PUCRS

ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

RÉGIO ANTONIO MICHELIN

A LIGHTWEIGHT BLOCKCHAIN DATA MODEL FOR THE INTERNET OF THINGS

Porto Alegre
2019

PÓS-GRADUAÇÃO - *STRICTO SENSU*

Pontifícia Universidade Católica
do Rio Grande do Sul

PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
SCHOOL OF TECHNOLOGY
COMPUTER SCIENCE GRADUATE PROGRAM

A LIGHTWEIGHT BLOCKCHAIN DATA MODEL FOR THE INTERNET OF THINGS

RÉGIO ANTONIO MICHELIN

Dissertation submitted to the Pontifical Catholic University of Rio Grande do Sul in partial fulfillment of the requirements for the degree of Ph. D. in Computer Science.

Advisor: Prof. Dr. Avelino Francisco Zorzo

Porto Alegre
2020

Ficha Catalográfica

M623L Michelin, Regio Antonio

A lightweight blockchain data model for the Internet of Things / Regio Antonio Michelin . – 2019.
109 p.
Tese (Doutorado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Avelino Francisco Zorzo.

1. Blockchain. 2. Internet of Things. 3. IoT. 4. Security. I. Zorzo, Avelino Francisco. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).
Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

Régio Antonio Michelin

A lightweight blockchain data model for the Internet of Things

This Thesis has been submitted in partial fulfillment of the requirements for the degree of Doctor of Computer Science, of the Graduate Program in Computer Science, School of Technology of the Pontifícia Universidade Católica do Rio Grande do Sul.

Sanctioned on Mar 27, 2019.

COMMITTEE MEMBERS:

Prof. Dr. Raul Ceretta Nunes (UFSM)

Prof. Dr. Rodrigo da Rosa Righi (PPGCA/UNISINOS)

Prof. Dr. Weverton Luis da Costa Cordeiro (PPGC/UFRGS)

Prof. Dr. Fabiano Passuelo Hessel (PPGCC/PUCRS)

Prof. Dr. Avelino Franciso Zorzo (PPGCC/PUCRS - Advisor)

“A common mistake that people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools.”
(Douglas Adams)

ACKNOWLEDGMENTS

This Ph.D. thesis is a result of the work developed in the last four years, and it was concluded with the support of mentors, colleagues, friends, and family. Many thanks for all the people who in some manner were part of the development of this work, especially to my advisor Professor Dr. Avelino Francisco Zorzo, for the opportunity to develop my Ph.D. and for sharing his knowledge and guidance on this research. A special thanks to my friend and colleague Roben Castagna Lunardi, which helped me to improve the current research.

Thanks in advanced to all Ph.D. committee members that who accepted being part of the thesis evaluation - Prof. Dr. Raul Ceretta Nunes, Prof. Dr. Rodrigo da Rosa Righi, Prof. Dr. Weverton Luis da Costa Cordeiro, and Prof. Dr. Fabiano Passuelo Hessel.

Many thanks to Ph.D. colleagues from CONSEG research group, Charles Neu, Daniel Dalalana, Alex Orozco, Henry Nunes, and Aline Zanin that somehow helped me during this long research path. Thanks to Professor Salil Kanhere from The University of New South Wales for the knowledge exchange among research groups, and especially for his advice during my stay in his group.

Finally, thanks to my family, in particular to son Murilo and my wife, for their patience and acceptance of my many unavailable moments, and for their inspiration to always do good. Thank you also for the support of my parents who always believed in me and encouraged me to continue studying, this thesis is dedicated to you.

