticity of transactions and to check the accompanying integrity. Therefore, the security of the system relies on

the security provided by ECDSA, the used digital signature scheme. The security of ECDSA relies on the elliptic curve discrete logarithm problem (ECDLP), which is considered to be computationally hard [34].

DECOUPLES Our second proposed system, DECOUPLES, discards the assumption that actors are willing to share data openly. To the best of our knowledge, DECOUPLES is the first decentralized, privacy-preserving traceability system for supply chains. The system takes privacy-sensitive information, certificate verifiability, and auditability into account. The result is a feasible system for supply chains, without any linkable transactions to the responsible actors. Also, we introduced a novel protocol that allows for product-auditable tracking keys, named PASTA. In case of a recall, a single tracking key concerning a product p_{id} can be distributed allowing auditability. Our proposed system satisfies the requirements that we have set up, namely

1. untraceability,
2. unlinkability,
3. hidden relationships,
4. anonymous certificates to hide the owner and
5. hide privacy-sensitive information.

In DECOUPLES, cryptographic protocols are used to anonymize both the sender and receiver for each transaction. Since the parties are anonymized, the relationship between these actors is also hidden. The untraceability and unlinkability properties are provided by MLSAG ring signatures and our PASTA protocol. Furthermore, transaction amounts are hidden since throughput is considered as privacy-sensitive information. To achieve certificate verifiability, we incorporated certificate organizations. Non-interactive zero-knowledge proofs are used to hide the owner of a certificate, whereas the proofs are verifiable by everyone. DECOUPLES uses our proposed PASTA protocol for auditability purposes. The protocol anonymizes the recipient of a transaction, whereas a tracking key enables the recipient of a specific product in case of a recall, disaster or audit, to be revealed.

Based on our theoretical and experimental analysis, we note that the performance is limited by the creation of transactions rather than the validation. Our implementation of DECOUPLES shows promising results, where 31 transactions per minute can be created, 38 transactions can be validated each minute. For an actor to construct a new transaction, all transactions destined to him are collected. From our experimental results, approximately $4.4 \cdot 10^4$ transactions can be checked per second, using parallelization. Nevertheless, due to the transportation requirements of products in supply chains, which takes significantly longer than one minute, a chance of network congestion is unlikely to occur. The experimental results indicate that DECOUPLES is feasible to be applied in a real-world setting.

For DECOUPLES, a number of cryptographic protocols are used: MLSAG ring signatures, the PASTA protocol, Borromean ring signatures, Schnorr signatures, and ECIES. All the used protocols, except for the PASTA protocol, have been proven to be secure in their respective works. In this research, we provided a se-

curity analysis of the PASTA protocol concerning its properties: anonymity and unlinkability. However, the overall security and privacy implications of DECOUPLES rely on the correct appliance of the aforementioned protocols. In particular, since actors are in control of their data, the additional information stored in a transaction should not leak any privacy-sensitive or linkable information.

7.2 FUTURE WORK

The presented DECOUPLES system is, to the best of our knowledge, the first system that takes privacy, confidential data and, verifiability of certificates into account. Furthermore, it eliminates the assumption that actors are willing to share their data openly. The system shows promising results regarding privacy and performance. However, there is spacious opportunity to improve the system and the protocols it utilizes.

PERFORMANCE OF BORROMEAN RING SIGNATURES Although the performance achieved by DECOUPLES is promising and indicates a feasible solution as a traceability system for supply chains, the most significant bottleneck for the system is the Borromean ring signatures protocol. The protocol grows linearly in k, which is the size in bits for the range proofs. Since the transaction amount is required to be proven in a range, and it is represented in a binary form of k-bits, k Pedersen commitments are required to be computed. Also, a set of public keys is computed as input to create a Borromean ring signature, which is of size $2k$.

A possible direction for improvement is the appliance of Bulletproofs [10] instead of Borromean ring signatures. Bulletproofs are shown to be more storage-efficient. Bulletproofs contain shorter proofs, which are only of a logarithmic size in the number of multiplication gates used. The work by Bünz et al. discusses that a cryptographic proof can be reduced from over 10kB to less than 1kB. However, the computational complexity of Bulletproofs is significantly higher than Borromean ring signatures. Therefore, Bulletproofs have not been applied in this research.

