University of Nevada, Reno

Integrating Blockchain into Supply Chain Safeguarded by PUF-enabled RFID.

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in
Computer Science and Engineering

by

Md Didarul Islam

Dr. Haoting Shen - Thesis Advisor
Dr. Shahriar Badsha - Thesis Co-Advisor
May 2021

THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

MD DIDARUL ISLAM

entitled

Integrating Blockchain into Supply Chain Safeguarded by PUF-enabled RFID

be accepted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

Haoting Shen, Ph.D.
Advisor

Shahriar Badsha, Ph.D.
Co-advisor

Mohammed Ben-Idris, Ph.D.
Graduate School Representative

David W. Zeh, Ph.D., Dean
Graduate School

May, 2021

Abstract

Due to globalization, supply chain networks are moving towards higher complexity and becoming vulnerable to various kinds of attacks such as counterfeiting, information tampering, and so on. Appropriate approaches are necessary to tackle different types of attacks and to ensure the required supply chain security. In this thesis, we have addressed the product counterfeiting issue using Physical Unclonable Function (PUF) enabled Radio Frequency Identification (RFID) tag. Applying blockchain technology to supply chain can add many useful features to the supply chain, such as decentralization and immutability. On the other hand, linking supply chain products to blockchain can bring transparency, traceability, and non-repudiation as well. As a preferred alternative to the traditional centralized databases, blockchain can address certain supply chain management issues such as complicated record-keeping, provenance tracking of the products, and distrust among different supply chain parties. In this research, blockchain technology has been leveraged to support anti-counterfeiting and deal with data attacks. We have also introduced a reputation-based consensus algorithm for the blockchain which is less resource-intensive and thus will not impose additional cost on supply chain products indirectly. In the same research direction, we have devised our system architecture that is suitable for lightweight supply chain devices. The proposed three protocols, namely: registration protocol, verification protocol, and transaction protocol along with the blockchain technology help to transfer the ownership of the authentic product and keep the sensitive supply chain information safe. An encryption-based secret sharing technique has also been introduced to assist data protection.

Dedication

Dedicated to Rukaiya Islam (Aurora)

Acknowledgments

I would like to extend sincere thanks to my advisor, Dr. Haoting Shen, who was a constant source of inspiration and enlightenment required during all the phases of my research.

I also thank my co-advisor Dr. Shahriar Badsha for his guidelines.

My gratitude to the graduate school representative Dr. Mohammed Ben-Idris for taking time to review this thesis and providing guidance on my research.

I thank my family for supporting me throughout this journey.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

A supply chain refers to the interconnected and synchronized system of all related organizations, personnel, resources, processes, information, and technology involved in the production and successful delivery of a finished product to the end user, including the moving of the product throughout the whole chain from the very beginning of the supply of the raw materials.

Nowadays, some large-scale supply chains are so complex that it is almost impossible to prevent all the attacks, as the adversaries are constantly in search of the slightest weak point and attack it. Therefore, the operation of such large supply chains which are comprised of numerous interconnected entities will definitely face some security, privacy, trust, and other concerns that may put the whole supply chain at risk and affect the connected entities. Although much research has already been conducted to address these concerns, still it is a difficult problem to resolve and a lot of aspects need to be explored.

1.1 Motivation

There were multiple motivations behind this research work. Also, the system architecture was derived to address some issues. All these motivations are discussed in this section.

1.1.1 Supply chain attacks

Modern-day supply chains are expanding day by day in terms of connected parties, resources, and complexity. As there are lots of entities involved in these supply chains; sharing products, personnel, and information with business partners is unavoidable. This also increases several supply chain security issues. These security issues should be dealt with a high priority as a breach within the system could lead to loss of intellectual property, loss of revenue, and inefficient delivery schedules. In addition to that, delivering unauthorized or tampered products could be harmful to customers and may lead to unwanted lawsuits. The primary focus of a secured supply chain is to ensure the integrity of the product and the secondary target is to protect the supply chain related sensitive information during all stages of the supply chain. The devastating consequences of supply chain attacks can be imagined from the statement that: the US semiconductor industry is losing over $7.5 billion per year due to counterfeit electronic components [1].

In our system design, we have addressed both issues. Our system will work against the counterfeiting of the product and will also protect the system-related secret information.

1.1.2 Lightweight supply chain devices

Traditional supply chain devices are usually handheld RFID tag readers with smartphones. These types of devices are battery-operated and highly power constraint. Although sometimes the RFID tags come with their own power supply, in most cases RFID tags are passive types and depend on the power of RFID tag readers to work. So, when scanning an RFID tag, the reader also loses a significant amount of energy. Considering these factors, it is not feasible to run high-energy-required services like blockchain on lightweight supply chain devices.

Our system architecture works in such a way that lightweight devices do not need to run blockchain in themselves, whereas lightweight devices can communicate with the blockchain network and exploit the facility of blockchain.

1.1.3 Open supply chain

Another motivation behind our system architecture is that we have considered an open supply chain, which emphasizes the fact that any prospective party can join. In other words, at the beginning of the supply chain, it is not defined that which party will join next. To create a blockchain network a certain number of nodes are required. This requirement might not be fulfilled if we try to run the blockchain system within the supply chain parties at the beginning of the supply chain.

1.2 Contribution

Applying blockchain technology to supply chain can bring many useful features to the supply chain, such as decentralization, immutability, transparency, traceability, and non-repudiation as well. Blockchain can also address certain supply chain management issues such as complicated record-keeping, provenance tracking of the products,

and distrust among different supply chain parties.

We have addressed the product counterfeiting issue using PUF enabled RFID tag. Blockchain technology has also been leveraged to support anti-counterfeiting. The proposed protocols along with the blockchain technology help to keep the sensitive supply chain information safe. Our system architecture is suitable for lightweight supply chain devices. We have also used a reputation-based consensus algorithm for the blockchain which is less resource-intensive and thus will not indirectly impose additional cost on supply chain products.

Last but not the least, we have used 'secret sharing' mechanism to store the challenge-response pairs of the PUF in the blockchain so that if any blockchain node is compromised, the information is not leaked. Encryption technique has also been imposed on the secret sharing so that other nodes which were not intended, do not have the secret information from the blockchain.

The remainder of this thesis is organized as follows: Chapter 2 deals with the related background and literature review. Threat model for our proposed system is stated in Chapter 3. The system design is described in Chapter 4. In Chapter 5, the security and privacy analysis of our proposed system is performed. The result analysis part is illustrated in Chapter 6. Chapter 7 concludes the thesis with possible future research.

Chapter 2

Literature Review

2.1 Background on Supply Chain

A supply chain refers to the interconnected and synchronized system of all related organizations, personnel, resources, processes, information, and technology involved in the production and successful delivery of a finished product to the end user, including the moving of the product throughout the whole chain from the very beginning of the supply of the raw materials. A supply chain is the network of organizations and service providers where they work together to maintain the quality of the product, and the protection of information, equipment, and facilities in every stage of the supply chain [2]. Organizations directly involved in the supply chain are raw material suppliers (such as mines, farms), IP (Intellectual Property) owners, manufacturers or producers, assemblers, transportation, distributors, wholesalers, retailers, end users, and so on. A very simplified structure of a basic supply chain can be visualized in Figure 2.1. However, there are other lots of entities who participate in the supply chain for the smooth functioning of the system such as inventory managements like

warehouses, vendors, logistic service providers, intermediate companies, financial institutes (e.g. banks), government agencies, etc. [3]. These organizations or companies interact with each other not only for the better performance and enhanced security of the supply chain, but also to make the supply chain global which necessarily requires the movement of the products along with necessary personnel and equipment across the international borders.

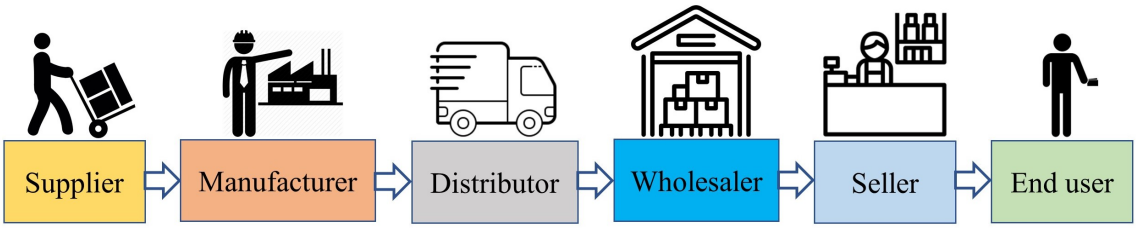

Figure 2.1: Different stages of a basic supply chain.

2.1.1 Related Supply Chain Attacks

Supply chains might suffer from several security issues and the adversaries may commit various kinds of attacks on supply chains to achieve their goal. Major supply chain attacks closely related to our research work are discussed in this section. The state-of-the-art countermeasures of these attacks are also mentioned in this section.

1. Counterfeits

Counterfeit or fake products are the main concern of any supply chain. These types of products' or components' quality are significantly degraded than expected and they compromise the basic purpose of a secured supply chain. Counterfeit products cause lots of revenue loss in any industry, downgrade the reputation of manufacturers and sellers. An example of counterfeit electronics products can be drawn here to demonstrate the gravity of the situation. In 2008, the FBI seized $76 million of

counterfeit Cisco networking equipment which was sold to the U.S. Navy, the U.S. Marine Corps., the U.S. Air Force, the U.S. Federal Aviation Administration, and the FBI [4]. Most common types of counterfeit products are as follows [5] [6]:

i **Adulterated:** In this type of counterfeit product one or more component(s) of the product is fake. Adversaries do that to reduce the production cost and to represent the product as authentic at the same time.

ii **Recycled:** For the case of recycled products, components are taken from a used system, repackaged, and then resold in the market as fresh. These recycled parts are either damaged or the previous usage may have severely reduced their performance.

iii **Cloned:** An illegal replica of a product is referred to as cloned. Cloning can be accomplished by two means – by reverse engineering, or by obtaining IPs illegally.

iv **Overproduced/over-run:** nowadays companies give contracts to different factories to produce different components. When an unfaithful factory produces components in an excess amount outside of the contract and sells them in the open market, the components are referred to as overproduced or over-run.

v **Defective:** Products which were found out-of-specification during the manufacturing tests, might be sold in the open markets intentionally or unintentionally. These components are a serious threat to any sophisticated system.

vi **Information tampered:** Faking the quality information of a product is also considered counterfeiting. The actual information of a lower quality product is edited to a higher one.

2. Data attacks

As the supply chain grows bigger and bigger, the data linked with the supply chain also increases in volume. The consistency of data throughout the supply chain is one of the prime requirements for a resilient supply chain. For any particular organization, the average number of third parties with sensitive information access is increasing day by day. Data attacks from any of the parties may expose the organization to high threats. A study conducted by the Ponemon Institute in 2018 found that 56% of organizations suffered a data breach caused by one of their vendors [7]. Some noteworthy attacks on supply chain data are discussed below:

i **Data breach:** Data breach results from the attack on data integrity. Inappropriate disclosure of personal or confidential data may occur when the authorization or control of access is compromised [8].

ii **Malicious insertion of system data/information:** This attack refers to the insertion of faulty data into the supply chain system. Substitution and alteration of vital data such as design, manuals, architectures, and roadmaps, etc. also fall under this category [9].

Apart from the above-mentioned product counterfeiting and data attacks, several other device-level attacks can be pulled off on product security devices to facilitate counterfeiting. These types of attacks include RFID tag disabling, RFID tag switching, RFID tag cloning, Denial of Service, Distributed Denial of Service, and so on. Although these attacks are very specific for supply chains, supply chains can always be affected by some common forms of network attacks to tamper or eavesdrop the vital information of the supply chain, such as Man-in-the-middle attack, Sniffing attack, Spoofing attack, etc. But these attacks are not in the scope of this research work.

2.1.2 Present techniques to address those attacks:

1. Anti-counterfeiting

Until now, counterfeit products are being tried to be identified and prevented by using some primitive techniques. Quality information of the raw materials and finished products such as temperature, vibration, humidity, aroma, and so on are collected using different kinds of sensors. In addition to that, several analytical techniques such as X-Ray, microscopy, and chemical analysis are also performed to detect the counterfeit products [10]. For the identification purpose of the products, which is one of the initial requirements to perform the anti-counterfeiting, many identifying technologies such as barcodes, QR codes, RFID tags, NFC tags, phosphor PUF [11], LASER security mark, and digital watermarking are in practice now. As counterfeiters are getting smarter day by day, researchers are working more and more on advancing the techniques of anti-counterfeiting.

In one such work, Al-Bahri et al. (2019) [12] have proposed an anti-counterfeiting system based on digital object identifier (DOA). Here, a digital object identifier (DOI) is encoded in a physical identifier and the physical identifier is attached with the desired product. To verify the product, the DOI is retrieved by decoding the physical identifier and compared with a trusted reference.

In Pathak (2010) [13], the author has devised a mechanism named authenticated product label (APL) based on digital signatures, where not only the counterfeit product is detected but also the source of the counterfeit in the supply chain can be pinpointed.

In another work, Tian (2016) [14] has researched agri-food supply chain traceability system, where the RFID and the blockchain technology have been exploited. This system utilizes RFID technology to implement data acquisition and data sharing

in multiple stages of the agri-food supply chain. The intention of using blockchain technology is to ensure that, the information which has been shared and published in this traceability system is authentic and immutable.

2. Addressing data attacks

The consequence of the vital supply chain data getting into the wrong hand might be detrimental. Few works dealing with supply chain security are mentioned here.

In one such work, Ahemd et al. (2017) [15] have explained that each device in the network should come with an error detection mechanism to minimize the risk of data tampering. Some noteworthy error detection mechanisms in use are parity bit, checksum, etc. They have also suggested applying cryptographic hash function to make the data more secure. In addition to that, different authentication mechanisms such as point-to-point encryption can be exploited to achieve data privacy by preventing illegal access to the nodes of the network.

In another work, Reed et al. (2014) [9] have discussed how cryptography can be utilized to limit the malicious insertion of system data. Digital signatures, encryption, checksums, and/or other cryptographic techniques can be used to verify the source authenticity of all received data/information. Apart from that, critical and sensitive system information concerning the design, development, maintenance, and delivery is assessed for its trustworthiness. This vital data/information is monitored from origination to storage and delivery to ensure that the integrity of the information is not violated.

2.2 Background on Blockchain

Blockchain is a distributed ledger technology. This technology first evolved from the concept of cryptocurrency Bitcoin which was introduced by Satoshi Nakamoto [16]. Generally speaking, blockchain is an immutable (write once and read-only), decentralized and shared database where all the transactions within the system are recorded. Based on the permission policy of the system, allowed nodes can join the system, perform and verify transactions.

2.2.1 Features of blockchain

1. Basic Structure of Blockchain

A blockchain network consists of multiple nodes that record and share all transactions that occur within the network [17]. Multiple transactions are incorporated in a single block. Before adding the new suggested block to the blockchain, it is verified by the network nodes that the block contains valid transactions and refers to the right previous block through a cryptographic pointer. How this whole process is done is determined by the consensus mechanism adopted by the blockchain network [18] [19].

Figure 2.2 illustrates the basic structure of a blockchain.

A block contains multiple transactions in the block body. The block header contains parameters like hash of the current block, hash of the previous block, timestamp, and other information. Each block indicates its previous block to maintain the proper sequence among the blocks and create a valid blockchain. This is done by the 'hash of the previous block' field which is the cryptographic pointer mentioned earlier. Obviously, the very first block (termed as the genesis block) in the blockchain cannot refer to any previous block, and for that reason, the 'hash of the previous block' field of the genesis block is necessarily 0 [20] [19].

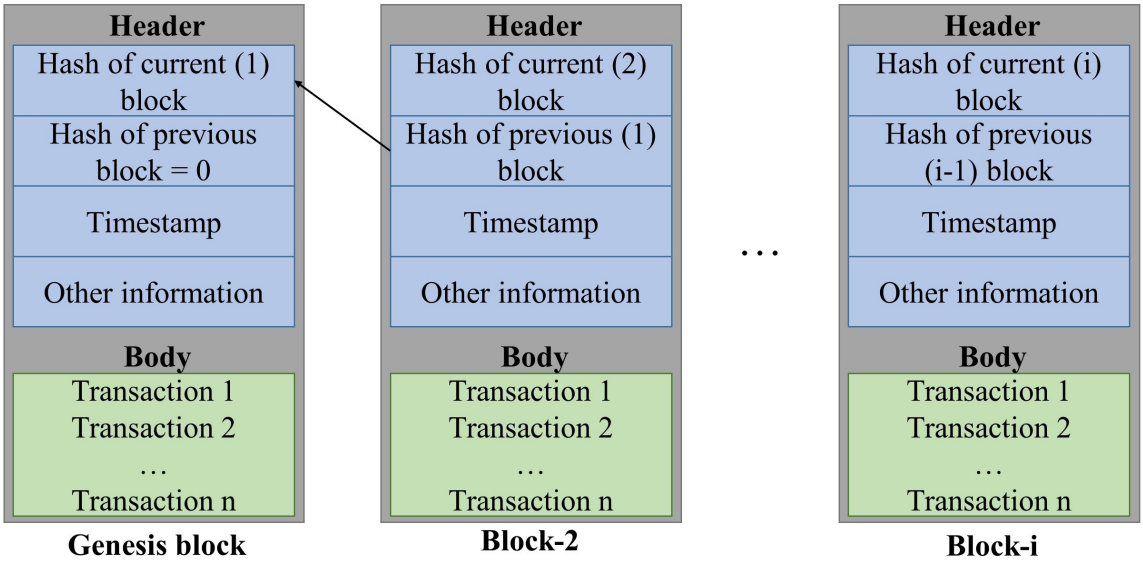

Figure 2.2: Basic structure of blockchain.