A LIGHTWEIGHT BLOCKCHAIN DATA MODEL FOR THE INTERNET OF THINGS

RESUMO

O número de dispositivos conectados a Internet tem aumentado de forma significativa nos últimos anos. Estes devices não estão apenas limitados a computadores tradicionais, mas também se apresenta na forma de dispositivos com hardware limitado, tais como, TVs, câmeras IP, relógios inteligente, capazes de executar processamento de dados e interagir através de uma rede. Devido ao crescimento do uso destes dispositivos através da Internet (IoT), eles passaram a ser um alvo atrativo para usuários maliciosos. O número de ataques executados nos dispositivos IoT apresentam um crescimento significativo nos últimos anos, portanto, é fundamental melhorar a segurança nos dispositivos com hardware limitado. Desse modo, uma nova tecnologia que garanta a integridade dos dados, resiliência através de uma arquitetura descentralizada foi investigada, a fim de apresentar soluções de segurança no ambiente de IoT. Essa tecnologia é chamada blockchain. Apesar dos benefícios que a blockchain traz, ele ainda apresenta algumas desvantagens, como alta demanda de armazenamento, poder de processamento e alta latência, o que poderia restringir sua adoção em ambientes de IoT. Com base nisso, a tese atual propõe uma blockchain leve capaz de rodar em hardware limitado comum usado na IoT. A solução proposta é chamada SpeedyChain. Para que a solução proposta seja considerada leve, é proposto um novo modelo de dados, e através dessa mudança a blockchain é capaz de adicionar uma ou mais transações ao mesmo tempo em diferentes blocks e ainda desacoplar a as trasações dos blocks da blockchain. Para demonstrar a viabilidade da solução proposta, ela foi avaliada em três experimentos: Casa Inteligente, Cidade Inteligente e IoT Industrial. Os resultados alcançados são promissores, o tempo para gerenciar transações não excede a casa de milissegundos. Além disso, o modelo de dados da SpeedyChain é capaz de garantir as propriedades de integridade dos dados e não-repúdio com a intrudução mínima

de processamento extra. Esta tese também discute os principais ataques a blockchains e como a solução proposta pode evitar esses ataques.

Palavras-Chave: *Blockchain*, Internet das Coisas, IoT, Segurança.

A LIGHTWEIGHT BLOCKCHAIN DATA MODEL FOR THE INTERNET OF THINGS

ABSTRACT

The number of connected devices increased significantly in the last years. These devices are not limited to traditional computers, but nowadays it is also common to find hardware-constrained devices, *e.g.*, TVs, IP cameras, smart watches, able to handle information and interact through a computer network. Due to the growing on the use of these devices in the Internet of Things (IoT), they become an attractive target for malicious users. The number of attacks performed in IoT devices increased in the last years, hence it is paramount to improve security on the hardware constrained devices. Therefore, a new technology that guarantees data integrity, resilience and a decentralized architecture has been investigated in order to bring solutions in the IoT environment. This technology is called blockchain. Despite the benefits a blockchain brings, it still presents some drawbacks, such as, high storage demand, processing power demand and high latency, which could restrain its adoption in IoT environments. Based on that, the current thesis proposes a lightweight blockchain able to run in common constrained hardware used in IoT. The proposed solution is called SpeedyChain. To achieve the lightweight solution, a new data model is proposed, and this change makes the blockchain able to add one or more transactions at the same time and still decouple the payload from the blockchain. In order to show the viability of the proposed solution, it was applied to three experiments: Smart Home, Smart City, and Industrial IoT. The achieved results are promising, keeping the time to handle transactions in milliseconds. Furthermore, SpeedyChain data model is able to ensure data integrity as well as non-repudiation security properties, with a minimum processing overhead. This thesis also discusses main attacks on blockchains and how the proposed solution avoids these attacks.

Keywords: Blockchain, Internet of Things, IoT, Security.

LIST OF FIGURES

Figure 2.1 – Bitcoin transaction fields [Nak16] 34
Figure 2.2 – Merkle tree sample 35
Figure 2.3 – Bitcoin block structure [Nak08] 36
Figure 2.4 – Bitcoin transaction chain [Nak08] 38
Figure 2.5 – Mining Pool 39
Figure 2.6 – Hashrate distribution among the largest mining pools (03/01/2019) [Blo19] 40
Figure 2.7 – Blockchain layers [ZNL+18] 42
Figure 2.8 – IoT tiers architecture (adapted from [GKN+11]) 43
Figure 4.1 – Double Spending attack 57
Figure 4.2 – Finney attack 58
Figure 4.3 – Vector 76 attack 59
Figure 4.4 – Selfish mining attack 60
Figure 4.5 – Fork after withhold attack (Adapted from [Col18]) 62
Figure 4.6 – AS8 acting as a rogue user intercepting part of the traffic and delay the delivery of a block for 20 minutes to a victim C (adapted from [AZV17]) . 66
Figure 4.7 – Sybil attack 66
Figure 4.8 – Eclipse attack 67
Figure 5.1 – Architecture components 71
Figure 5.2 – SpeedyChain components 73
Figure 6.1 – Smart Home hardware architecture 84
Figure 6.2 – Performance for appending and sending information to the gateways 85
Figure 6.3 – Performance for appending new transaction on a block 85
Figure 6.4 – Smart City evaluated architecture 86
Figure 6.5 – Required time in an RSI to add a new block to the blockchain. 87
Figure 6.6 – Required processing time to add new transaction to the blockchain. . 88
Figure 6.7 – Time (ms) to update peer's blockchain with received transactions . . . 89
Figure 6.8 – Time (ms) to update peer's blockchain with received blocks 90
Figure 6.9 – Time for block consensus 91