PERFORMANCE MLSAG In DECOUPLES, for the proof-of-concept implementation, we achieved an average runtime of approximately 0.17s. However, since the computational complexity is linear in the number of input transactions, this grows significantly for actors in supply chains that receive products from more than one source. A future research direction is therefore to investigate the performance of the MLSAG ring signatures protocol and to improve the speed for both the signing and verification procedure. A possibility is to introduce parallelization in the protocol at several levels, yet this won't achieve the necessary speedup due to the dependency of previous results within the protocol.

SIDE CHAINS As aforementioned, DECOUPLES enables the ability to share confidential data with other actors. However, the size of the public keys for the destined actors and the size of the encrypted data can become a problem regarding storage. This problem might be overcome using sidechains and is

left for future research [4]. Encrypted data can be stored on a sidechain, while a reference to the sidechain can be inserted into the main blockchain. Using sidechains, the storage complexity can be reduced. In case a destined actor requires the encrypted data, it can be requested from the sidechain. However, sidechains introduce additional complexity on a network and asset level. The drawbacks of sidechains are required to be addressed first [4].

7.3 CONCLUDING REMARKS

The prior art in traceability studies failed to address the presence of privacy-sensitive information, certificate verifiability, and auditability. The objective of this research has been to address these concerns. The two systems presented in this research achieve the first research objective to decentralize the system and distribute thrust, previously placed in a single party, among the actors in the network. The first presented system, TRADE, took the same assumption as the previous works and achieved a fully transparent, decentralized system. The system did not take the concerns into account that are required to be addressed to achieve a feasible traceability system for supply chains. However, the second system, DECOUPLES, utilized anonymization techniques and applied cryptographic primitives that enables actors to prove they hold a certificate in a privacy-preserving manner. Moreover, the system enables the presence of confidential data through encryption. DECOUPLES and its feasibility suggest that a decentralized, confidential and unlinkable system is realizable for supply chains.

The privacy and performance achieved by DECOUPLES provides the first step towards adoption of privacy-preserving methods for traceability systems in supply chains. We have shown that the system provides properties that are crucial for supply chains and the need for traceability. The actors can achieve their desired traceability while preserving their privacy. In the current state, actors in supply chains hold and maintain individual systems and require substantial effort to achieve traceability throughout the entire supply chain. DECOUPLES shows that this is unnecessary by creating a single, decentralized system to satisfy the needs of the actors as well as consumers who desire visibility and verifiability of the products they acquire. The results of this research show that traceability within supply chains benefits both the included business as well as the end-client.

BIBLIOGRAPHY

[1] Saveen A. Abeyratne and Radmehr P. Monfared. "Blockchain Ready Manufacturing Supply Chain Using Distributed Ledger." In: *International Journal of Research in Engineering and Technology* 05.09 (2016), pp. 1–6. ISSN: 23217308. DOI: 10.15623/ijret.2016.0509001. URL: http://esatjournals.net/ijret/2016v05/i09/IJRET20160509001.pdf.

[2] *Adidas Global Factory List*. 2017. URL: https://www.adidas-group.com/en/sustainability/compliance/supply-chain-structure/.

[3] J Michael Martinez de Andino. *Counterfeits in the Supply Chain: A Big Problem and it's Getting Worse*. 2014. URL: http://www.industryweek.com/inventory-management/counterfeits-supply-chain-big-problem-and-its-getting-worse.

[4] Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille. "Enabling blockchain innovations with pegged sidechains." In: (2014). URL: https://blockstream.com/sidechains.pdf.

[5] Ann Baier. "Organic Certification Process." In: *Review Literature And Arts Of The Americas*. 2005, p. 8.

[6] Arati Baliga. *Understanding blockchain consensus models*. Tech. rep. Persistent Systems Ltd., 2017.

[7] Benita M Beamon. "Supply chain design and analysis:: Models and methods." In: *International journal of production economics* 55.3 (1998), pp. 281–294.

[8] Alessio Bechini, Mario G C A Cimino, Francesco Marcelloni, and Andrea Tomasi. "Patterns and technologies for enabling supply chain traceability through collaborative e-business." In: *Information and Software Technology* 50.4 (2008), pp. 342–359.