2. Blockchain Types

Based on the types of permission used, blockchain technology can be broadly categorized into the following three classes.

i **Public:** A public blockchain is a truly decentralized permission-less blockchain. It is an open distributed ledger where any node can join the network, conduct the transactions, and can also participate in the consensus mechanism without any authentication from the central agency [21].

ii **Private:** A private blockchain is a permissioned blockchain. It is a controlled distributed ledger, where every node can not participate in the blockchain, rather a Certificate Authority determines who can join the network [21]. All permissioned nodes are authenticated, and their identities are known to the whole network [17].

iii **Consortium:** Sometimes it is necessary to coordinate both public and private types of blockchain rather than using completely public or completely private ledger architecture. Here, the blockchain is not controlled by a single entity as in

the case of a private blockchain; rather, it is controlled by a group of approved individuals. The data in blockchain can be public or private, and can be considered as partially decentralized [20] [22].

3. Different Consensus Algorithms

A consensus algorithm is a procedure through which all the nodes present in the blockchain network reach a common agreement about the present state of certain data in the network. Fundamentally, the consensus algorithms achieve reliability in the blockchain network by establishing trust among unknown peers of the distributed system. The main purpose of the consensus mechanisms is to make sure that every new block that is added to the blockchain is verified as valid by the majority of the nodes in the blockchain.

There are several consensus algorithms in practice each with its pros and cons. Some widely used consensus algorithms are briefly discussed below.

i **Proof of Work (PoW):** The main concept of this consensus algorithm evolves around the idea of solving a complex mathematical puzzle and providing the solution. This cryptographic puzzle is bound to a threshold. In PoW, miners perform a resource-intensive computational task intending to generate a PoW that matches the required threshold, and present the block to the other nodes of the network. Once the majority of the nodes in the network verify that the block is authentic, the block is appended in the blockchain [16] [23] [24].

ii **Proof of Stake (PoS):** In PoS, the nodes who want to participate in the block creation process must prove that they own a certain amount of resources. They must lock up this resource as a stake to guarantee that they will behave as per the network rules. The higher the stake of a node is, the greater the chance of this node is being selected as a validator. The validator will lose its stake if it

validates a wrong block, and this way PoS reduces the possibility of the validator's misbehavior [25] [23] [24].

iii **Practical Byzantine Fault Tolerance (PBFT):** In PBFT, nodes are sequentially ordered in such a way that one node is elected as a leader node and others are considered as backup nodes. Honest backup nodes are supposed to follow the instructions of the leader node. The communication level is pretty high among the nodes as they want to eliminate any false information in the network by verifying them through several stages (Request, Pre-prepare, Prepare, Commit and Reply). Provided that the maximum number of malicious nodes is not greater than or equal to one-third of all the nodes in the system, PBFT can achieve consensus [26] [27].

Among the other consensus algorithms, Proof of Authority, Proof of Capacity, Proof of Burn, Proof of Elapsed Time, etc. can be mentioned.

2.2.2 Blockchain Related Works

As an emerging technology, a lot of research work is going on related to blockchain and these works are enhancing blockchain technology gradually. For instance, Ferdous et al. (2020) [24] have provided an insight into the layered architecture of blockchain. They have also discussed the taxonomy of different consensus algorithms in detail. Furthermore, researchers are exploring several application areas of blockchain. In their research work, Fernández-Caramés et al. (2018) [28] have mentioned few application areas of blockchain apart from cryptocurrencies, such as supply chain management, financial transactions, healthcare, industry 4.0, intelligent transportation systems, energy management systems, telecommunications, defense & public safety, government law enforcement, farming, data storage, timestamping services, mobile

crowdsensing and so on. Not only they have explained how applying blockchain technology can improve these sectors, but also they have discussed the challenges (e.g. privacy, security, energy efficiency, speed, and scalability) of integrating blockchain in these fields.

It is highly relevant to mention that blockchain technology is also playing a significant role in terms of solving security issues in areas that are closely related to the supply chain, such as IoT, Industrial IoT (IIoT), manufacturing, and so on. Researchers are applying blockchain technology into these fields and constantly trying to make these areas more security attack resilient. In one such work, Cha et al. (2018) [29] have proposed a Blockchain Connected Gateway for Bluetooth Low Energy (BLE) enabled IoT devices to maintain secure and adaptive user privacy. There are mainly three types of participants in the system architecture. They are: (1) the owners of the IoT devices, (2) the BC gateway administrators, and (3) the end users. User privacy is maintained because the gateway prevents users' sensitive data from being accessed without user permission.

In another work, Kim et al. (2017) [30] have proposed a Blockchain of Things (BoT) model to overcome problems related to the hacking of IoT devices. They proposed a methodology to overcome the security vulnerabilities by presenting a color spectrum chain among blockchain consensus algorithms.

Lin et al. (2018) [31] have proposed a blockchain-based system named BSeIn for Industry 4.0, to enforce secure mutual authentication remotely with fine-grained access control. In their research work, they have also illustrated the security requirements of such a blockchain-based mutual authentication for Industry 4.0 deployments. Then they have explained how BSeIn can ensure these security requirements.

2.3 Background on PUF

2.3.1 Features of PUF

Physical Unclonable Functions (PUFs) are circuits that utilize the natural variation that occurred during the fabrication processes of devices [32]. To extract the intrinsic property of the device, input challenge bitstreams are given to PUF which is mapped into output bitstreams and collected as responses [1]. This unique random natural variation within the device might be considered as 'digital fingerprint' of that device and can be used for the device identification purpose.

1. Properties of PUF

A PUF has some properties which come naturally and make the circuit unclonable. PUF is simple to implement and evaluate, and also works as a completely random function [33]. In order to be considered as an authentication technique, PUF must exhibit these properties. These properties can be summarized as follows [34]:

i Same challenge (C_1) will produce same response (R_1) for the same PUF. Any other challenge (C_2) will not be able to generate the same response (R_1) for that PUF.

ii Same challenge (C_1) will produce different responses (R_1 and R_2 respectively) for different PUFs (PUF_1 and PUF_2 respectively). In other words, 2 responses will never be same for different PUFs for the same challenge.

iii Different challenges will never produce same responses for same PUF. If challenge C_1 produces response R_1 and C_2 produces response R_2 respectively for the same PUF, R_1 and R_2 will never be same.

2. PUF classifications

PUFs can be classified based on the strength of the PUF and based on the implementation of the PUF circuit. Both of them are discussed with their subcategories below:

A. Based on strength

The strength of the PUF depends on the number of challenge-response pairs (CRPs) that can be generated from a single device [35]. Based on strength, PUF can be classified in the following categories:

i **Strong PUF:** Strong PUF supports a very large number of CRPs. The number of these pairs is so large that even if an attacker has access to the PUF they cannot possibly record them all. This type of PUF is frequently used for device authentication purposes

ii **Controlled PUF:** This type of PUF has a strong PUF inside which is supported by a control logic. The challenges are pre-processed before passing to the strong PUF and the responses are processed before coming out of the strong PUF, the CRPs of the strong PUF are never accessed directly. The outputs that are given to the outside world are the processed outputs and the actual outputs from the strong PUF are never revealed.

iii **Weak PUF:** This type of PUF has very few numbers of fixed challenges. Weak PUFs can be used for entity authentication techniques and secure key generation/storage in cryptography.

B. Based on implementation

PUF can also be classified based on how the PUF circuit is implemented. Depending on the different PUF implementation techniques PUF can be categorized into [1]:

i **Cover based:** a coating layer with a random dielectric coefficient is distributed and the capacitance variation over that coating generates the random response.

ii **Delay based:** The propagation delay between identical circuits is used to derive responses. Examples of delay based PUFs are: Arbiter PUF, Ring Oscillator PUF, Tristate buffer PUF.

iii **Memory based:** They produce an output response based on the unpredictable startup state of memory devices such as latches, flip flops, and Static RAM (SRAM). E.g. SRAM PUF, Butterfly PUF, Flip-flop PUF.

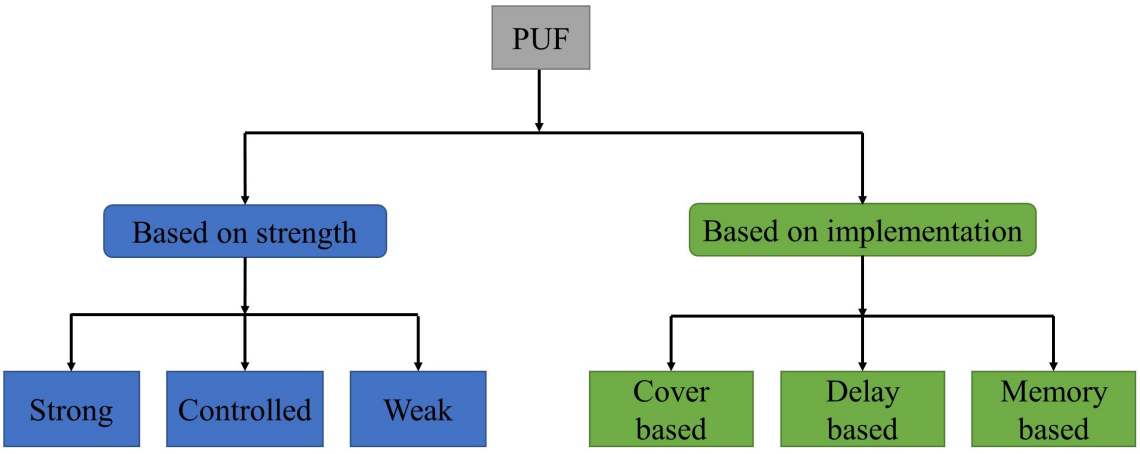

Figure 2.3: Classifications of PUF.

3. PUF working principle

Figure 2.4 shows an arbiter PUF consisting of MUXes and a latch. The circuit has a multiple-bit control signal (challenge) and a single-bit output (response). Here, two

MUXes are controlled by the same input bit. The input bits determine the delay paths, which are crossing through the MUXes. Thus the circuit creates two delay paths for each challenge. To evaluate the response for a particular challenge, a rising input signal is given to both paths at the same time, the signals race through the two delay paths and the final output is collected from the latch [36].

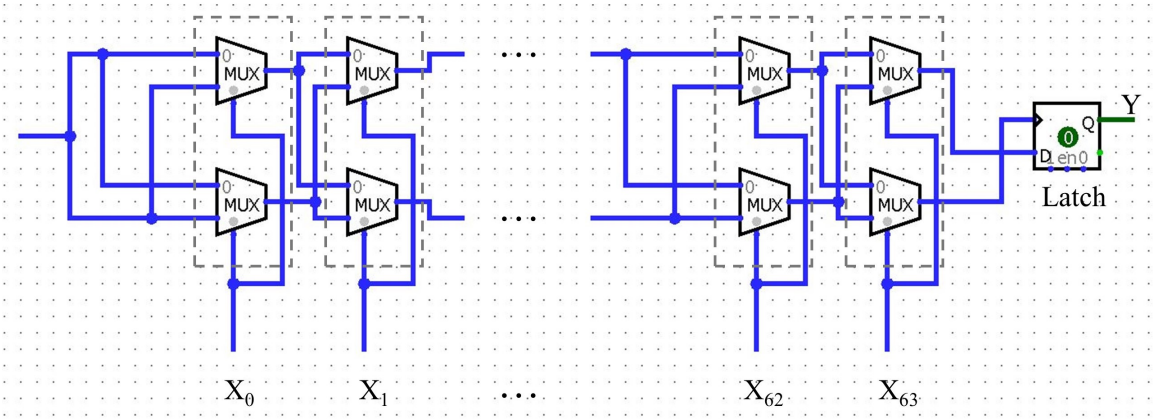

Figure 2.4: An arbiter PUF circuit.

There are two ways to construct a k-bit response from the 1-bit response for this PUF circuit. First, a challenge is used as a seed to generate k pseudo-random numbers. Then, the PUF circuit is evaluated k times, using the k different pseudo-random numbers used as controller bits. Each evaluation will provide 1-bit response, and thus k evaluations will give k bit response.

Second, single-output PUF circuit is cascaded k times to obtain k bits response with a single challenge. Here, the same challenge is repeated k times and response is collected at the output of every stage [37].

2.3.2 PUF Related Works

In Mohanty et al. (2020) [34], the authors have proposed a novel IoT architecture where the blockchain integrates PUFs for robust security and data management,

called "PUFchain". Applying blockchain for IoT applications will suffer some inherent challenges such as scalability, latency, and high power consumption, and security and privacy issues. To overcome these issues this paper also presents a new consensus algorithm called "Proof of PUF-Enabled Authentication" (PoP) so that PUF integrated blockchain can be applied to the IoT environment.

In another work, Aman et al. (2017) [38] presents a PUF based efficient mutual authentication protocol in IoT systems. This protocol exploits hashing and nonce for authentication purposes. The proposed mechanism requires very low communication overhead and memory requirements, and does not need any secrets to be stored locally. They have present security and performance analysis of their proposed protocol and shown that it is safe from physical and side-channel attacks. One drawback of this protocol is that, the IoT device requires hashing which might be a constraint for low-power IoT devices.

Islam et al. (2019) [39] have proposed a PUF integrated blockchain-based IC supply chain traceability system that allows a potential customer to both trace and track the IC. For automated execution of enrollment, authentication, ownership traceability, and ownership transfer of an IC, they have deployed a smart contract in blockchain. The clauses of the smart contract are set in such a way that, only the registered genuine IP owners claim the initial ownership of an IC and write the relevant PUF data in the blockchain; and only the current owner of the IC can create a new transaction. They have also demonstrated their proposed solution in the Ethereum blockchain.

2.4 Blockchain based Supply Chains

Finally, we can shed some light on the research where the authors have worked on the intersection area of supply chain and blockchain. In one such work, Bocek et al.

(2017) [40], the authors have focused on the pharmaceutical supply chain. They have considered the start-up Modum.io AG as the research subject and illustrated how this company has integrated blockchain into this pharmaceutical supply chain. In the system architecture, it is shown how IoT sensors are monitoring the temperature of each parcel and transmitting these values to the mobile devices working in the front-end. In the back-end, blockchain with ethereum smart contract is used to collect, verify and store these temperature values in a secured manner.

However, for their system's working principle, they have mentioned that temperature sensors compatible with Bluetooth technology are programmed to send data in a fixed interval to mobile devices. The security analysis of this communication is not performed. The data sent by the sensor to the mobile device might be eavesdropped by adversaries since this data is being sent proactively in a repetitive manner. We have minimized this type of insecure data transmission issue in our research work by using the mutual authentication between the RFID tag and the reader.

Helo et al. (2019) [41] have conducted a bit generalized research on incorporating blockchain into supply chains. Rather than focusing on a particular type of supply chain, they have indicated some fields of the supply chain, such as asset, identity, transactions where blockchain technology can be applied. After they have proposed a blockchain-based logistics monitoring system architecture, they have performed a software implementation of this system. Since they have considered a generalized scenario, system details such as type of products, details of the supply chain, tag/sensor description are not specified. The verification and transaction actions are just mentioned briefly, whereas proper sequence diagrams or algorithms are missing. These issues are specifically addressed in our research work. They have used a local database to store completed transactions for fast search and retrieval, whereas blockchain is providing transaction immutability. However, if either the local database or any of

the blockchain nodes get compromised, then supply chain related sensitive information might get into the hand of the adversaries, although they cannot tamper any ongoing transactions. This attack is handled very well using the encryption-assisted secret sharing technique in our work.

In another work, Mondal et al. (2019) [42] have used a test prototype of the RFID integrated sensor in their paper for the food supply chain. The RFID integrated sensor is attached to a food package from where information regarding the package can be extracted. When the retailer, seller, logistic, or storage terminals scans RFID integrated sensor, transaction information is broadcast to all other participating terminals. After all the acknowledged transactions are received by the network, the network validates the transaction into a block by authenticating all the acknowledgments and includes the block information in the blockchain. A new parameter named confidence level is introduced to authenticate a transaction in case of a lower number of acknowledgments is received than expected. Once consensus is reached with desired confidence level about a transaction, the transaction is validated into a block.

The limitations of their work are as follows. When a lower number of acknowledgments are received than expected for a particular transaction, this transaction can still be validated into a block using the parameter called confidence level, and this parameter is determined by the managers of the network. So there is still a chance of error and data manipulation by collusion. We have tackled this scenario by making sure that a transaction is accepted only when all the messages from the validators are the same. In addition to that, they have randomly selected a node from a set of nodes to verify each transaction acknowledgment. Whereas we have chosen a group of representative nodes based on their reputation to validate transactions. From their simulation result, it is visible that, the probability of fake block being appended is around 10% for the case of 100 total nodes containing 10 malicious nodes.

The probability of a false message being appended in the blockchain is nearly 0 (with an appropriate number of representative nodes) for our system for the same network scenario.

To the best of our knowledge, our research work has addressed a lot of privacy and security related issues in the field of blockchain based supply chains including the above-mentioned drawbacks. Other details of how our research work is advancing the blockchain based supply chain technology can be found out throughout the thesis.

Chapter 3

Threat Model

This section describes the threat model for our proposed system. Here we have mentioned the possible attacks on our system and also stated the assumptions made on the participating entities in the system. For our research, we have assumed that the registration of any product is authentic. The manufacturer will not register fake products in the system as genuine. The manufacturer will do that naturally as the customer review of the fake products will demolish the company's reputation and thus will cause a loss of revenue.

Any supply chain party can act maliciously, although they were registered in the network as their previous party considered them trustworthy.