LIST OF TABLES

Table 1.1 – Number of papers using “blockchain” as query string - from 2016 to 2018 28
Table 1.2 – Number of papers using “blockchain and security” as query string - from 2016 to 2018 28
Table 1.3 – Number of papers using “blockchain and IoT” as query string - from 2016 to 2018 28
Table 1.4 – Number of papers for “blockchain and security and iot” string - from 2016 to 2018 29
Table 3.1 – Blockchain comparison (data collected on 08/02/2019) [Bit19] 45
Table 3.2 – IoT related work general information and security demand 48
Table 3.3 – Smart cities related work aspects comparison 50
Table 3.4 – Industrial IoT related work summary 52
Table 3.5 – Related work summary 53
Table 4.1 – Number of papers that considers blockchain vulnerabilities 55
Table 4.2 – Blockchain vulnerability summary 70
Table 5.1 – SpeedyChain attacks analysis 79
Table 6.1 – Performance evaluation of constrained devices with RSA, AES256 and SHA256 82
Table 6.2 – Performance for connecting and appending new block in a blockchain 83
Table 6.3 – Time taken by vehicles to calculate the Merkle tree root. 89
Table 6.4 – Performance evaluation considering different consensus 91

LIST OF ALGORITHMS

2.1 Simplified mining process 35
2.2 Bitcoin block header hash definition 37
5.1 Insertion of new blocks in SpeedyChain 74
5.2 Appending new transactions into the block ledger 75
5.3 Algorithm for key update 76
5.4 Generic consensus algorithm 77

LIST OF ACRONYMS

AES – Advanced Encryption System

API – Application Program Interface

BAAS – Blockchain-as-a-Service

BC – Blockchain

BIP – Bitcoin Improvement Proposals

CA – Certificate Authority

CONSEG – Grupo de Confiabilidade e Segurança de Sistemas

CPU – Central Processing Unit

DDOS – Distributed Denial of Service

DLT – Distributed Ledger Technology

DNS – Domain Name Server

ECDSA – Elliptic Curve Digital Signature Algorithm

FAW – Fork After Withholding

FPOW – Full Proof-of-Work

GB – Gigabyte

GPS – Global Positioning System

INPI – Instituto Nacional de Propriedade Industrial

IOT – Internet of Things

IP – Internet Protocol

ITS – Intelligent Transportation System

M2M – Machine to Machine

MHZ – Megahertz

OS – Operating System

OWASP – Open Web Application Security Project

P2P – Peer to Peer

PBFT – Practical Byzantine Fault Tolerance

PC – Personal Computer

PHD – Doctor of Philosophy

POS – Proof-of-Stake

POW – Proof-of-Work

PPOW – Partial Proof-of-Work

PUCRS – Pontifical Catholic University of Rio Grande do Sul

RAM – Random Access Memory

RFID – Radio Frequency Identification

RSA – Rivest-Shamir-Adleman

RSI – Road Side Infrastructure

RSU – Road Side Unity

SC – Smart Contract

SDN – Software Defined Network

SHA – Secure Hash Algorithm

SP – Service Provider

SSD – Solid State Drive

TV – Television

TX – Transaction

UNSW – University of New South Wales

USD – United States Dollar

V2I – Vehicle to Infrastructure

VM – Virtual Machine

CONTENTS

1 INTRODUCTION 25
1.1 RESEARCH SCOPE 27
1.2 HYPOTHESIS AND RESEARCH QUESTIONS 29
1.3 THESIS PUBLICATIONS 30
1.4 THESIS STRUCTURE 31