[9] Anders Björk, Martin Erlandsson, Janne Häkli, Kaarle Jaakkola, Åsa Nilsson, Kaj Nummila, Ville Puntanen, and Antti Sirkka. "Monitoring environmental performance of the forestry supply chain using RFID." In: *Computers in industry* 62.8 (2011), pp. 830–841.

[10] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. *Bulletproofs: Efficient Range Proofs for Confidential Transactions*. Tech. rep. IACR Cryptology ePrint Archive, 2017: 1066, 2017. URL: http://web.stanford.edu/~buenz/pubs/bulletproofs.pdf.

[11] Miguel Castro, Miguel Castro, Barbara Liskov, and Barbara Liskov. "Practical Byzantine fault tolerance." In: *OSDI {'}99: Proceedings of the third symposium on Operating systems design and implementation* February (1999), pp. 173–186. ISSN: 07342071. DOI: 10.1.1.17.7523. arXiv: arXiv:1203.6049v1.

[12] W Chansud, J Wisanmongkol, and U Ketprom. "RFID for poultry traceability system at animal checkpoint." In: *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-CON 2008. 5th International Conference on*. Vol. 2. IEEE. 2008, pp. 753–756.

[13] Tim Connor. *Still waiting for Nike to do it: Nike's labor practices in the three years since CEO Phil Knight's speech to the National Press Club*. Global Exchange, May 2001. ISBN: 0971144303.

[14] Sebastiaan Deetman. *Bitcoin Could Consume as Much Electricity as Denmark by 2020*. 2016. URL: https://motherboard.vice.com/en_us/article/aek3za/bitcoin-could-consume-as-much-electricity-as-denmark-by-2020.

[15] Hans Delfs, Helmut Knebl, and Helmut Knebl. *Introduction to Cryptography: Principles and Applications*. Vol. 3. Springer, 2015. ISBN: 3662479745.

[16] Guerric Meurice de Dormale and Jean-Jacques Quisquater. "High-speed hardware implementations of elliptic curve cryptography: A survey." In: *Journal of systems architecture* 53.2 (2007), pp. 72–84.

[17] A Louis Dorny and Gordon & Rees LLP. *Protecting Confidential Information in the Supply Chain Checklist*. 2014. URL: https://www.gordonrees.com/Templates/media/files/pdf/Protecting%20Confidential%20Information_%20Supply%20Chain%20Checklist.pdf.

[18] Evan Duffield and Daniel Diaz. *Whitepaper - DASH*. 2015. URL: https://github.com/dashpay/dash/wiki/Whitepaper.

[19] Morris J. Dworkin. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. Tech. rep. August. 2015. DOI: 10.6028/NIST.FIPS.202. URL: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf.

[20] Sara D Elder, Hisham Zerriffi, and Philippe Le Billon. "Is Fairtrade certification greening agricultural practices? An analysis of Fairtrade environmental standards in Rwanda." In: *Journal of Rural Studies* 32 (2013), pp. 264–274.

[21] Amos Fiat and Adi Shamir. "How to prove yourself: Practical solutions to identification and signature problems." In: *Conference on the Theory and Application of Cryptographic Techniques*. Springer. 1986, pp. 186–194.

[22] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. "Quadratic span programs and succinct NIZKs without PCPs." In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2013, pp. 626–645.

[23] Larry C Giunipero and Richard R Brand. "Purchasing's role in supply chain management." In: *The International Journal of Logistics Management* 7.1 (1996), pp. 29–38.

[24] Elise Golan, Barry Krissoff, and Fred Kuchler. "Food traceability." In: *Amber Waves* (2004), p. 14.

[25] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The knowledge complexity of interactive proof systems." In: *SIAM Journal on computing* 18.1 (1989), pp. 186–208.

[26] Klaus G Grunert, Sophie Hieke, and Josephine Wills. "Sustainability labels on food products: Consumer motivation, understanding and use." In: *Food Policy* 44 (2014), pp. 177–189.

[27] *Guide to Elliptic Curve Cryptography*. Vol. 5659. Springer, 2004, p. 311. ISBN: 038795273X. DOI: 10.1007/b97644.