Blockchain nodes can also be malicious. It is not possible to know beforehand which node is going to act maliciously but only after the adversarial behavior. We have considered maximum 50% malicious nodes in the system.

We have listed few possible attacks for our blockchain based supply chain system. Among these attacks, the former two attacks are counterfeiting attacks, whereas the latter two can be considered as data attacks. However, data attacks can also assist product counterfeiting in some cases. The probable attacks for our proposed systems

are as follows:

Counterfeiting attack: Malicious supply chain parties will try to sell counterfeit products as authentic to benefit more. This counterfeiting activity can be performed in several ways.

Supply chain parties can create counterfeit products that resemble the original products and sell these fake products by attaching fake tags with them. They can also remove the original tag from an authentic product and attach it with a counterfeit product.

For extreme cases, malicious people outside of the supply chain (external adversary) can get the original product in hand, remove the original tag from the authentic product, attach it with a counterfeit product and sell it outside of the supply chain claiming that the product is a genuine one.

These types of attacks may circumvent the buyers and they might buy the fake product considering genuine.

Product double-spending attack: Supply chain adversarial parties can get the original tag information and produce multiple copies of the original tag. This way the adversaries try to double-sell an already sold product. This can be attempted in the following manner:

When a product along with the tag is being handed over to a malicious supply chain party, the attacker can clone an original RFID tag and attach the cloned tags with the counterfeit products. Similarly, instead of physically cloning the RFID tag, the supply chain party can try to collect the CRP information and mimic the tag using a programmable RFID tag.

These fake products with the cloned RFID tags will be interpreted as genuine and thus will pose a severe threat to the consumers.

Malicious and fake representative nodes: Since we are using representative

nodes as a communicating bridge between the supply chain parties and the blockchain network, two adversarial scenarios might occur here. First, the selected representative nodes might work maliciously. Second, few blockchain nodes who are not selected as representative nodes (fake representative nodes), might express themselves as representative nodes.

Malicious representative nodes' activities might be: representative nodes' leader node can send false messages to both the supply chain parties and the blockchain network. And, representative nodes' backup nodes can vote leader node's message maliciously.

On the other hand, fake representative nodes' adversarial attack is: unauthorized or not selected blockchain nodes can collude together and send their address as representative nodes to the supply chain party.

For either of the above-mentioned attacks, the adversaries try to manipulate the communication between the supply chain parties and the blockchain network and thus control the supply chain related vital information for their benefit.

Secret share manipulation: As supply chain related secret (i.e. CRP) is being split into shares and distributed into specific blockchain nodes, we can avoid the possibility of information leakage. However, blockchain nodes who are holding secret shares can manipulate their corresponding shares and send them when requested in order to make a fake product pass as genuine or an authentic product fail as counterfeit. If enough malicious nodes collude together and manipulate a significant amount of shares, then their attack might become successful.

How our proposed system is resilient enough to handle the above-mentioned counterfeiting and data attacks are discussed in detail in Chapter 5.

Chapter 4

System Design

This chapter describes the system architecture along with the proposed protocols in detail. The blockchain section discusses the lightweight blockchain consensus algorithm and its security analysis. Blockchain structure for our system is also described here. A brief discussion regarding the system's supply chain, PUF and secret sharing is also provided.

4.1 Supply chain

For our system design, we have considered non-perishable or non-edible supply chain items or products. It is also assumed that the supply chain starts from the manufacturer/IP owner, not from the raw materials supplier for simplicity.

Our proposed architecture supports an open supply chain, so that any party whom its previous party thinks trustworthy can join.

Supply chain parties are lightweight devices such as RFID tag readers with smartphones. Although these devices are not capable of high computational operations such as continuous hashing, they will be able to perform simple tasks such as encryption,

decryption, signing, verification, and communication.

4.2 Blockchain

For our research work, we have considered consortium blockchain. Some selected or preferred nodes may create this network and these nodes do not suffer power constraints.

Block structure along with the genesis block are described below:

4.2.1 Block Structure

1. Genesis block

Genesis block is the beginning block of a blockchain. For our supply chain environment, this block contains the necessary registration information of the product.

<table>
<tr><td colspan="8">Hash of the current (genesis) block</td></tr>
<tr><td colspan="8">Block number</td></tr>
<tr><td colspan="4">Tx-1</td><td colspan="4">Tx-2</td></tr>
<tr><td colspan="4">Manufacturer ID(1)</td><td colspan="4">Manufacturer ID(2)</td></tr>
<tr><td colspan="4">Tag ID(1)</td><td colspan="4">Tag ID(2)</td></tr>
<tr><td rowspan="4">CRP(Tag1)-LUT</td><td>Secret</td><td>Node</td><td>Stored in BC as</td><td rowspan="4">CRP(Tag2)-LUT</td><td>Secret</td><td>Node</td><td>Stored in BC as</td></tr>
<tr><td>Sec1</td><td>Node-1</td><td>ENC.pubNode-1(Sec1)</td><td>Sec1</td><td>Node-1</td><td>ENC.pubNode-1(Sec1)</td></tr>
<tr><td>Sec2</td><td>Node-2</td><td>ENC.pubNode-2(Sec2)</td><td>Sec2</td><td>Node-2</td><td>ENC.pubNode-2(Sec2)</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr>
<tr><td colspan="4">Tag ID(1) ← Product details</td><td colspan="4">Tag ID(2) ← Product details</td></tr>
<tr><td colspan="4">Next party ID(1)</td><td colspan="4">Next party ID(2)</td></tr>
<tr><td colspan="4">Representative nodes' list(1)</td><td colspan="4">Representative nodes' list(2)</td></tr>
</table>

Figure 4.1: Genesis block of the blockchain.

i **Hash of the current block:** This is the hash of the genesis block.

ii **Block number:** Position of the block in the blockchain.

iii **Transaction number in block:** Each block may contain multiple transactions. This field indicates the index of the transaction in a particular block.

iv **Manufacturer ID:** Party ID of the manufacturer to be registered and used within the system.

v **Tag ID:** Unique identification number of an RFID tag.

vi **CRP Lookup Table:** This lookup table indicates which secret is stored in which node. Also, the encrypted secrets can be found here for the reference of nodes when they require them for decryption.

vii **Product details:** This field will refer to the product attributes such as product image, manufacturing details, and secret code of the product, etc.

viii **Next supply chain party:** This field will indicate who is going to be the next supply chain party and will register that party in the system.

ix **Representative nodes' list:** For a particular block, the selected representative nodes, based on the consensus algorithm will be listed here. How these nodes are being selected is discussed in detail in the Consensus Algorithm section.

Whereas the genesis block is the very first block of the blockchain, it has a bit different structure compared to other later blocks (since the genesis block contains the product and party registration). All other blocks apart from the genesis block have the same structure.

2. Other blocks

i **Hash of the current block:** Hash value of this block.

Hash of the current block	
Hash of the previous block	
Block number	
Tx-1	**Tx-2**
$Tag_{ID}^{(1)}$	$Tag_{ID}^{(2)}$
$Tx_{ID}^{(1)}$	$Tx_{ID}^{(2)}$
Geo location$^{(1)}$	Geo location$^{(2)}$
Timestamp$^{(1)}$	Timestamp$^{(2)}$
Representative nodes' list$^{(1)}$	Representative nodes' list$^{(2)}$
Receiver$^{(1)}$ (current owner)	Receiver$^{(2)}$ (current owner)
Sender$^{(1)}$ (previous owner)	Sender$^{(2)}$ (previous owner)

Figure 4.2: Other block's structure apart from genesis block.

ii **Hash of the previous block:** Hash value of the previous block, which is used to maintain the sequence of the blocks in the chain.

iii **Block number:** Same as for genesis block.

iv **Transaction number in block:** Same as for genesis block.

v **Tag ID:** Same as for genesis block.

vi **Transaction ID:** Unique ID assigned with each transaction.

vii **Geo location:** Geographic location of the place where the transaction occurred.

viii **Timestamp:** Time of the transaction.

ix **Representative nodes' list:** Same as for genesis block.

x **Receiver:** Current owner of the product after transaction.

xi **Sender:** Previous owner of the product before transaction.

4.2.2 Consensus Algorithm

1. Overview of the Consensus Algorithm

We have proposed a reputation-based blockchain consensus algorithm. The motivation behind this consensus is that it does not require resource-intensive computation. Any algorithm that requires high energy (and thus high cost) if used to protect any supply chain product will eventually impose that cost on that product. Figure 4.3 explains the core concept of the proposed consensus algorithm.

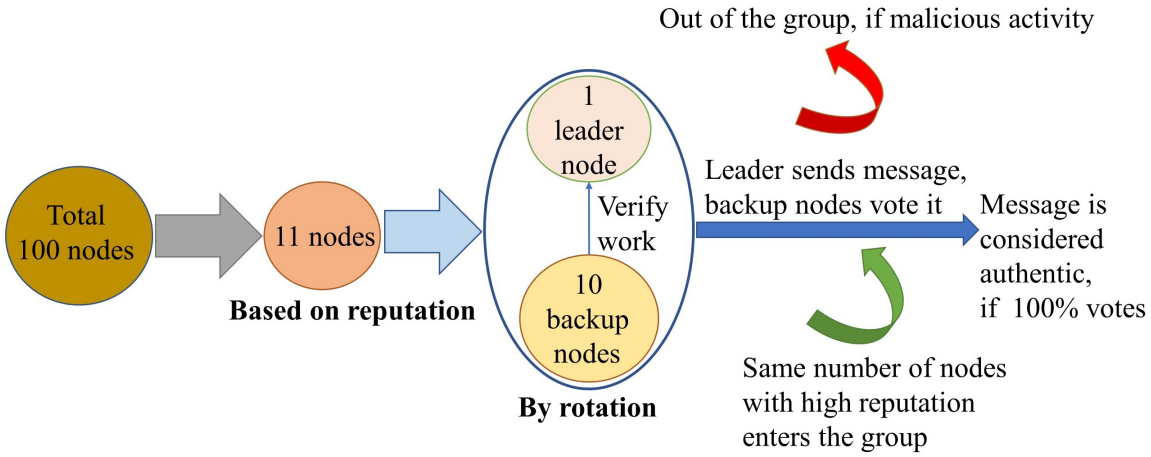

Figure 4.3: Main idea of the consensus algorithm.

For the very first time when the algorithm runs, (10% + 1) of the total nodes are selected as representative nodes based on their reputation. This value is taken as a rule of thumb considering the fact that we are interested in defending against 50% malicious nodes. Reputation comes from how well the nodes have validated transactions, how long they have been active in the network, and so on. The '1' in the '10% + 1' comes from the consideration that, if there are a total of 10 or fewer nodes, then there will be only 1 representative node. Among these nodes, 1 node is selected as the leader and the other nodes remain as backup nodes. Every node among these (10% + 1) nodes will take their turn as a leader by rotation. The whole selection

process is repeated after every node's turn, if required. All messages are sent to the leader as well as to the backup nodes. When the leader tries to append a block in the blockchain or send any message to a supply chain party after verifying the message, backup nodes also verify the message and vote on whether the leader node is acting honestly or maliciously. Any node in the blockchain network will accept the block in their own copy of the blockchain or any supply chain party will accept the message as authentic if the proposed block/message has 100% vote from backup nodes. If any message does not have 100% votes, it means there are malicious nodes present in the representative nodes. The malicious nodes' reputations are decreased by 1 unit, whereas the honest nodes' reputations are increased by 1 unit. And the network again goes for the consensus algorithm to find out the replacement of the malicious nodes by throwing the malicious nodes out of the representative nodes' group and taking in the same amount of highest reputed nodes from the remaining (*total nodes-representative nodes*) nodes. For different types of votes: the majority of the votes are considered authentic, whereas the minority is considered malicious, irrespective of whether the leader is in majority or minority. If the leader is found malicious, the algorithm gives the leadership to the next backup node by turn; and pushes away the leader out of the group by decreasing its reputation and find a replacement by considering reputation. This process continues until consensus is reached (i.e. there are no malicious nodes inside the representative nodes).

However, any supply chain network can set that percentage of required votes. That will necessarily be a trade-off between the security and the speed of the transaction.

2. Security analysis of the consensus algorithm

A question may arise that, what if some malicious nodes group up together and send their addresses as selected representative nodes, whereas they actually are not. The

answer to that question is that,

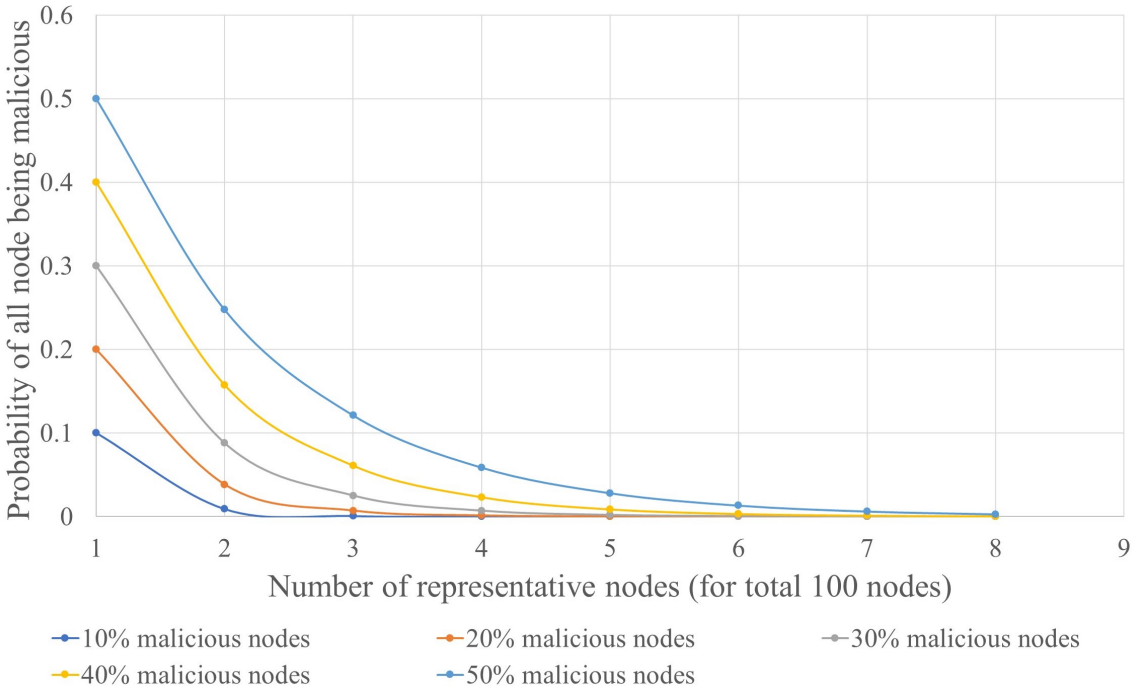

Figure 4.4: Probability of all representative nodes being malicious

Simulation result shows that selecting 8 representative nodes gives almost 0 possibility (for 100, 500, 1000, 1500, and 2000 total nodes respectively) of all nodes being malicious considering 50% malicious nodes. This probability will never become 0 as long as there is a single malicious node in the network. So, it is up to the supply chain network that, based on the security and timing requirements of the system, they can choose (10+1) representative nodes or (10%+1) of the total nodes as representative nodes.

4.3 System Architecture

4.3.1 Layers

In our system architecture, we have two layers, namely: the supply chain layer and the blockchain layer. To avoid confusion, supply chain entities will be mentioned as parties, and blockchain entities will be mentioned as nodes. Supply chain and blockchain will be two completely different systems and will have no overlap with each other. Supply chain parties will be lightweight devices such as RFID readers with mobile phones, whereas blockchain nodes will be high-performance computers with sufficient computational efficiency. While supply chain parties will be entities like manufacturer, distributor, seller, and buyer; blockchain network could be a network that consists of nodes provided by the manufacturer, parties trusted by manufacturer, business associates, government policymakers, and so on. The communication between supply chain and blockchain will happen through representative nodes which are a subset of blockchain nodes and will be selected based on the consensus algorithm for each stage of communication. The system architecture with two layers (supply chain and blockchain) are shown in Figure 4.5

4.3.2 Interlayer communication concept

The high-level communication concept between the supply chain layer and the blockchain layer is explained in Figure 4.6. Whenever any party wants to communicate the blockchain network for any purpose such as registration, verification, and transaction; the party first sends a communication request to the whole blockchain network. We can consider this stage a pre-communication, as the actual communication has not started yet. Once the blockchain network gets this communication request from any party, the blockchain network runs its consensus algorithm and selects representative

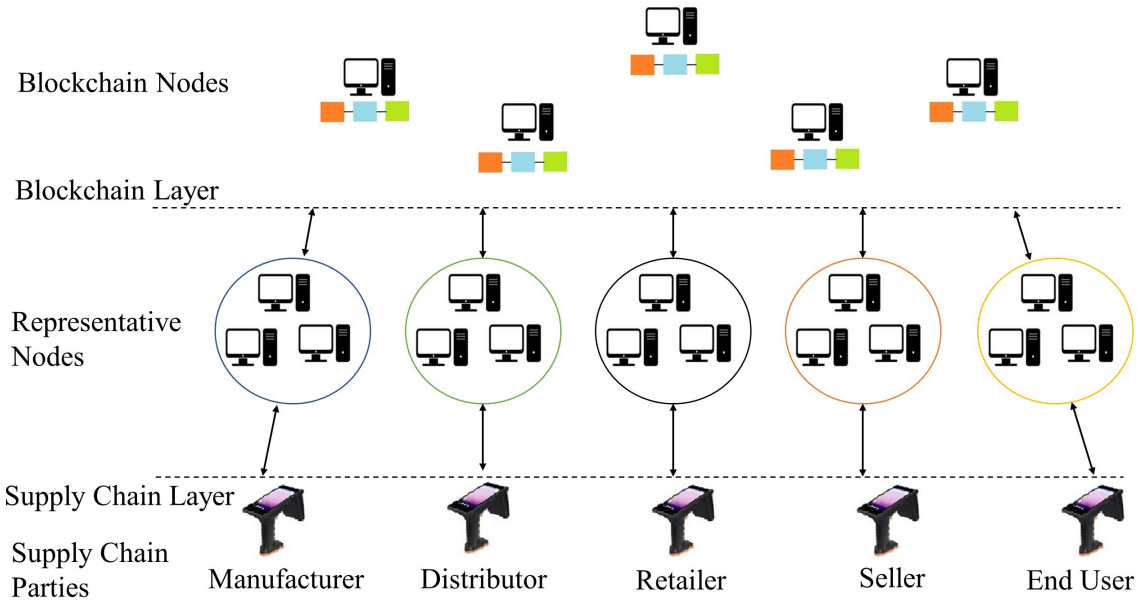

Figure 4.5: Supply chain and blockchain layers.

nodes based on it. Representative nodes consist of a leader node and multiple backup nodes (details provided on Consensus Algorithm section). After that, the leader node sends the addresses of representative nodes to that party and that message is voted by all other backup nodes. From here, the actual communication starts. If the party finds out that all the votes from the backup nodes are the same then the party starts communicating with the representative nodes. Hereafter, if the party requires any information from the blockchain, the information is sent by the leader node and voted by the backup nodes. If the information has 100% backup node votes, then the party accepts the information. Similarly, if any information needs to be updated in the blockchain, it is sent via leader node and backup nodes vote it. If blockchain nodes find out that all backup nodes' votes are the same, then they append that information in their own blockchain.