2 BACKGROUND 33
2.1 BITCOIN BLOCKCHAIN 33
2.1.1 MINING POOLS 39
2.1.2 SMART CONTRACTS 40
2.2 BLOCKCHAIN LAYERS 41
2.3 CHALLENGES OF APPLYING BLOCKCHAIN IN IOT 42

3 RELATED WORK 45
3.1 BLOCKCHAINS 45
3.2 INTERNET OF THINGS 46
3.3 SMART CITIES 48
3.4 INDUSTRIAL INTERNET OF THINGS 50
3.5 CHAPTER SUMMARY 52

4 SECURITY ASPECTS 55
4.1 BLOCKCHAIN ATTACKS 56
4.2 CHAPTER SUMMARY 69

5 SPEEDYCHAIN FRAMEWORK DEFINITION 71
5.1 ARCHITECTURE 71
5.2 BLOCKCHAIN DEFINITION 72
5.3 SPEEDYCHAIN MAIN OPERATIONS 74
5.3.1 APPENDING BLOCKS 74
5.3.2 APPENDING TRANSACTION 75
5.3.3 KEY UPDATE 75
5.3.4 CONSENSUS 76
5.4 SECURITY ANALYSIS 77

5.5 CHAPTER SUMMARY 80

6 EVALUATION 81
6.1 SMART HOME 81
6.2 SMART CITY 85
6.3 INDUSTRIAL INTERNET OF THINGS 89
6.4 DISCUSSION 92

7 CONCLUSION 95
7.1 HYPOTHESIS FOUNDATION 95
7.2 FUTURE DIRECTIONS 97

REFERENCES 99

1. INTRODUCTION

The popularization of devices with some embedded technology has become increased in the past years, especially for end-users. These devices are equipped with components such as processor, memory, power, and so on, which can provide devices data handling capability, making them able to interact with the environment or produce data. These devices are popular among end-users not only due to this environment interaction capability but also due to their small size, which makes many of them portable. This devices list is vast; however, we can highlight devices such as smartwatches, smart shoes (among other wearable devices), IP cameras, smart TVs, safety/security equipment applied in smart homes/offices, temperature sensors, smart vehicles, traffic lights and so on. All these devices have some processing capacity, enabling them to execute computing tasks. Additionally, these computing tasks performed by devices enable them to communicate with other devices, once a network connection is available, and produced information can be exchanged. Once a device can perform some computing tasks and also capable of exchanging information through a network connection, we can say that the device is part of the Internet of Things (IoT).

The term IoT was defined, initially around 1999 by Kevin Ashton to manage a production chain through Radio Frequency Identification (RFID) [Ash09]. Despite that, the IoT concept is recent in the Computer Science field, and its definition does not have a consensus among researchers. Despite this lack of consensus, IoT is applied in different domains such as smart cities [ZBC+14], smart offices [MA18], smart homes [KSM13], wearables [AWHJ15], pervasive and ubiquitous computing [Xu11].

The IoT concept follows a multidisciplinary domain and it considers that machines can sense the environment where they are located, also through actions, they are capable of changing this environment [GKN+11]. Furthermore, this multidisciplinary domain can be divided into [AIM10]: (*i*) transportation and logistics; (*ii*) health care; (*iii*) smart environment (such as houses, offices, factories) and, (*iv*) social/personal. In order to enable the device application among the different domains, IoT must ensure a communication layer and all devices must be equipped with microcontrollers and transceivers that enable them to interact through this layer [AIM10]. Once this capability is present, the devices can interact with other devices, in the same way as they can interact with a user. Hence, this research considers **as IoT a network of devices capable of performing processing tasks and able to exchange information**.

As stated before, communication and processing capabilities are the main properties that help IoT devices to reach end-users. However, ensuring device security is a challenging task, especially due to devices processing capability and connectivity and how these devices interact with the users' environment. Current research [MYAZ15] [AMV16]

[AK14] [RNL11] [SRGCP15] shows that there are different security threats that can affect IoT devices or their infrastructure. On October 2016 a famous attack against an Internet service had a huge impact on many other services on the Internet. Particularly, in that attack, the Mirai botnet [Jga16] used devices with default configurations (especially default user and password) to attack a dynamic Domain Name Server (DNS) provider, *i.e.* the Dyn DNS. In that attack, millions of devices, *e.g.* IP cameras, vacuum cleaners, and domestic routers were used to produce a Distributed Denial of Service (DDoS) attack. Consequently, different applications and services that were using this dynamic DNS provider became unavailable [Jga16]. This single example shows the importance of enforcing security for any device that is connected to the Internet. From 2016 to 2018, the number of attacks on IoT has increased three-fold [Kas19], which shows that this problem is increasing as the number of devices on IoT increases. Some companies foresee that by 2025, more than 75 billion devices will be connected to the Internet around the world [Sar19].