[28] Don Gunasekera. *Data transparency and confidentiality in food supply chain*. Keynote during the 2015 APEC Meeting on Expert Consultation on Methodology of Fishery and Livestock Losses, Taipei. July 2015. URL: http://apec-flows.ntu.edu.tw/upload/edit/file/2%20SR_2015_C_S4-01_Dr.%20Don%20Gunasekera.pdf.

[29] Garrett Hardin. "The tragedy of the commons." In: *Journal of Natural Resources Policy Research* 1.3 (2009), pp. 243–253. ISSN: 19390467. DOI: 10.1080/19390450903037302. arXiv: arXiv:1011.1669v3.

[30] *ISO 9000 2015 Definitions*. 2014. URL: http://www.praxiom.com/iso-definition.htm#Traceability.

[31] Thomas Icart. "How to Hash into Elliptic Curves." In: *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*. 2009, pp. 303–316. DOI: 10.1007/978-3-642-03356-8_18. URL: https://doi.org/10.1007/978-3-642-03356-8_18.

[32] Monique H Jansen-Vullers, Christian A van Dorp, and Adrie J M Beulens. "Managing traceability information in manufacture." In: *International journal of information management* 23.5 (2003), pp. 395–413.

[33] In collaboration with Jeff Garzik. *Public versus Private Blockchains - Part 1: Permissioned Blockchains*. Tech. rep. BitFury Group, Oct. 2015.

[34] Don Johnson, Alfred Menezes, and Scott Vanstone. "The Elliptic Curve Digital Signature Algorithm (ECDSA)." In: *International Journal of Information Security* 1.1 (2001), pp. 36–63. ISSN: 1615-5262.

[35] Janjoost Jullens, Mattijs Taanman, and Ilhan Ünlü. *Blockchain X Energy, A Natural Match*. Tech. rep. Blocklab, Sept. 2017.

[36] Yong-Shin Kang and Yong-Han Lee. "Development of generic RFID traceability services." In: *Computers in industry* 64.5 (2013), pp. 609–623.

[37] Vivek Katiyar, Kamlesh Dutta, and Syona Gupta. "A survey on elliptic curve cryptography for pervasive computing environment." In: *International Journal of Computer Applications* 11.10 (2010), pp. 41–46.

[38] Thomas Kelepouris, Katerina Pramatari, and Georgios Doukidis. "RFID-enabled traceability in the food supply chain." In: *Industrial Management & data systems* 107.2 (2007), pp. 183–200.

[39] Henry M. Kim and Marek Laskowski. "Towards an Ontology-Driven Blockchain Design for Supply Chain Provenance." In: *CoRR* abs/1610.02922 (2016). arXiv: 1610.02922. URL: http://arxiv.org/abs/1610.02922.

[40] Neal Koblitz and Alfred J. Menezes. "Cryptocash, cryptocurrencies, and cryptocontracts." In: *Des. Codes Cryptography* 78.1 (2016), pp. 87–102. DOI: 10.1007/s10623-015-0148-5. URL: https://doi.org/10.1007/s10623-015-0148-5.

[41] Nir Kshetri. "Can Blockchain Strengthen the Internet of Things?" In: *IT Professional* 19.4 (2017), pp. 68–72. DOI: 10.1109/MITP.2017.3051335. URL: https://doi.org/10.1109/MITP.2017.3051335.

[42] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. "Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups (Extended Abstract)." In: *Information Security and Privacy: 9th Australasian Conference, ACISP 2004, Sydney, Australia, July 13-15, 2004. Proceedings*. 2004, pp. 325–335. DOI: 10.1007/978-3-540-27800-9_28. URL: https://doi.org/10.1007/978-3-540-27800-9_28.

[43] Christopher Malmo. *Bitcoin Is Unsustainable*. 2015. URL: https://motherboard.vice.com/en_us/article/ae3p7e/bitcoin-is-unsustainable.

[44] Juri Mattila. "The blockchain phenomenon." In: *Reuters* (2013), pp. 1–7. ISSN: 0035-9149. URL: http://blogs.reuters.com/felix-salmon/2013/04/09/the-disruptive-potential-of-native-advertising/.

[45] Greg Maxwell. "CoinJoin: Bitcoin privacy for the real world." In: *Post on Bitcoin Forum*. 2013.