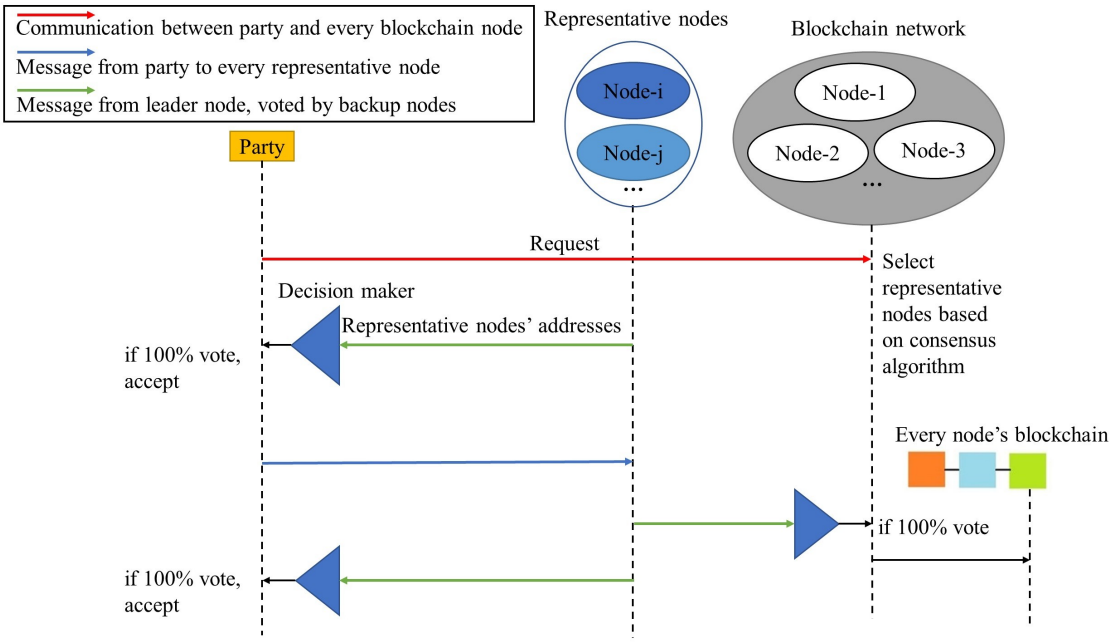

Figure 4.6: Concept of communication between supply chain and blockchain layers.

4.4 PUF

First of all, we have assumed that RFID tag is not removable from the product. RFID tag is usually fragile in nature and will be attached to the product with strong glue. Forcefully trying to remove the tag will significantly damage the product and the tag. Then we considered that PUF is integrated inside the RFID tag. To be precise, RFID tag will correspond to the tag ID and the PUF inside it will be responsible for the CRPs. We have selected Arbiter PUF for our research work, since this one is a strong PUF [43]. The motivation behind using strong PUF is that it will be able to support a larger set of CRPs and thus provide higher security for the product. We have usually used 128 bit CRPs unless we needed to vary the length of the CRPs and 50 CRPs were used for the registration purpose of the tag.

4.5 Secret Sharing

4.5.1 Method of secret sharing

Several secret sharing techniques are in practice now. Some noteworthy secret sharing techniques are: Shamir's secret sharing, Reed-Solomon codes, Linear secret sharing schemes, etc [44] [45]. We can use any of them to keep our system-related sensitive information safe. For simplicity, we have only considered the CRPs which we want to share secretly. Other information such as supply chain party information, product details can also be protected in a similar fashion.

Figure 4.7 shows the basic concept of the secret sharing introduced by Adi Shamir [46].

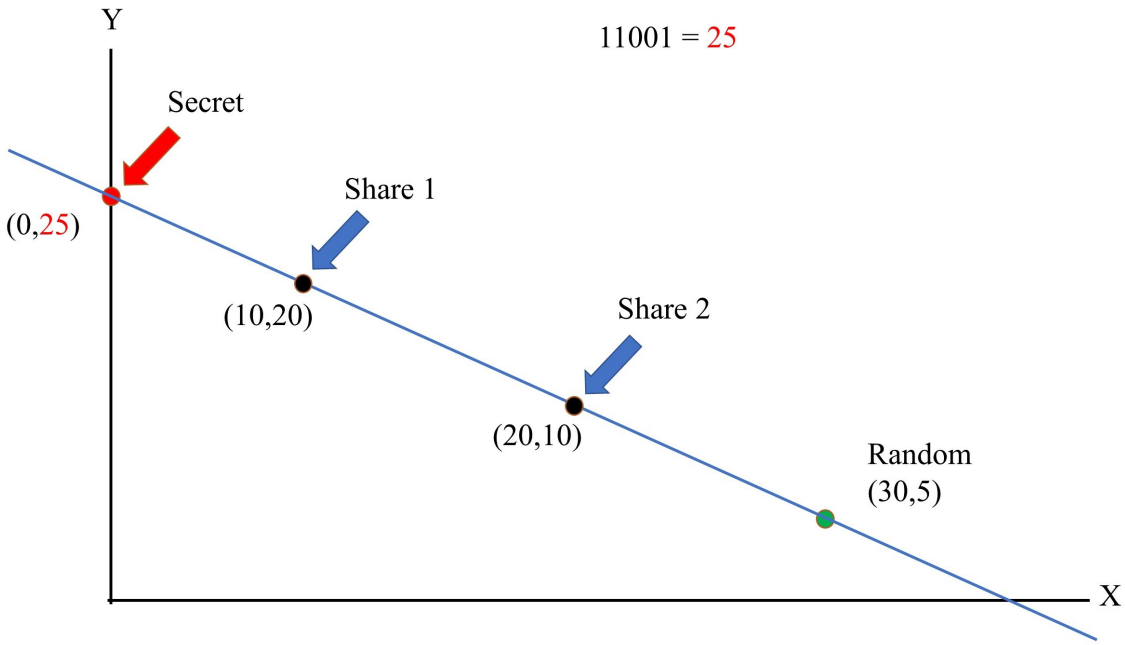

Figure 4.7: Secret sharing concept by Adi Shamir.

Let's assume, we have a challenge whose value is 11001. We want to keep this challenge safe by secret sharing. This challenge value (25 in decimal) will be taken as the Y-axis value as indicated by the red dot. Now, we will take any random point

(for our example, the green dot) and make a straight line by connecting the red dot and the green dot. At this time, if we take any two random points on the straight line (as indicated by the black dots) they will be our shares. Only these two shares will be stored in the blockchain. And whenever we need to construct the secret, we retrieve these two shares, join them, and the intersection with the Y-axis will reveal our secret.

For the illustrated example it is visible that, it requires at least 2 shares to reconstruct the secret. This value is known as the threshold. We can increase the threshold by replacing the straight line with a higher order polynomial curve.

4.5.2 Security of the secret sharing

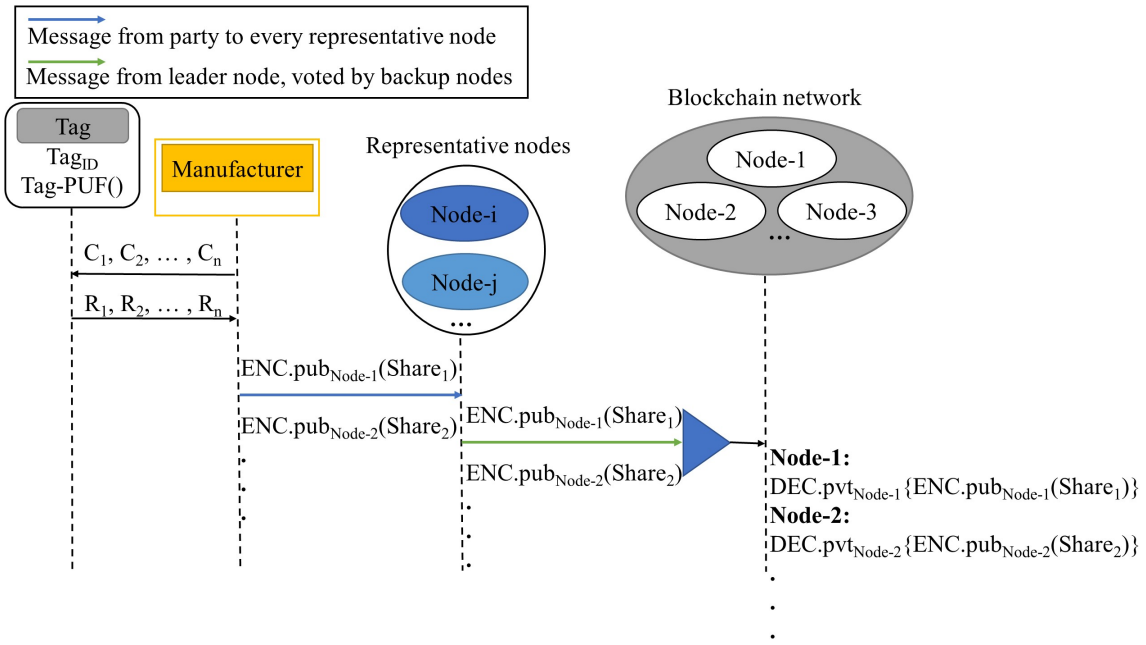

Figure 4.8: Distributing secrets using representative nodes.

Let's assume, C_1 is split into 2 shares $Share_1$ and $Share_2$. Consider that, $Share_1$ will be shared with Node-1 and $Share_2$ will be shared with Node-2. In order to do so, Manufacturer will encrypt the $Share_1$ with the public key of Node-1 and encrypt

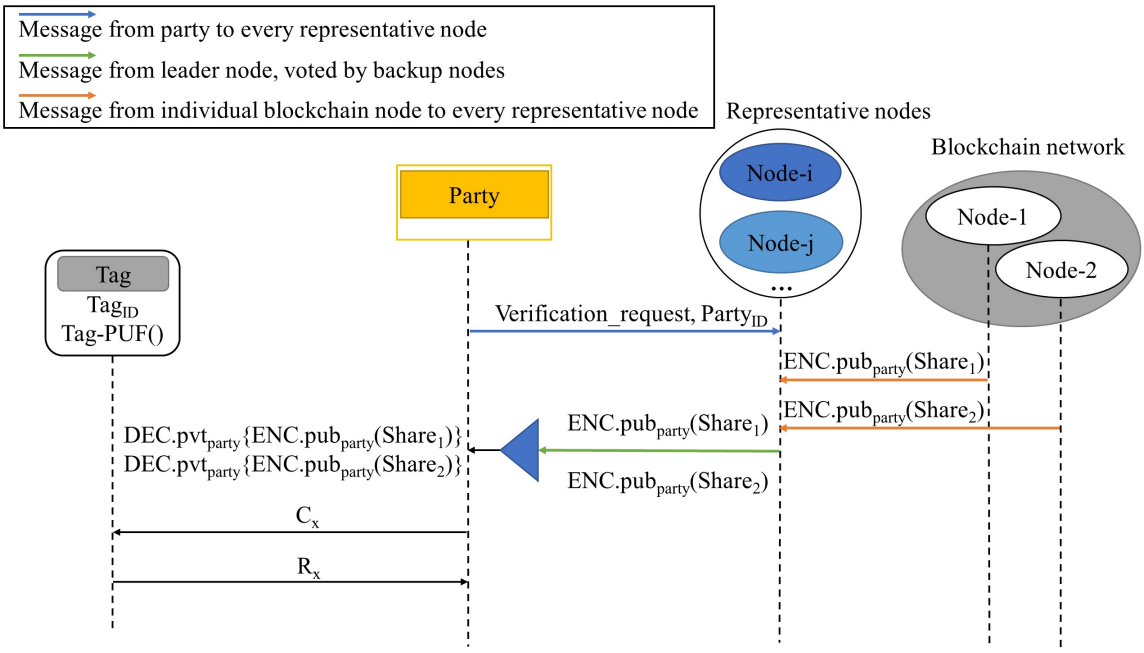

Figure 4.9: Collecting secrets using representative nodes.

$Share_2$ with the public key of Node-2. These encrypted messages will be sent to the representative nodes and the leader node will forward these messages to all the nodes in the blockchain network. Since these messages are encrypted, representative nodes will not be able to have the knowledge of the shares. On the other hand, all the blockchain nodes will receive the encrypted shares. But only the intended node will be able to decrypt it with their corresponding private keys. For instance, Node-1 will be able to get the $Share_1$ as a plain text after decrypting it with Node-1's private key. A similar case will happen for Node-2, when it decrypts the encrypted share with its private key.

Similarly, when any party wants to verify a product, that particular party will send a request to the blockchain network with its corresponding party ID. From that party ID, it will be possible to get that party's public key. So, when Node-1 sends $Share_1$ and Node-2 sends $Share_2$, they encrypt these secrets with the party's public key. The encrypted shares are sent to the representative nodes and the leader node

will forward these encrypted shares to that desired party. That way representative nodes will not have the knowledge of the shares, but the party will be able to decrypt the shares using its private key.

4.6 Protocols

In order to complete the transaction of a valid product, we have devised three protocols, namely: (i) Registration protocol (ii) Verification protocol (iii) Transaction protocol. These protocols are explained with detailed sequence diagrams here. Also, pseudo-codes for each algorithm are provided.

4.6.1 Registration protocol

1. Product Registration Protocol

Every product in the supply chain needs to be registered in the system to have a record of them so that later any registered supply chain party can verify them at any stage of the supply chain. The sequence diagram of the product registration protocol is shown in Figure 4.10.

Algorithm 1 explains the pseudocode for product registration protocol.