[46] Gregory Maxwell and Andrew Poelstra. "Borromean ring signatures." 2015. URL: http://diyhpl.us/~bryan/papers2/bitcoin/Borromean%20ring%20signatures.pdf.

[47] Alan McKinnon, Michael Browne, Anthony Whiteing, and Maja Piecyk. *Green logistics: Improving the environmental sustainability of logistics*. Kogan Page Publishers, 2015.

[48] Ralph C. Merkle. "A Digital Signature Based on a Conventional Encryption Function." In: *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*. Springer. 1987, pp. 369–378. DOI: 10.1007/3-540-48184-2_32. URL: https://doi.org/10.1007/3-540-48184-2_32.

[49] Victor S. Miller. "Use of Elliptic Curves in Cryptography." In: *Advances in Cryptology - CRYPTO '85, Santa Barbara, California, USA, August 18-22, 1985, Proceedings*. Springer. 1985, pp. 417–426. DOI: 10.1007/3-540-39799-X_31. URL: https://doi.org/10.1007/3-540-39799-X_31.

[50] Robert M Monczka, Robert B Handfield, Larry C Giunipero, and James L Patterson. *Purchasing and supply chain management*. Cengage Learning EMEA, 2015. ISBN: 140801744X.

[51] Alan Morrison. *Private blockchains, public, or both?* 2016. URL: http://usblogs.pwc.com/emerging-technology/private-blockchains-public-or-both/.

[52] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System." In: *www.Bitcoin.org* (2008), p. 9. ISSN: 09254560. DOI: 10.1007/s10838-008-9062-0. arXiv: 43543534534v343453. URL: https://bitcoin.org/bitcoin.pdf.

[53] Katsuyuki Nakano and Masahiko Hirao. "Collaborative activity with business partners for improvement of product environmental performance using LCA." In: *Journal of Cleaner Production* 19.11 (2011), pp. 1189–1197.

[54] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.

[55] Shen Noether. "Ring Signature Confidential Transactions for Monero." In: *IACR Cryptology ePrint Archive* 2015 (2015), p. 34. URL: http://eprint.iacr.org/2015/1098.

[56] Tara Norton, Julia Beier, Lauren Shields, Anita Househam, Elena Bombis, and Daniella Liew. "A Guide to Traceability: A Practical Approach to Advance Sustainability in Global Supply Chains." In: *Bsr* (2014), p. 47.

[57] Hannah Parry. *Beware the Fairtrade fraudsters: Shoppers warned to watch out for produce with fake labels as criminals attempt to cash in on premiums on 'ethical' goods*. 2015. URL: http://www.dailymail.co.uk/news/article-3069609/Shoppers-warned-watch-produce-fake-Fairtrade-labels-criminals-attempt-cash-premiums-ethical-goods.html.

[58] NIST FIPS Pub. "197: Advanced encryption standard (AES)." In: *Federal information processing standards publication* 197.441 (2001), p. 311.

[59] Minghua Qu. "SEC 2: Recommended Elliptic Curve Domain Parameters." In: (2000).

[60] *Quality management systems – Requirements*. Standard. Geneva, CH: International Organization for Standardization, 2015.

[61] Fergal Reid and Martin Harrigan. "An analysis of anonymity in the bitcoin system." In: *Security and privacy in social networks*. Springer, 2013, pp. 197–223.

[62] John Ryu. *GS1 Standards Document GS1 Global Traceability Standard*. Standard. Global Standards One, 2012.

[63] Nicolas Van Saberhagen and Nicolas van Saberhagen. "Cryptonote v2.0." In: *Self-published* (2013), pp. 1–20. URL: https://cryptonote.org/whitepaper.pdf.

[64] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. "Zerocash: Decentralized anonymous payments from bitcoin." In: *Security and Privacy (SP), 2014 IEEE Symposium on*. IEEE. 2014, pp. 459–474.

[65] *Serial Shipping Container Code (SSCC)*. 2017. URL: https://www.gs1.org/serial-shipping-container-code-sscc.

[66] Sununtar Setboonsarng, Jun Sakai, and Lucia Vancura. *Food safety and ICT traceability systems: Lessons from Japan for developing countries*. Tech. rep. ADBI working paper series, 2009.