2. Party Registration Protocol

Supply chain parties also need to be registered in the system in order to avoid any unauthorized party to extract product information from the system. It can be said that there might be two types of supply chain: open supply chain and predefined supply chain. For the open supply chain, the next supply chain party is not fixed. If any current party thinks that some party is trustworthy and can be considered to do

Algorithm 1 Product registration protocol

1: Manufacturer→Tag: getID()
2: Tag→Manufacturer: Tag_{ID}
3: **for** i=1:n **do**
   Manufacturer→Tag: C_i
   Tag: R_i=Tag-PUF(C_i)
   Tag→Manufacturer: R_i
4: **end for**
5: Manufacturer→Blockchain network: Product_Registration_Request
6: Blockchain network: **do** select representative nodes
7: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Manufacturer: representative nodes' addresses
8: Manufacturer: **do** checks all votes from Backup nodes
9: **if** 100% votes **then**
10: **do** go to next step
11: **else**
   do repeat from step 5
12: **end if**
13: Manufacturer→Representative nodes: SetAsManufacturer(Tag_{ID},$Manu_{ID}$)
14: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Blockchain network: SetAsManufacturer(Tag_{ID},$Manu_{ID}$)
15: Blockchain network: **do** checks all votes from Backup nodes
16: **if** 100% votes **then**
17: **do** write in every node's blockchain
18: **else**
   do repeat from step 6
19: **end if**
20: Manufacturer: **do** assign $CRP_{(1:n)}$ and Product_details to Tag_{ID}
21: Manufacturer: **do** sign (pvt_{Manu}) CRP assignment and Product_details
22: Manufacturer→Representative nodes: signed CRP assignment and Product_details
23: Representative nodes: **do** verify (pub_{Manu}) signed CRP assignment and Product_details
24: **if** TRUE **then**
25: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Blockchain network: signed CRP assignment and Product_details
26: **else**
   do discard message
27: **end if**
28: Blockchain network: **do** repeat steps 15-19

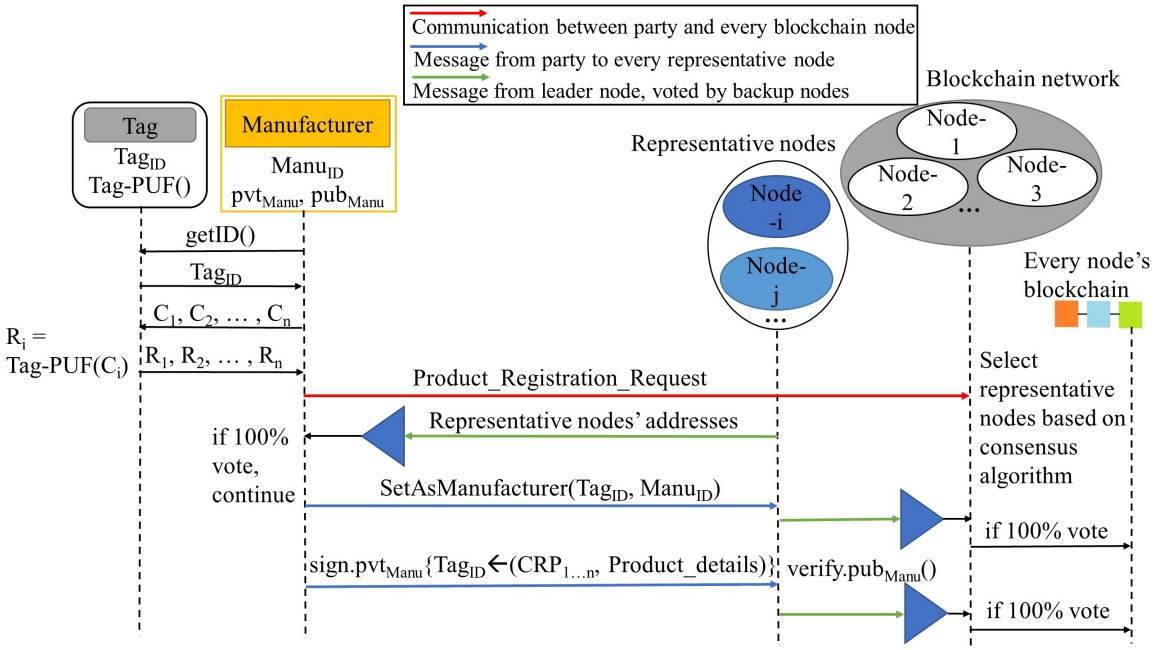

Figure 4.10: Product Registration protocol.

business with, then the current party can register the next party in the system. That approach makes the supply chain more flexible and suitable for the global competitive environment. On the other hand, a predefined supply chain would be that type of supply chain where all the parties throughout the supply chain are selected at the beginning of the supply chain. No new party can enter the supply chain which is not listed. For that type of case also, all the parties need to be registered. But for our research work, we are considering an open supply chain. So, whenever one party wants to verify any product, that party needs to be registered by its previous party. The same rule is applied to product transactions. When any product is being transferred from party A to party B, party B needs to be registered by party A. Figure 4.11 shows the sequence diagram of party registration protocol.

The pseudocode for party registration is shown in Algorithm 2.

Algorithm 2 Party registration protocol

1: Previous party→Blockchain network: Party_Registration_Request
2: Blockchain network: **do** select representative nodes
3: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Manufacturer: representative nodes' addresses
4: Manufacturer: **do** checks all votes from Backup nodes
5: **if** 100% votes **then**
6: **do** go to next step
7: **else**
do repeat from step 1
8: **end if**
9: Previous party: **do** assign $\text{Nextparty}_{\text{ID}}$ to Tag_{ID}
10: Previous party: **do** sign ($\text{pvt}_{\text{Prevparty}}$) party assignment
11: Previous party→Representative nodes: signed party assignment
12: Representative nodes: **do** verify ($\text{pub}_{\text{Prevparty}}$) signed party assignment
13: **if** TRUE **then**
14: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Blockchain network: signed party assignment
15: **else**
do discard message
16: **end if**
17: Blockchain network: **do** compares all messages from Representative nodes
18: **if** 100% votes **then**
19: **do** write in every node's blockchain
20: **else**
do repeats from step 2
21: **end if**

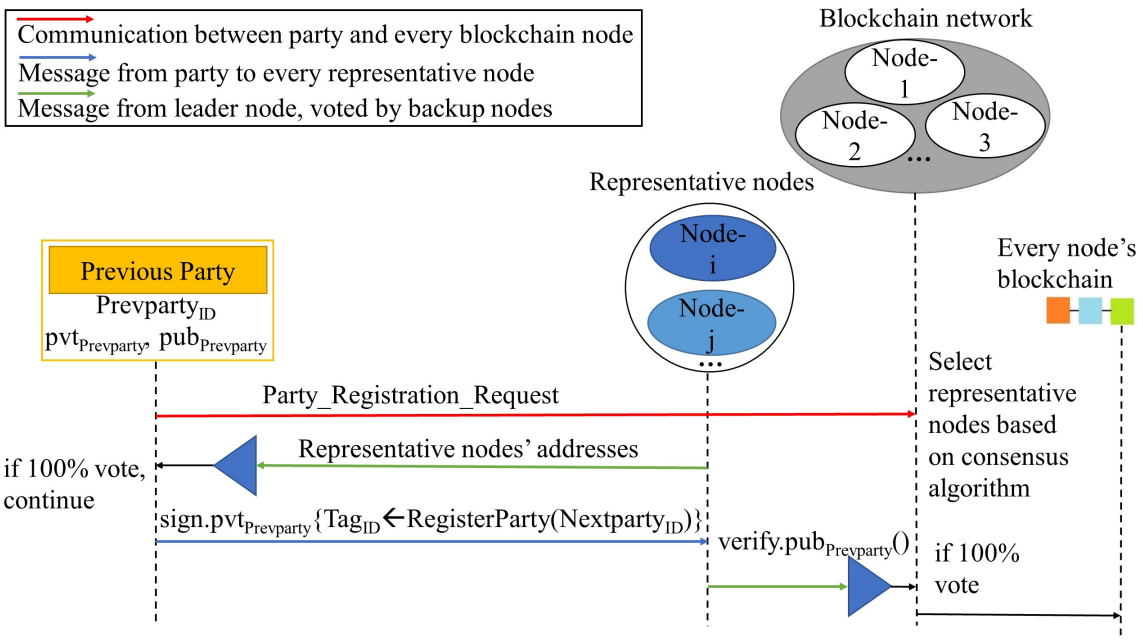

Figure 4.11: Party Registration protocol.

4.6.2 Verification protocol

1. General Verification Protocol

Before the verification protocol, party registration protocol is performed where the previous party registers its next party. This is done to avoid malicious parties to be able to check product information. So, only a registered party in the system can get the CRPs. We have also added an extra layer of security by using the mutual authentication between the tag and the tag reader, which ensures that any party who is not registered in the system cannot verify a product. It is a wise assumption that, the party who is performing verification on a product will be involved in the transaction of the product as well. For this reason, the verification and transaction requests to the blockchain are tied together, which can be separated as well. Another purpose of doing that is to select the representative nodes once instead of twice, which will significantly reduce the energy requirement by running the consensus algorithm

fewer times. The verification protocol is illustrated by a sequence diagram in Figure 4.12 and also illustrated via a pseudocode in Algorithm 3.

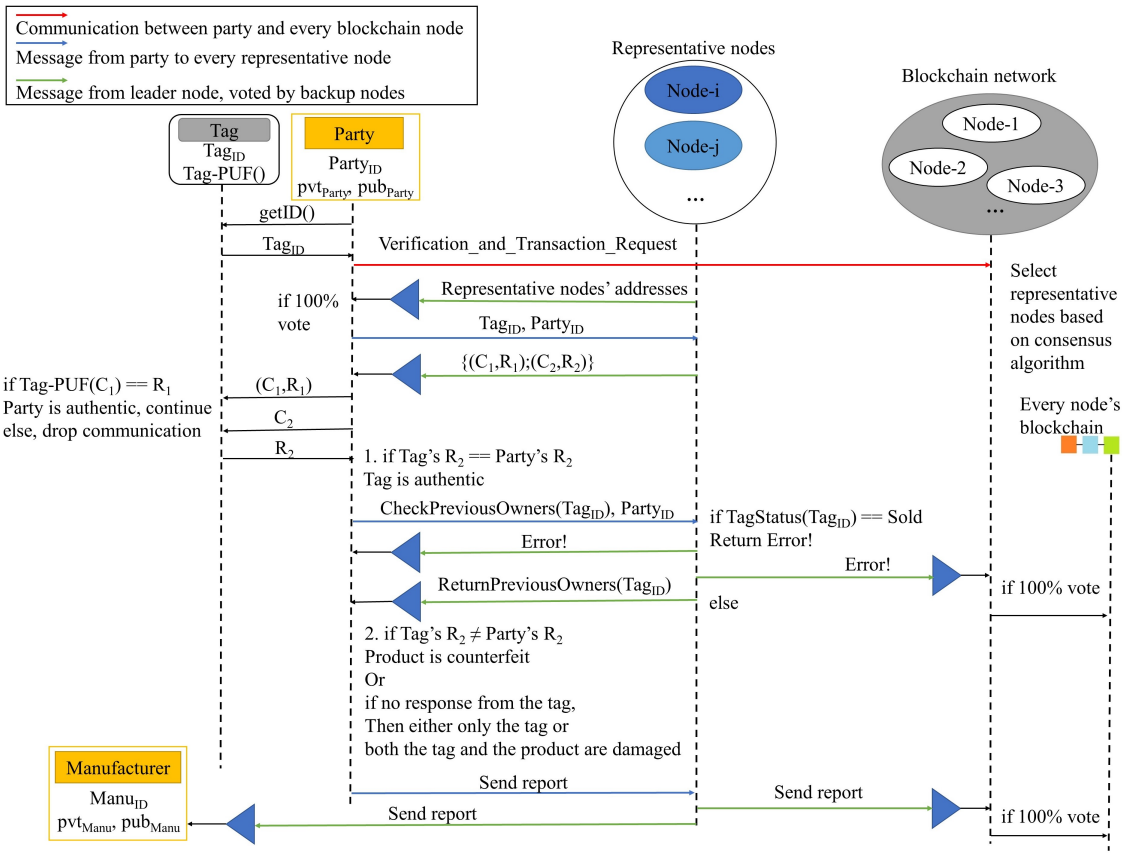

Figure 4.12: Verification protocol.

2. Verification Protocol for End User without RFID Reader

The previous verification protocol was designed for any authentic supply chain party that holds an RFID reader and a smartphone. But this might not always be the case. An end user might also want to verify a product using only a smartphone. Figure 4.13 explains the process of verifying a product without an RFID reader. This special type of verification protocol's algorithms is shown in Algorithm 4.

Now a question may arise that, what is the purpose of introducing two different verification protocol? The motivation behind this is that, the later protocol requires

Algorithm 3 Verification protocol

1: Party→Tag: getID()
2: Tag→Party: Tag_{ID}
3: Party→Blockchain network: Verification_and_Transaction_Request
4: Blockchain network: **do** select representative nodes
5: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Party: representative nodes' addresses
6: Party: **do** checks all votes from Backup nodes
7: **if** *100% votes* **then do** go to next step ;
8: **else do** repeat from step 3 ;
9: **end if**
10: Party→Representative nodes: $\{\text{Tag}_{\text{ID}},\text{Manu}_{\text{ID}}\}$
11: i = 1; // (for the very first round/ stage of supply chain)
12: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Party: $\{(C_i,R_i);(C_{i+1},R_{i+1})\}$
13: Party: **do** repeat steps 6-9
14: Party→Tag: (C_i,R_i)
15: Tag: **do** Tag-PUF(C_i)
16: **if** *Tag-PUF(C_i) == R_i* **then** Party is authentic, **do** go to next step ;
17: **else do** drop communication ;
18: **end if**
19: Party→Tag: C_{i+1}
20: Tag→Party: R_{i+1}
21: **if** Tag's R_{i+1} == Representative node's R_{i+1} **then**
22: Tag is authentic, **do** go to next step
23: Party→Representative nodes: $\{\text{CheckPreviousOwners}(\text{Tag}_{\text{ID}}),\text{Party}_{\text{ID}}\}$
24: **if** TagStatus(Tag_{ID}) == Sold **then**
25: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Party: Error!
26: Party **do** checks all votes from Backup nodes
27: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Blockchain network: Error!
28: Blockchain network: **do** checks all votes from Backup nodes
29: **if** *100% votes* **then do** write in every node's blockchain ;
30: **else do** repeat from step 4 ;
31: **end if**
32: **else**
Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Party: $\text{ReturnPreviousOwners}(\text{Tag}_{\text{ID}})$
33: **end if**
34: Party: **do** repeat steps 6-9
35: **else if** Tag's $R_{i+1}\neq$Representative node's R_{i+1} **then**
product is counterfeit
36: **else**
tag is damaged; // (no response from Tag)
37: **end if**
38: **if** (step 36 or step 37 TRUE) **then**
39: Party→Representative nodes: report
40: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Blockchain network: report
41: Blockchain network: **do** repeat step 29-31
42: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Manufacturer: report; // (Optional)
43: **end if**
44: i = i+2;

Algorithm 4 Verification protocol for end user/buyer without RFID reader

Seller→Tag: getID(); // (Optional)
Tag→Seller: Tag_{ID}; // (Optional)
Seller→Blockchain network: Send_Product_Details_Request
Blockchain network: **do** select representative nodes
Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Seller: representative nodes' addresses
Seller: **do** checks all votes from Backup nodes
if *100% votes* **then do** go to next step ;
else do repeat from step 3 ;
end if
Seller→Representative nodes: {Tag_{ID},$Seller_{ID}$}
i = i; // (value from previous stage. considering there exists at least one party between Manufacturer and Buyer)
Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Party: C_i
Seller: **do** checks all votes from Backup nodes
if *100% votes* **then do** go to next step ;
else do repeat from step 3 ;
end if
Seller→Tag: C_i
Tag: **do** R_i = Tag-PUF(C_i)
Tag→Party: R_i
Seller: **do** message = [send_product_details({C_i,R_i},Tag_{ID}) to $Buyer_{ID}$]
Seller: **do** sign (pvt_{Seller}) message
Seller→Representative nodes: signed message
Representative nodes: **do** verify (pub_{Seller}) signed message
if TagStatus(Tag_{ID}) == Sold **then**
    Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Buyer: Error!
    Buyer: **do** checks all votes from Backup nodes
    Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Blockchain network: Error!
    Blockchain network: **do** checks all votes from Backup nodes
    if *100% votes* **then do** write in every node's blockchain ;
    else do repeats from step 4 ;
    end if
else
    Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Buyer: ReturnPreviousOwners(Tag_{ID})
end if
Buyer: **do** checks all votes from Backup nodes

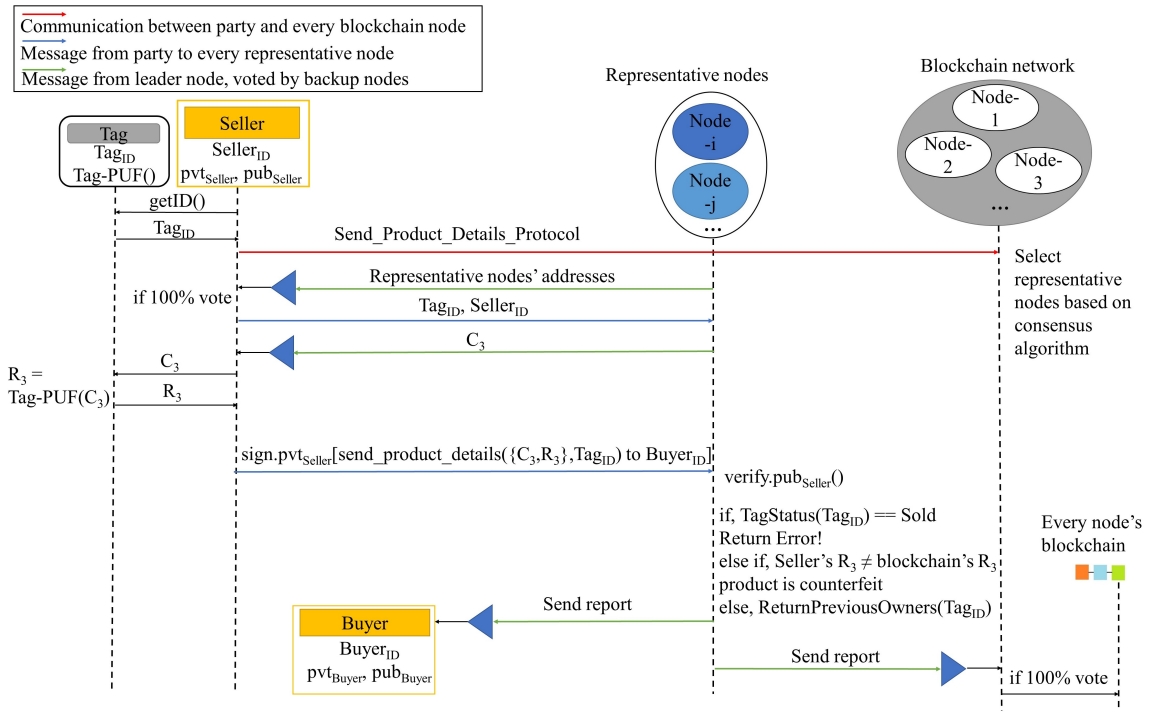

Figure 4.13: Verification protocol for end user without RFID reader.

the involvement of the previous party in the verification process, which might not be desired by supply chain parties who own an RFID reader and a smartphone. They may just prefer to verify the product by themselves.

4.6.3 Transaction protocol

Transaction protocol is executed to transfer the ownership of the product. This protocol also gives each transaction a unique transaction ID (along with a timestamp and geographic location) to each transaction which makes every transaction identifiable. Also, when a product finishes its journey throughout the supply chain, the product is listed as sold in the system to make the supply chain a linear one. Figure 4.14 shows the step-by-step processes required for the transaction protocol in detail. The pseudocode for the Transaction protocol is shown in Algorithm 5.

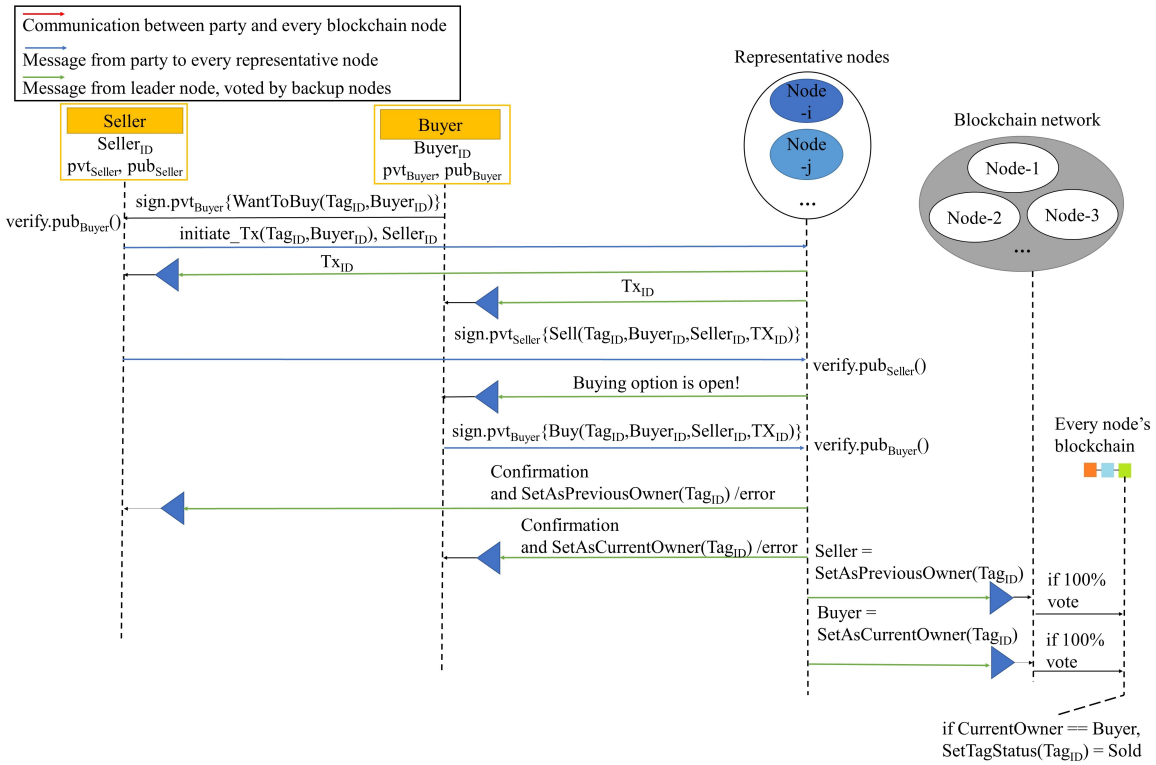

Figure 4.14: Transaction protocol.

Algorithm 5 Transaction protocol

1: Buyer: **do** sign (pvt_{Buyer}) message1 [=WantToBuy(Tag_{ID},$Buyer_{ID}$)]
2: Buyer→Seller: signed message1
3: Seller: **do** verify (pub_{Buyer}) signed message1
4: **if** *TRUE* **then** **do** go to next step ;
5: **end if**
6: Seller→Representative nodes: message2={initiate_Tx(Tag_{ID},$Buyer_{ID}$),$Seller_{ID}$}
7: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Seller: Tx_{ID}
8: Seller: **do** checks all votes from Backup nodes
9: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Buyer: Tx_{ID}
10: Buyer: **do** checks all votes from Backup nodes
11: Seller: **do** sign (pvt_{Seller}) message3 [=Sell(Tag_{ID},$Buyer_{ID}$,$Seller_{ID}$,Tx_{ID})]
12: Seller→Representative nodes: signed message3
13: Representative nodes: **do** verify (pub_{Seller}) signed message3
14: **if** *TRUE* **then** Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Buyer: Buying option is open! ;
15: **end if**
16: Buyer: **do** sign (pvt_{Buyer}) message4 [= Buy(Tag_{ID},$Buyer_{ID}$,$Seller_{ID}$,Tx_{ID})]
17: Buyer→Representative nodes: signed message4
18: Representative nodes: **do** verify (pub_{Buyer}) signed message4
19: **if** TRUE **then**
20: Leader node$\xrightarrow{\text{voted by Backup nodes}}$Seller:
confirmation and SetAsPreviousOwner(Tag_{ID})
21: Seller: **do** checks all votes from Backup nodes
22: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Buyer:
confirmation and SetAsCurrentOwner(Tag_{ID})
23: Buyer: **do** checks all votes from Backup nodes
24: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Blockchain network:
Seller = SetAsPreviousOwner(Tag_{ID})
25: Blockchain network: **do** checks all votes from Backup nodes
26: **if** *100% votes* **then** **do** write in every node's blockchain ;
27: **else** **do** modifies representative nodes selection // (not shown) ;
28: **end if**
29: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Blockchain network:
Buyer = SetAsCurrentOwner(Tag_{ID})
30: Blockchain network: **do** repeats steps 25-28
31: **else**
32: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Seller: Error !
33: Seller: **do** checks all votes from Backup nodes
34: Leader node $\xrightarrow{\text{voted by Backup nodes}}$ Buyer: Error !
35: Buyer: **do** checks all votes from Backup nodes
36: **end if**
37: Blockchain network:
38: **if** CurrentOwner == Buyer **then**
39: **do** SetTagStatus(Tag_{ID}) = Sold
40: **end if**

This chapter provides a complete description of the system architecture and step-by-step illustrations of the protocols. These details; and the concepts of our supply chain, PUF, and secret sharing are necessary as we move forward to the Security and Privacy Analysis of the System, and the Result Analysis chapters.

Chapter 5

Security and Privacy Analysis of the System

In this chapter, based on the system architecture and the proposed protocols, the security and privacy analysis of our system is performed. How our proposed system can successfully counter the attacks mentioned in Chapter 3 (Threat Model) is discussed here with detailed explanation.

Defense against 'Counterfeiting attack': Dishonest supply chain parties may try to sell counterfeit products with fake tags. For this attack, during the Verification protocol (Algorithm 3, step 21), since the tag's generated response will not be similar to the response collected from the representative nodes, the tag will be failed to prove itself as authentic and product will be considered as a fake one. But, the dishonest supply chain parties may also try to remove an original tag from an authentic product and attach it with a fake product to make the fake product pass as a genuine one. However, this attack is not very easy to pull off. The explanation behind this is that: to get an authentic tag, an internal adversary will have to buy an original product

since the RFID tag is strongly attached to the product and it is not cloneable due to the PUF (that means copies of the authentic tag are not available). Then they will have to make an exact counterfeit product, where they are supposed to attach the authentic tag. So, that way they are spending more than they are probably going to benefit from the fraudulent activity.

In a similar fashion, if an external adversary gets an original product with an authentic tag in hand, they may try to detach the authentic tag and attach it with a fake product. Then they can attempt to sell the fake product with an authentic tag outside of the supply chain claiming that the product is a real one. In addition to the countermeasures mentioned for internal adversaries, this attack can be handled with the help of a unique secret code that will be assigned to a particular product. The tag ID will correspond to original product attributes (including product image, manufacturing details, and a secret code of the product) which will be stored in the blockchain and not be accessible by external adversaries. So in order to become successful, the adversary will have to remove the tag from the product forcefully, create an exact counterfeit, and break-in into the blockchain to get that secret code.

Defense against 'Product double-spending attack': Similar to the double-spending attack in cryptocurrency, fraud supply chain parties may attempt to double-spend the physical products as well. That means they can try to sell fake products using the information of already sold genuine products. This attack can be performed by physically cloning the RFID tag or by simulating the tag properties using a programmable device. For the former attack, since we have used PUF inside RFID, original RFID tags cannot be cloned due to the OUF properties. Also, if the adversaries desire to clone the original tag using some sort of programmable RFID tag, they will not be able to do so because one supply chain party is provided with two CRPs only (one CRP for party authentication and another CRP for product authen-

tication) for the verification purpose. Only two CRPs are not enough to mimic a tag that will pass all the stages of the supply chain verification.

However, if an adversary manages to clone the original tag somehow, they will not be able to use a single tag multiple times. The defense mechanism against this attack is as follows: selling the product for the first time will not be an issue. But when the adversary will attempt to sell another product with the same tag, during the Verification protocol (Algorithm 3, step 24) it will be revealed that the product was already sold. This filtering will be possible to perform because in the Transaction protocol (Algorithm 5, step 38) the tag has already been marked as sold when the first time the product crossed the whole supply chain.

Defense against 'Malicious and fake representative nodes': Selected representative nodes might work maliciously, as there is no way of knowing whether the representative nodes will act honestly or maliciously after selection. Selected representative nodes' malicious activity include: false message sent by the representative nodes' leader node, and backup nodes give false votes on leader node's massage. For the first type of attack, the leader node's message is voted by backup nodes, and a message from the leader node is accepted only when it contains 100% identical backup votes. So, as long as there is at least one honest backup node, no false message from the leader node will be accepted. For the second type of attack, backup nodes can group up and vote maliciously. But, even a single different vote will cause the consensus algorithm to run one more round and the algorithm will search for a full set of honest representative nodes. This approach works pretty well since the probability of all backup nodes voting maliciously is very less as shown in Figure 4.9.

In the case of fake representative nodes, if such an attack happens, the supply chain party will at first believe the malicious nodes and start communicating with them. But when the malicious nodes will try to append anything in the blockchain,

the blockchain network will reject that message, as they have the knowledge of who are the actual representative nodes. And thus after the revealing of such adversarial activity, the reputation of the malicious nodes will be decreased.

Defense against 'Secret share manipulation': After getting the secret (i.e. CRP), the manufacturer splits this secret into multiple shares and distributes these shares among several blockchain nodes to avoid data leakage. Then, when the shares are being collected from blockchain nodes, malicious blockchain nodes can change their shares and send the manipulated shares. This issue can be resolved by increasing the number of shares for a particular secret. Figure 6.8 represents with the increased number of shares, what is the maximum number of manipulated shares that can be handled. If it is predicted that malicious blockchain nodes might manipulate their shares, then an increased number of shares are recommended to reconstruct the secret.

From the above-mentioned countermeasures, it is understandable that our system architecture and protocols along with the blockchain and PUF technology are robust enough to handle the possible attacks on our system. After reading the details about the proposed system, a question might arise that: why do we require blockchain? Why not any regular distributed storage system is sufficient enough for the system? The answers to this question are as follows:

1. Our proposed Registration and Verification protocols might just work fine with the help of a distributed storage. But for the Transaction protocol, transactions are not encrypted (which was done intentionally to avoid high power-required encryption at the lightweight supply chain party's end). So, adversaries can attack the distributed system and try to modify the transaction history. In order to handle that, blockchain is used to make the transactions immutable by using hash as a pointer to the previous block.

2. An honest link is required between the supply chain party and the blockchain

network (or the distributed system), which will work as a carrier between them to convey their messages to each other. Direct communication between the supply chain party and the blockchain (or the distributed system) is not preferable due to high communication overhead. This honest link was chosen using the blockchain consensus algorithm. Choosing some representatives based on consensus (e.g. reputation) is not possible in a distributed storage system since a consensus is absent there, and also randomly selected nodes might get compromised.

3. In our proposed system, some decisions need to be made or works need to be done at particular stages, such as product ownership handover, signature verification, condition checking, and so on. Someone needs to be dedicated (i.e. Representative nodes for our case) to achieve that purpose. Which is difficult to perform with a distributed storage system only.

The above-mentioned points validate that, even though a distributed storage might have been a simpler choice, blockchain technology is definitely the best possible option for our intended research purpose.

Chapter 6

Result Analysis

Time and power are the two vital parameters to measure the performance of any system. The response time of our proposed system should be less since any unnecessary delay is not accepted in the real-world supply chain. On the other hand, power consumption must be also minimum considering the lightweight supply chain devices. However, since we have performed a simulation-based analysis of our system, we have only considered the timing analysis of our proposed system, and power consumption analysis is left for future work. In this chapter, we have shown how the time requirement for our system varies with respect to consensus, CRP length, number of transactions per block, number of supply chain parties, and number of secret shares. In order to do that, we have used the same simulation environment for every data generation. The simulation environment was as follows: the computer used to perform the simulation was equipped with an Intel Core-i7, 3 GHz processor; 16 GB DDR4 Ram; Windows 10, 64-bit operating system. Python 3.8.2 was used to simulate our proposed system.

6.1 Timing analysis of the consensus algorithm

Here, we have compared our consensus with the most popular consensus algorithm Proof-of-Work (PoW) in terms of the time requirement. To assess how much time PoW may require to reach consensus, we have used the algorithm as described in Algorithm 6. The simulation environment was the same as for our proposed system (as mentioned earlier). Here we have set the miner's CPU power or hashing capability by the maximum possible number of nonce he/she can try, which is 2^{32} for our simulation. Then, within the miner's power, the miner will try to find out whether he/she can achieve the desired target. The difficulty bits are chosen between 16 and 24 inclusive. Since, for difficulty bits below 16 bits, the consensus is reached too quickly; and above 24 bits, it requires a huge amount of time to reach consensus, these values are not shown in the graph.

Algorithm 6 Algorithm for setting PoW difficulty

```
1: for difficulty_bits=16:24 do
2:    target = 2^(256-difficulty_bits)
3:    for nonce=1:2^32 do
4:       hash_result = Hash(message + nonce)
5:       if hash_result <target then
6:          Success
7:       else
8:          Failure
9:       end if
10:   end for
11: end for
```

Figure 6.1 explains how with the increased number of difficulty bits, the time required to reach consensus increases. The same simulations were performed in the same environment with 20 different messages for every difficulty bit. The graph represents the average required times with standard deviations from the 20 messages. The messages used here for the simulation purpose are the actual messages we have used for our proposed protocols.

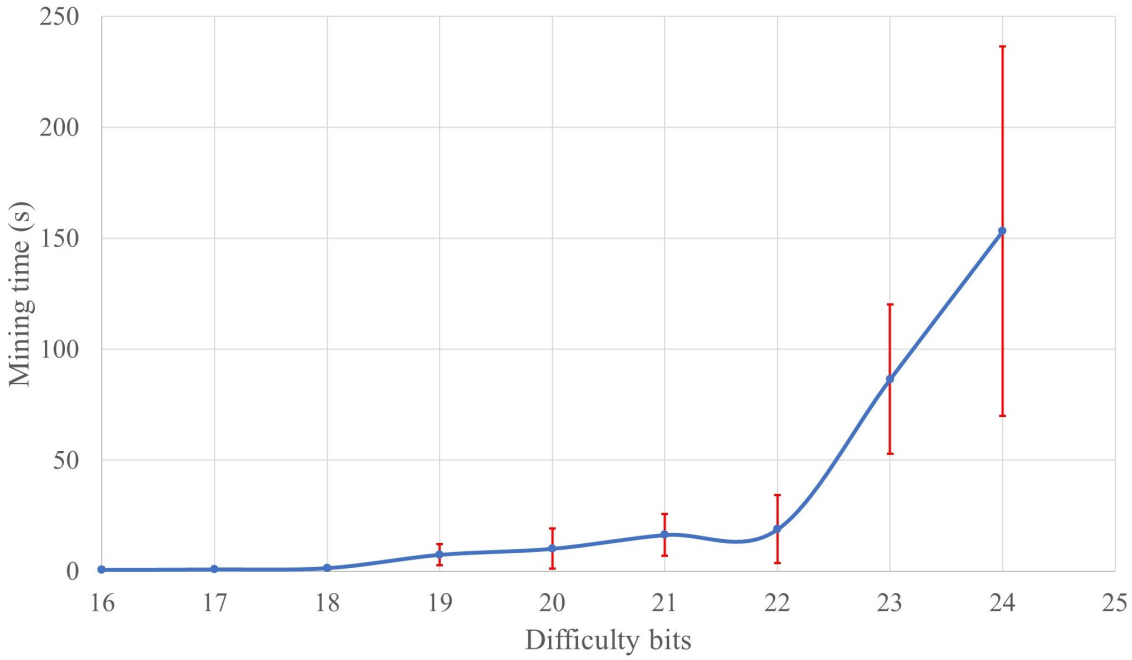

Figure 6.1: Different difficulty bits vs mining time for PoW. Graph represents average and standard deviation from 20 different messages.

It can be said from the graph that, if the difficulty bits are small, then it will require less time to mine and eventually less power. So, an adversary with sufficient mining capability may try to append false messages in the blockchain by manipulating previous blocks. On the other hand, if it is attempted to fight against adversaries by increasing the difficulty bits, it will take a lot of time (e.g. more than 2.5 minutes on average for 24 bits) which is inconvenient for fast response required supply chain transactions.

Our proposed consensus is a reputation-based one, which selects a group of representative nodes from the total nodes based on their (highest) reputation. The whole consensus algorithm is restarted after throwing out the malicious node(s) and taking in the same amount of highest reputed node(s) (from the remaining nodes of the total nodes after taking the representative nodes) in the representative nodes group if any malicious nodes were found in the primarily selected representative nodes. Figure 6.2

shows a graph considering that there are total 100 nodes present in the network. It can be observed from the graph that, if there were no malicious nodes in the initial representative nodes, then the consensus would have reached in the very first stage, and the algorithm does not need to go to the second round. For that case, the time requirement is very trivial. Taking into consideration that, to reach consensus, the algorithm might require to go through several rounds, this figure illustrates how time requirement might change with the number of required rounds. So, for total 100 nodes with 50% malicious nodes, for the worst case scenario, we might have to go through 50 rounds (1 malicious node in each round). It can be said that, even for the worst case scenario (for 100 total nodes) consensus can be reached within 0.18 seconds, which is pretty fast.