[67] Victor Shoup. "A Proposal for an ISO Standard for Public Key Encryption." In: *IACR Cryptology ePrint Archive* 2001 (2001), p. 112. URL: http://eprint.iacr.org/2001/112.

[68] Melanie Swan. *Blockchain: Blueprint for a new economy*. "O'Reilly Media, Inc.", 2015, p. 152. ISBN: 1491920475.

[69] Paweł Szewczyk. "The Potential Impact of the Blockchain Technology on the Financial Sector." In: *Finance Today and Tomorrow: Opportunities, Threats, and Challenges* (2016), p. 63.

[70] Michael Szydlo. "Merkle Tree Traversal in Log Space and Time." In: *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*. 2004, pp. 541–554. DOI: 10.1007/978-3-540-24676-3_32. URL: https://doi.org/10.1007/978-3-540-24676-3_32.

[71] Michigan State University. Global Logistics Research Team and Council of Logistics Management (US). *World class logistics: the challenge of managing continuous change*. Council of Logistics Management, 1995.

[72] *The GS1 Traceability Standard: What you need to know*. Standard. Global Standards One, 2007.

[73] Feng Tian. "An agri-food supply chain traceability system for China based on RFID & blockchain technology." In: *Service Systems and Service Management (ICSSSM), 2016 13th International Conference on*. IEEE. 2016, pp. 1–6.

[74] *UTZ presents Impact Report 2014*. 2014. URL: https://utzcertified.org/nl/newsroom/utz-in-the-news/26582896-utz-presents-impact-report-2014.

[75] Jack G A J Van Der Vorst, Seth-Oscar Tromp, and Durk-Jouke van der Zee. "Simulation modelling for food supply chain redesign; integrated decision making on product quality, sustainability and logistics." In: *International Journal of Production Research* 47.23 (2009), pp. 6611–6631.

[76] F Vercauteren. "Final Report on Main Computational Assumptions in Cryptography." In: *European Network of Excellence in Cryptology II* 11 (Jan. 2013).

[77] Marko Vukolić. "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication." In: *International Workshop on Open Problems in Network Security*. Springer. 2015, pp. 112–125.

[78] Shu-Ching Wang, Kuo-Qin Yan, Chin-Ling Ho, and Shun-Sheng Wang. "The optimal generalized Byzantine Agreement in Cluster-based Wireless Sensor Networks." In: *Computer Standards & Interfaces* 36.5 (2014), pp. 821–830. DOI: 10.1016/j.csi.2014.01.005. URL: https://doi.org/10.1016/j.csi.2014.01.005.

[79] *What is Supply Chain Management? - SCM*. 2017. URL: https://scm.ncsu.edu/scm-articles/article/what-is-supply-chain-management.

[80] *World Shipping Council*. 2013. URL: http://www.worldshipping.org/about-the-industry/containers/global-container-fleet.

[81] Ian Wycherley. "Greening supply chains: the case of The Body Shop International." In: *Business Strategy and the Environment* 8.2 (1999), pp. 120–127. ISSN: 09644733. DOI: 10.1002/(sici)1099-0836(199903/04)8:2<120::aid-bse188>3.0.co;2-x.

[82] Qiannan Zhang, Tian Huang, Yongxin Zhu, and Meikang Qiu. "A Case Study of Sensor Data Collection and Analysis in Smart City: Provenance in Smart Food Supply Chain." In: *IJDSN* 9 (2013). DOI: 10.1155/2013/382132. URL: https://doi.org/10.1155/2013/382132.

[83] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, and Huaimin Wang. "Blockchain Challenges and Opportunities: A Survey." In: *International Journal of Web and Grid Services* January (2016), pp. 1–24. DOI: 10125/41338. URL: http://inpluslab.sysu.edu.cn/files/blockchain/blockchain.pdf.

[84] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends." In: *2017 IEEE International Congress on Big Data, BigData Congress 2017, Honolulu, HI, USA, June 25-30, 2017*. 2017, pp. 557–564. DOI: 10.1109/BigDataCongress.2017.85. URL: https://doi.org/10.1109/BigDataCongress.2017.85.