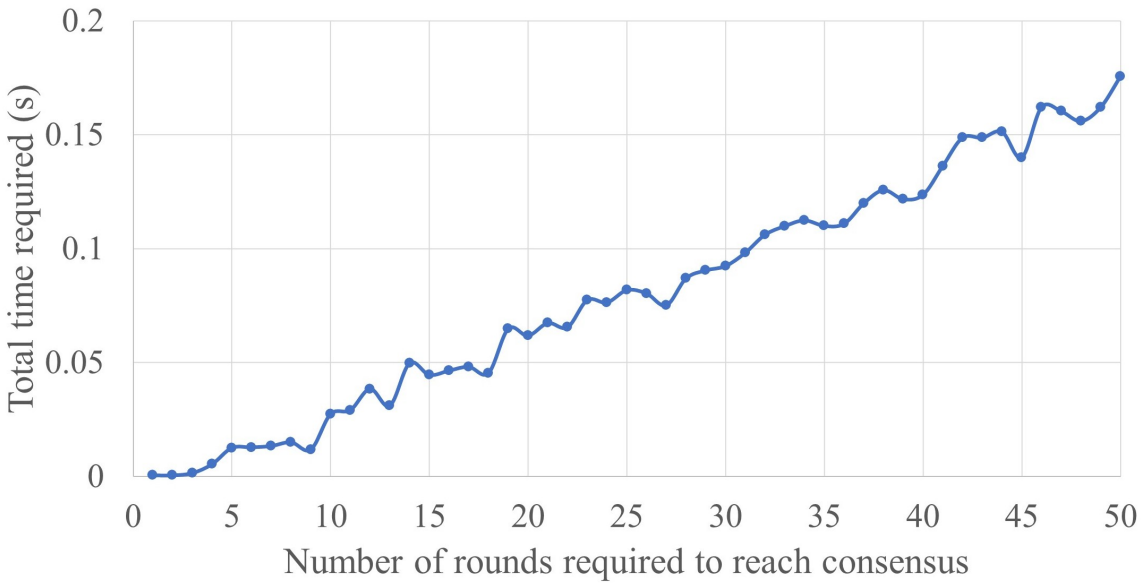

Figure 6.2: Time required for different rounds for the proposed consensus considering total 100 nodes.

In a similar fashion, for our proposed consensus algorithm, Figure 6.3 shows the worst case time requirements for 100, 1000, 1500, 2000, and 2500 total nodes considering 10%, 20%, 30%, 40%, and 50% malicious nodes for each of them.

Next, we have changed the total number of blockchain nodes in the system. For

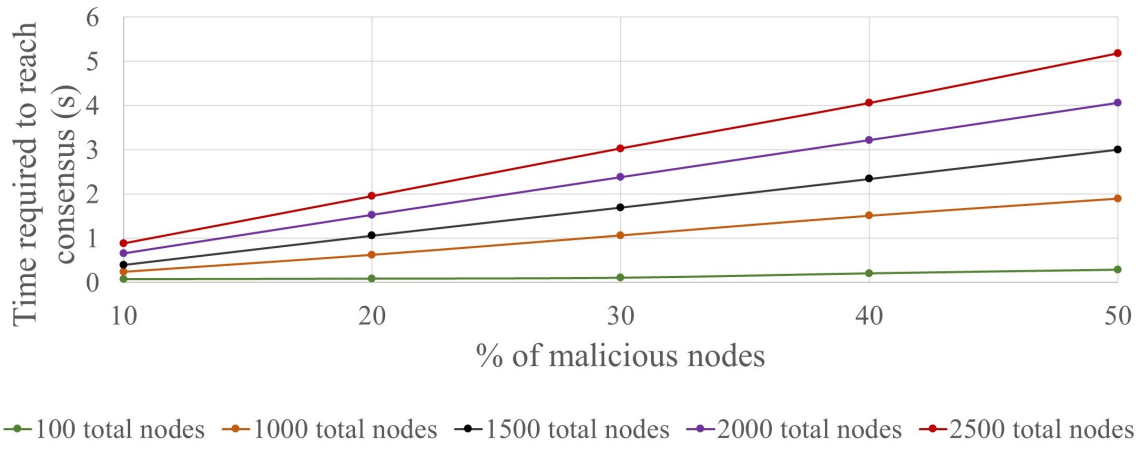

Figure 6.3: Worst case scenario required times for the proposed consensus for different percentage of malicious nodes. Results shown for 100, 1000, 1500, 2000, and 2500 total nodes.

each total number of blockchain nodes, we have shown the time required to reach consensus for 10%, 20%, 30%, 40%, and 50% malicious nodes. This is illustrated in Figure 6.4

From the graphs represented in Figure 6.3 and Figure 6.4, it can be said that, if we have 50% malicious nodes present in a total of 2500 nodes, the time required to reach the consensus will not exceed 5 seconds, whereas consensus can be achieved even faster if the selection process gets lucky.

By comparing the above simulation results, it can be clearly said that, the proposed consensus algorithm is quite faster than PoW.

In terms of security, if the malicious nodes in the PoW network have more than 50% computational power, they have the capability of controlling the network [47]. On the other hand, for our proposed consensus's case, we can fight against any percentage of malicious nodes by modifying the representative node group's size as long as there is a single honest node in the network, considering that the probability of all representative nodes being malicious at the same time will be close to zero.

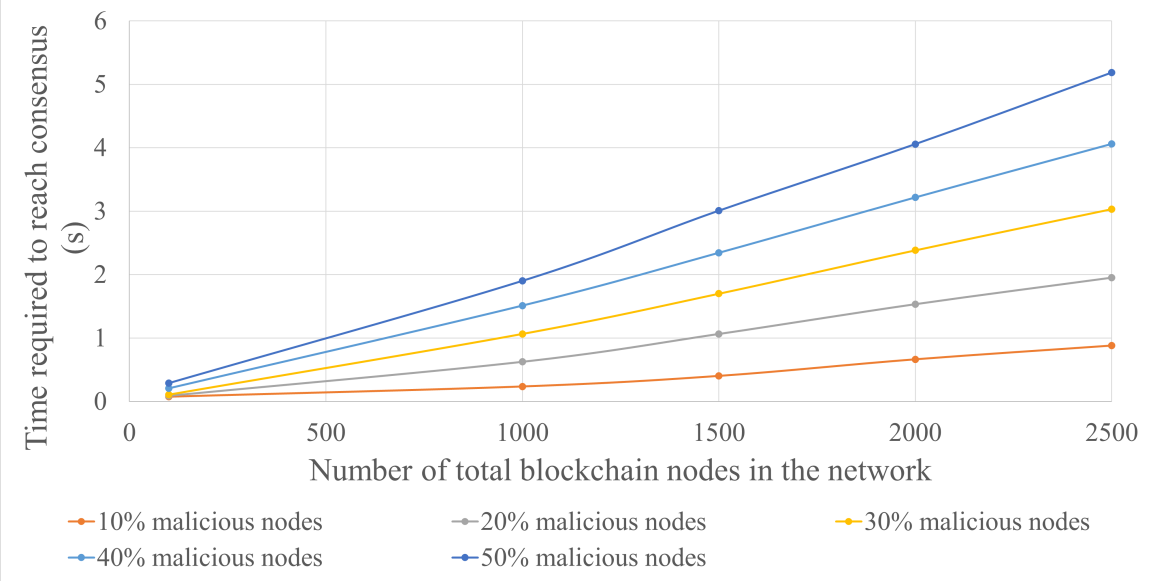

Figure 6.4: Worst case scenario time requirement of the proposed consensus for different number of total nodes. Results shown for 10%, 20%, 30%, 40%, and 50% malicious nodes

6.2 Effect of the CRP length

In this section, we have analyzed how the length of CRP affects the transaction time. We have considered the CRPs of different lengths (16, 32, 64, 128, 256, and 512 bits) and measured the corresponding time for one handover of the product (not the entire supply chain), which includes product registration, party registration, product verification, and transaction. We have used 50 CRPs for the product registration and considered only one transaction per block. The simulations were performed for 100, 1000, and 2000 total blockchain nodes, considering 50% malicious nodes for each of them.

From Figure 6.5, it is visible that time requirement increases almost exponentially with the increased length of CRPs. This happens since CRP generation time increases with the increased length (as stated in Table 6.1), and we need to generate CRPs for multiple protocols (i.e. product registration and verification).

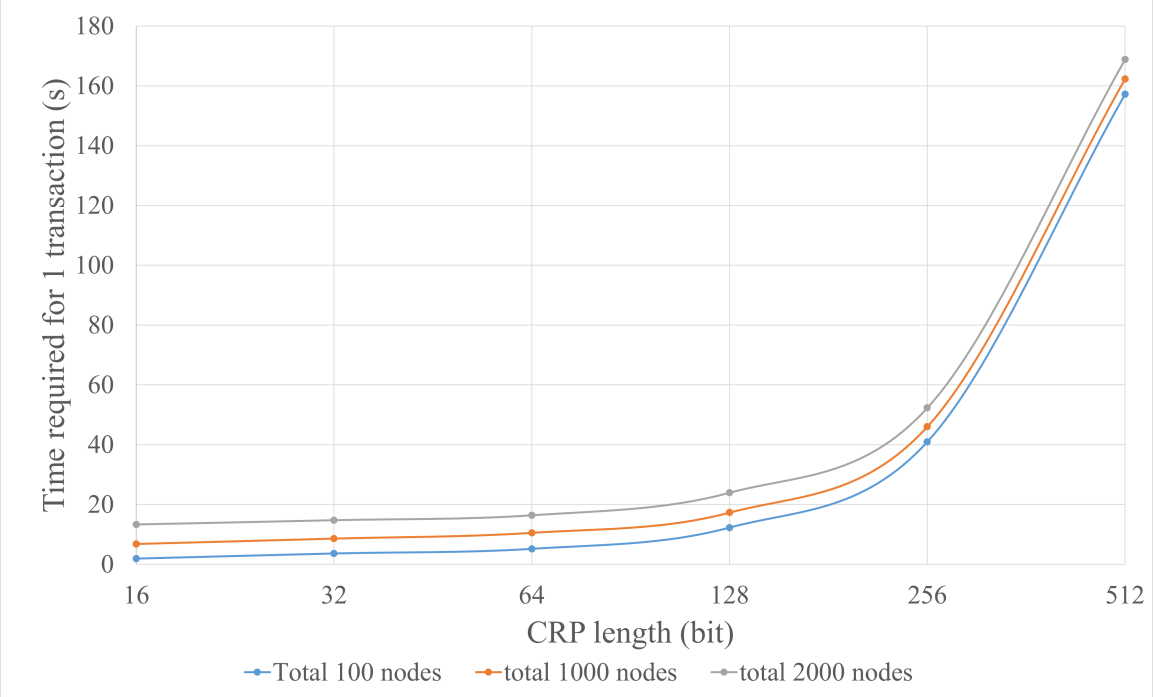

Figure 6.5: One ownership transfer (including registration, verification and transaction) time requirement for different CRP lengths.

6.3 Effect of the number of transactions per block

To observe the effect of the 'number of transactions in a block' in transaction time, we have incorporated multiple transactions (1 to 10) in one block and calculated the subsequent time required for one handover of the product (including product registration, party registration, product verification, and transaction). We have used 50 CRPs of 128 bit length for the product registration. Results are shown for total 100, 1000, and 2000 total blockchain nodes (each having 50% malicious nodes) in Figure 6.6.

The Figure illustrates that, increase in the time requirement is very trivial for the increased number of transactions per block. The reason behind this is that,

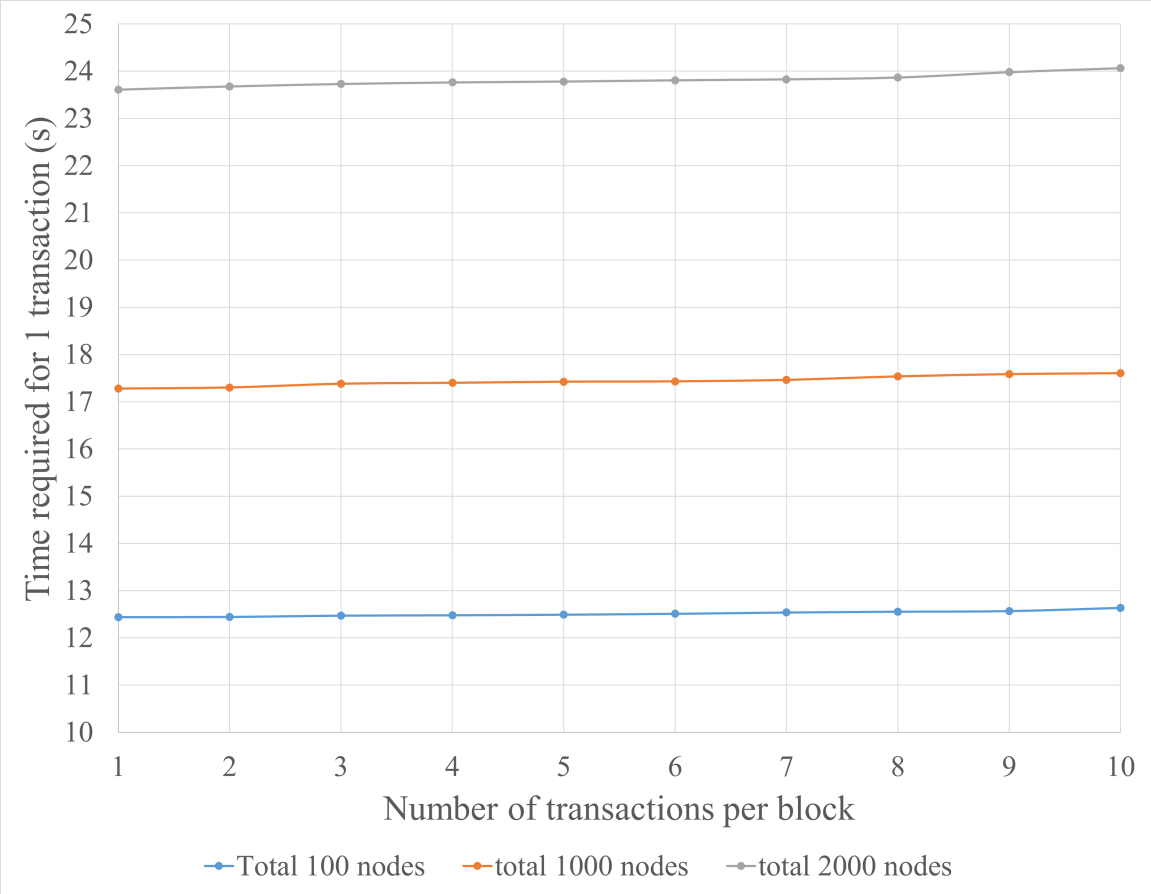

Figure 6.6: One ownership transfer (including registration, verification and transaction) time requirement for different number of transactions per block.

CRP length (bit)	Time required to generate 1 CRP (s)
16	0.001
32	0.0156216621398925
64	0.0533690452575683
128	0.195951223373413
256	0.742831707000732
512	2.98059892654418

Table 6.1: CRP generation time for different bit length.

irrespective of the block size, we are just signing and verifying the message inside the block once to send it (Algorithm 1, Step 21 and 23); and hashing it once to create the blockchain (Algorithm 1, Step 17). Simulation results reveal that, signing and hashing techniques used here do not consume more time depending on the data size, whereas time consumption depends on how many times signing and hashing are done.

6.4 Effect of the number of supply chain parties

In this section, we have changed the total number of supply chain parties (5, 10, 15, 20, and 25) and calculated the corresponding time required for a product to travel from the beginning to the end of the supply chain. Each simulation was performed for total 100, 1000, and 2000 total blockchain nodes with 50% malicious nodes. Fifty 128 bit CRPs were used for the product registration purpose and each block contained one transaction.

Figure 6.7 shows that, required time increases linearly with the increased number of supply chain parties. It is quite obvious from our protocols that, as we increase the number of supply chain parties, only the product registration is being performed once; but the party registration, product verification, and transaction protocols are just being repeated multiple times.

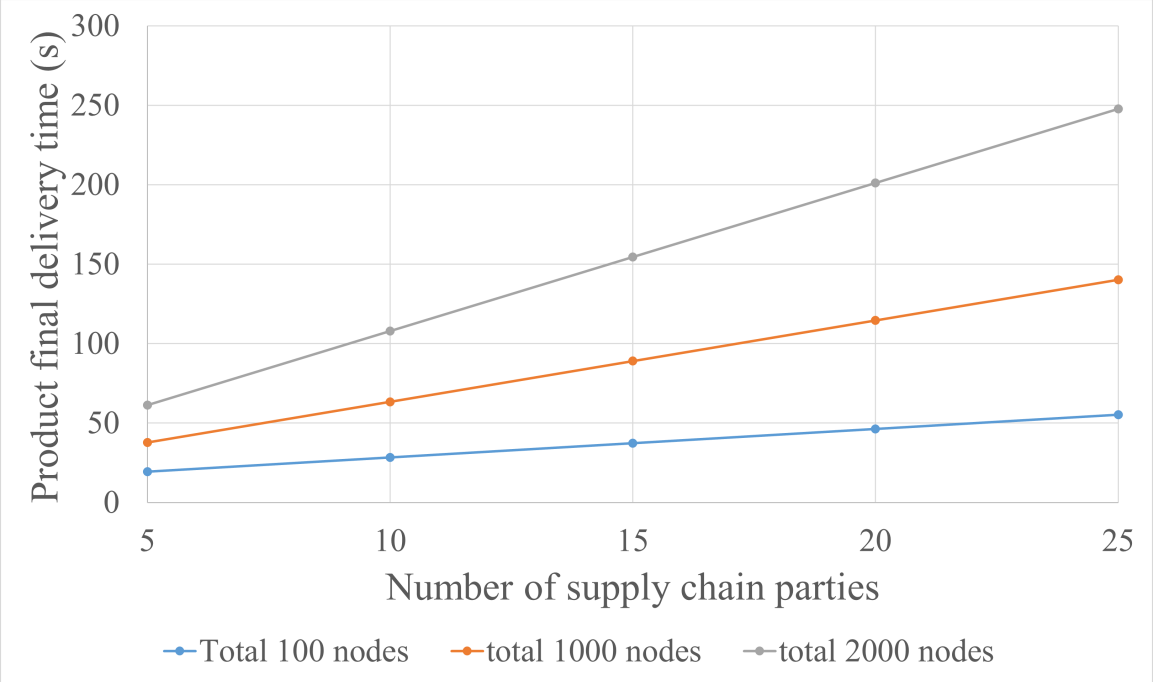

Figure 6.7: Product total travelling time from the beginning to the end of the supply chain for different number of supply chain parties.

6.5 Effect of the number of secret shares

This section gives a generalized idea of how splitting a secret into multiple shares may affect the time requirement for distributing and collecting shares. A brief idea is also given regarding the scenario that: if there is any possibility of blockchain nodes manipulating their own share, how by increasing the number of shares this attack can be handled. To do so, we have considered a 128 bit challenge as a secret. Then we have split this secret into 2, 4, 8, and 16 shares. In the primary axis of Figure 6.8, we have indicated how many maximum numbers of share manipulation can be handled with each number of shares. In the secondary axis, the related time requirement for distributing and collecting secret shares for each number of shares are mentioned.

Now, while collecting the shares, if we have any manipulated shares; considering the majority shares as authentic, it can be said that: if a secret is distributed among n shares, the maximum number of share manipulation can be handled is less than

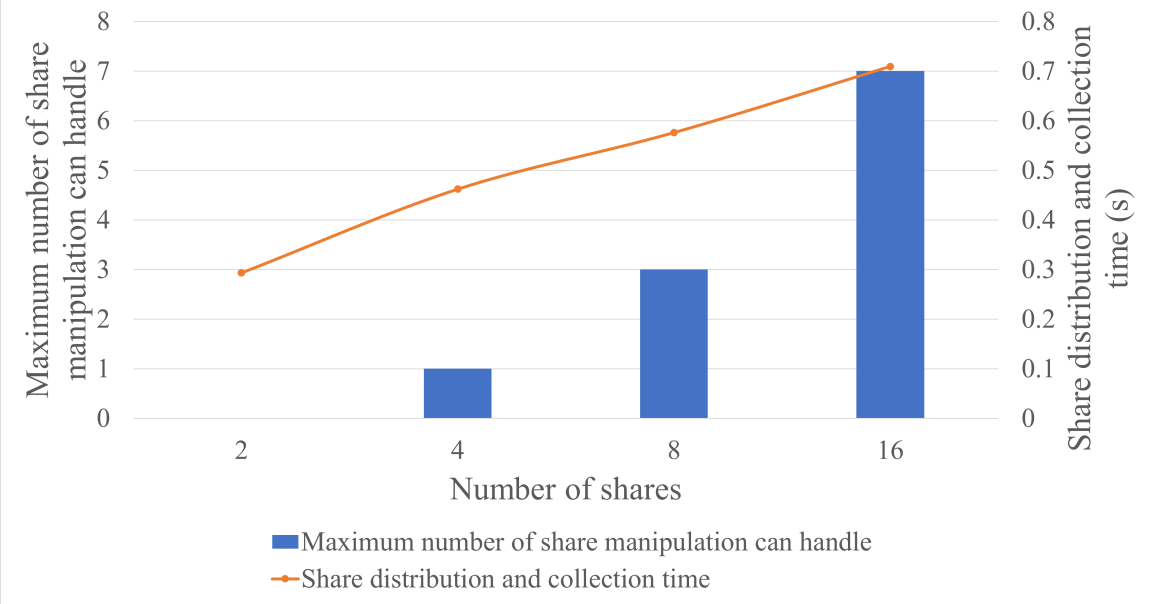

Figure 6.8: Time requirement for distributing and collecting secrets, and the number of maximum share manipulations can be handled for different number of shares .

half of the total shares, mathematically *ceiling(n/2)-1* shares. This is graphically illustrated in the primary axis of Figure 6.8.

The secondary axis of the Figure describes that, the time required for distributing and collecting secret shares increases almost linearly with the increased number of shares. The explanation behind the fact is that, when shares are being distributed by a supply chain party they are encrypted serially and the intended blockchain nodes are decrypting them in parallel. Similarly, when shares are being collected, blockchain nodes are encrypting them in parallel, and the supply chain party is decrypting them serially.

The purpose of the Result Analysis chapter is to measure the time requirements of our proposed system to check its compatibility for fast response required supply chains. Power consumption analysis to check whether it is suitable for lightweight supply chain devices is left for future research.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Supply chains are complex systems. Several parties are directly or indirectly connected to a single supply chain. As the number of connected parties grows, the complexity of supply chain management increases. With the growing number of involved supply chain entities, there will be certain inherent challenges. For instance, all the relevant parties will want to continuously share and access product-related information in real-time in a transparent manner using a common platform. However, apart from these challenges, there might be several attacks on the supply chain such as counterfeiting, data attacks and so on which will necessarily impact several supply chain security concerns.

In this thesis, we have designed a PUF enabled RFID-based tracking system to mitigate the counterfeiting attacks on supply chain products. We have relied on PUF to identify authentic supply chain products. We have validated the benefits of using PUF enabled RFID and find out that it is effective to fight against counterfeiting

attacks from both internal and external adversaries.

Blockchain technology seems to be a perfect solution to the above-mentioned issues since it establishes a shared, distributed, transparent and immutable record of data that can be accessed by anyone who is permitted. Deploying blockchain in the supply chain management system can address several issues like counterfeiting, origin tracking, and lack of trust among supply chain parties. Therefore, exploiting a platform like blockchain can facilitate supply chain management by solving several challenging features of the supply chain. The solution we have proposed based on PUF and blockchain provides high tolerance against product and data integrity attacks.

This paper has introduced a new system architecture that is specially designed for lightweight supply chain devices, since most of the supply chain devices are power constraint. The supply chain parties of our system do not belong to the blockchain network but still can take full advantage of the blockchain. This way the architecture keeps scope for an open supply chain where any desired party can join.

The proposed protocols with the help of PUF and blockchain ensure the verification and transaction of the authentic products. A new protocol has also been introduced here. This protocol assists end user who does not have a supply chain device to verify products. These protocols also work to make the supply chain linear, so that any sold product cannot reenter the supply chain.

After ensuring the product and data integrity with the help of PUF and blockchain, we have also taken into consideration the fact that, the trusted blockchain nodes also might be compromised. To address the information leakage from the blockchain, we have exploited the secret sharing scheme and also used public key cryptography to strengthen the secret sharing.

7.2 Future Work

In our research work, for the sake of simplicity, we have considered non-perishable items only so that the only required IoT device becomes an RFID tag. For the case of RFID tag, the communication between the tag and the outside world is reactive instead of proactive regular interval data transmitting. We have also exploited a mutual authentication between the RFID tag and the reader to make communication more secure. However, we could have also used GPS to locate the products, collect the quality information of the raw materials and products such as temperature, vibration, humidity, aroma, and so on using different sensors. For these types of IoT devices, the communication between the devices and the outside world is proactive and data are transmitted in a regular interval. This could pose a threat of sensitive product related data being hacked by the adversaries. Appropriate encryption techniques might resolve this issue. But incorporating this mechanism in our protocols is out of the scope of this research work and thus was deliberately excluded.

Our proposed blockchain consensus algorithm is a reputation based one, and the authenticity of a message passed between supply chain party and blockchain network solely depends on the representative nodes' leader node's sent message and backup nodes' votes. Although the possibility of all representative nodes being malicious is very less, still there is a chance that malicious representative nodes might collude together and pass a false message. To fight this scenario, we need to take a larger set of representative nodes so that the probability of false messages being passed becomes even less. However, if there is even a single malicious node in the representative nodes' group, the consensus algorithm will run one more time to search for a full set of honest representative nodes, since we allow to pass a message only if there are 100% identical votes, to provide maximum possible security. As the number of rounds increases, the algorithm will take more time to reach consensus. In that case, network policymakers

might set the identical voting requirement to an acceptable value less than 100% so that the consensus is reached faster. But that will be a trade-off between speed and security.

For future reference, we may introduce perishable items in the supply chain. That way supply chain will be able to deal with raw materials. In order to do that we will have to take the assistance of different kinds of sensors to measure the quality parameters of the raw materials and products as well.

Another future work direction might be developing a better blockchain consensus algorithm, where the policymakers will not have to choose between speed and security. Our next research work will try to find a consensus algorithm that is fast and secure at the same time.

We can also expand the range of attacks on the supply chain that can be addressed, instead of dealing only with counterfeiting and data attacks.

In order to motivate the blockchain nodes to behave honestly, we can also introduce an incentive mechanism in the blockchain. So, in addition to penalize the malicious nodes, incentivizing the honest nodes with real-world rewards (apart from reputation) will accelerate the blockchain-related activities.

Finally, the proposed system can be practically implemented by using real supply chain and blockchain devices, and timing (implementation based, rather than simulation based) and power consumption analysis can be performed.

Bibliography

[1] S. Joshi, S. Mohanty, and E. Kougianos, "Everything you wanted to know about pufs," *IEEE Potentials*, vol. 36, pp. 38–46, 11 2017.

[2] D. Closs, J. Mcconnell, C. In, and E. Director, "Enhancing security throughout the supply chain," *Washington, DC: IBM Center for the Business of Government*, 01 2004.

[3] K. Korpela, J. Hallikas, and T. Dahlberg, "Digital supply chain transformation toward blockchain integration," 01 2017.

[4] ESG: a division of TechTarget, "ESG Research Report: Cyber Supply Chain Security Revisited," https://research.esg-global.com/chapters/CyberSupplyChainRevisited/ExecutiveSummary, online; accessed on 28 March 2021.

[5] J. Spink, D. Moyer, H. Park, and J. Heinonen, "Defining the types of counterfeiters, counterfeiting, and offender organizations," *Crime Science*, vol. 2, p. 8, 12 2013.

[6] U. Guin and M. Tehranipoor, "On selection of counterfeit ic detection methods," 05 2013.

[7] Maria Korolov, "Supply chain attacks show why you should be wary of third-party providers," https://www.csoonline.com/article/3191947/what-is-a-supply-chain-attack-why-you-should-be-wary-of-third-party-providers.html, online; accessed on 28 March 2021.

[8] J. Sengupta, S. Ruj, and S. Dasbit, "A comprehensive survey on attacks, security issues and blockchain solutions for iot and iiot," *Journal of Network and Computer Applications*, 11 2019.

[9] J. F. M. Reed, Melinda and P. Popick, "Supply chain attack patterns: Framework and catalog," *Office of the Deputy Assistant Secretary of Defense for Systems Engineering*, 02 2014.

[10] C. E. Shearon, "A practical way to limit counterfeits," in *2019 Pan Pacific Microelectronics Symposium (Pan Pacific)*, 2019, pp. 1–7.

[11] D. Jiang and C. Chong, "Anti-counterfeiting using phosphor puf," 09 2008, pp. 59 – 62.

[12] A. Mahmood, A. Yankovsky, R. Kirichek, and A. Borodin, "Smart system based on doa iot for products monitoring anti-counterfeiting," 01 2019, pp. 1–5.

[13] V. Pathak, "Improving supply chain robustness and preventing counterfeiting through authenticated product labels," in *2010 IEEE International Conference on Technologies for Homeland Security (HST)*, 2010, pp. 35–41.

[14] F. Tian, "An agri-food supply chain traceability system for china based on rfid blockchain technology," 06 2016, pp. 1–6.

[15] M. Ahemd, M. Shah, and A. Wahid, "Iot security: A layered approach for attacks defenses," 04 2017, pp. 104–110.

[16] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Cryptography Mailing list at https://metzdowd.com*, 03 2009.

[17] T. Dinh, J. Wang, G. Chen, R. Liu, B. Ooi, and K.-L. Tan, "Blockbench: A framework for analyzing private blockchains," 03 2017.

[18] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 1–1, 01 2016.

[19] R. Azzi, R. Kilany, and M. Sokhn, "The power of a blockchain-based supply chain," *Computers Industrial Engineering*, vol. 135, 06 2019.

[20] I.-C. Lin and T.-C. Liao, "A survey of blockchain security issues and challenges," *International Journal of Network Security*, vol. 19, pp. 653–659, 09 2017.

[21] L. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," 01 2017, pp. 1–5.

[22] P. Helo and Y. Hao, "Blockchains in operations and supply chains: A model and reference implementation," *Computers Industrial Engineering*, vol. 136, 07 2019.

[23] GeeksforGeeks, "Consensus Algorithms in Blockchain," https://www.geeksforgeeks.org/consensus-algorithms-in-blockchain/, online; accessed on 28 March 2021.

[24] M. S. Ferdous, M. Chowdhury, M. Hoque, and A. Colman, "Blockchain consensus algorithms: A survey," 01 2020.

[25] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," 2012.

[26] M. Castro and B. Liskov, "Practical byzantine fault tolerance," *OSDI*, 03 1999.

[27] 101 Blockchains, "Consensus Algorithms: The Root of Blockchain Technology," https://101blockchains.com/consensus-algorithms-blockchain/, online; accessed on 28 March 2021.

[28] T. Fernández-Caramés and P. Fraga-Lamas, "A review on the use of blockchain for the internet of things," *IEEE Access*, vol. 6, pp. 32 979–33 001, 05 2018.

[29] S.-C. Cha, J.-F. Chen, C. Su, and K.-H. Yeh, "A blockchain connected gateway for ble-based devices in the internet of things," *IEEE Access*, vol. PP, pp. 1–1, 01 2018.

[30] S.-K. Kim, U.-M. Kim, and J.-H. Huh, "A study on improvement of blockchain application to overcome vulnerability of iot multiplatform security," *Energies*, vol. 12, p. 402, 01 2019.

[31] C. Lin, D. He, X. Huang, K.-K. R. Choo, and A. Vasilakos, "Bsein: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0," *Journal of Network and Computer Applications*, vol. 116, pp. 42–52, 08 2018.

[32] L. Aniello, B. Halak, P. Chai, R. Dhall, M. Mihalea, and A. Wilczynski, "Towards a supply chain management system for counterfeit mitigation using blockchain and puf," 08 2019.

[33] F. Zhu, P. Li, H. Xu, and R. Wang, "A lightweight rfid mutual authentication protocol with puf," *Sensors (Basel, Switzerland)*, vol. 19, no. 13, pp. 2957–, 2019.

[34] S. Mohanty, V. P. Yanambaka, E. Kougianos, and D. Puthal, "Pufchain: A hardware-assisted blockchain for sustainable simultaneous device and data security in the internet of everything (ioe)," *IEEE Consumer Electronics Magazine*, vol. 9, pp. 8–16, 03 2020.

[35] T. McGrath, I. E. Bagci, Z. M. Wang, U. Roedig, and R. J. Young, "A puf taxonomy," *Applied Physics Reviews*, vol. 6, no. 1, p. 011303, 2019.

[36] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, "Design and implementation of puf-based "unclonable" rfid ics for anti-counterfeiting and security applications," in *2008 IEEE International Conference on RFID*, 2008, pp. 58–64.

[37] G. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," 07 2007, pp. 9–14.

[38] M. N. Aman, K. C. Chua, and B. Sikdar, "A light-weight mutual authentication protocol for iot systems," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.

[39] M. N. Islam and S. Kundu, "Enabling ic traceability via blockchain pegged to embedded puf," *ACM Transactions on Design Automation of Electronic Systems*, vol. 24, pp. 1–23, 06 2019.

[40] T. Bocek, B. Rodrigues, T. Strasser, and B. Stiller, "Blockchains everywhere - a use-case of blockchains in the pharma supply-chain," 05 2017, pp. 772–777.

[41] P. Helo and Y. Hao, "Blockchains in operations and supply chains: A model and reference implementation," *Computers Industrial Engineering*, vol. 136, 07 2019.

[42] S. Mondal, K. Wijewardena, S. Karuppuswami, F. Nitya Kriti, D. Kumar, and P. Chahal, "Blockchain inspired rfid-based information architecture for food supply chain," *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, 03 2019.

[43] U. Ruhrmair and D. Holcomb, "Pufs at a glance," 01 2014, pp. 1–6.

[44] R. J. McEliece and D. V. Sarwate, "On sharing secrets and reed-solomon codes," *Commun. ACM*, vol. 24, no. 9, p. 583–584, Sep. 1981. [Online]. Available: https://doi.org/10.1145/358746.358762

[45] A. Beimel, "Secret-sharing schemes: A survey," in *Coding and Cryptology*, Y. M. Chee, Z. Guo, S. Ling, F. Shao, Y. Tang, H. Wang, and C. Xing, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 11–46.

[46] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, p. 612–613, Nov. 1979. [Online]. Available: https://doi.org/10.1145/359168.359176

[47] J. Bae and H. Lim, "Random mining group selection to prevent 51bitcoin," 06 2018, pp. 81–82.