Searching to identify or even propose a solution to IoT devices security, several different types of research have been developed during the last years in different levels of the IoT architecture (see Section 3.2). The main concerns for IoT security are related to protocol and network, data privacy, identity management, trust and governance, and fault tolerance. Based on these concerning points, different technologies could be applied to improve device security [RNL11].

Among the new technologies that are being presented to improve IoT devices security, blockchain [Nak08] has caught significant attention either from industry and academia. A blockchain is a distributed ledger technology (DLT) that gathers together different basic Computer Science concepts, for example, linked list structure, consensus algorithms, cryptography algorithms such as public key cryptography and hash functions, and a Peer-to-Peer (P2P) network. This technology initially was conceived to act as a public distributed ledger for an online cryptocurrency, although it can be applied to different contexts, such as, access control, and data management.

The blockchain applicability in IoT context could benefit from some characteristics presented by the blockchain technology. The resilience is a property that is present in the blockchain concept [LZSL17], for example. This benefit can maintain an IoT solution working even in an attack to a device or its unavailability, avoiding single points of failure [ZJ18] and giving high availability to the IoT network. Furthermore, the method in which information is appended to the blockchain ensures transparency, tampering resistance [SS16] and non-repudiation.

Despite providing some characteristics that improve an IoT solution in different aspects, blockchain technology presents problems related to the existing hardware constrained in some of the IoT devices. Among the possible limitations is the ledger size [ZLWS18], as the blockchain stores all the created information, which leads to considerable high storage demands. Another limitation is related to the high processing demand [ZLWS18], which is

caused by the algorithm that runs in the blockchain to make the information produced by network nodes to become trusted in an untrusted environment. These are some of the key points to be addressed when applying the blockchain technology into the IoT context. This research will consider as a lightweight solution in terms of low processing latency overhead, information management and required solution storage.

Motivated by the properties presented by the blockchain, and research to improve its capability, in a way that fits the IoT requirements, the current research is **focused on proposing a blockchain lightweight data model to be applied in the IoT context ensuring the security aspects**.

1.1 Research scope

IoT devices present several challenges regarding the support of a security mechanism. Most of these challenges are related to applying security in constrained IoT devices [CZ16] [RNL11], and the overhead that these security measures introduced. Among the security measures that are commonly applied to solutions, the authentication is a first stage to control/limit the user access [OMAA17]; however, the traditional approach using a combination of user and password is considered obsolete [BHvOS12].

Therefore, this research started evaluating authentication mechanisms to replace the traditional combination of user and passwords. In order to find the existing solutions for a distributed architecture to provide an authentication mechanism that would avoid user and password combination, a literature review was performed. During this review, it was set, also, that the solution should be able to store the authentication information and guarantee integrity and resilience. Solutions that seemed to stand out were the ones that are using the blockchain technology [OMAA17] [AEVL16] [SI16].

Since blockchain was identified as a possible research subject that would solve the authentication problem, a different literature review was performed in order to identify state of the art related to blockchain research. The first step was to find the number of papers that were published considering only the term “blockchain”, which was applied as a search string in different digital libraries, *i.e.* ACM, IEEE and Springer. As can be seen in Table 1.1, in 2016, very few papers were published with the term blockchain in its metadata. In the following years, the subject became a research topic for several researcher groups around the world, it was more than tenfold in 3 years.

On the one hand, the second step was performed, aiming to understand the research gap regarding security on the blockchain. In order to do that, a new search string was applied to the same digital libraries from Step 1, *i.e.* “blockchain and security”. The result, in the number of papers, from 2016 to 2018, is shown in Table 1.2. As can be seen in the table, the number of papers was reduced considerably.

Table 1.1 – Number of papers using “blockchain” as query string - from 2016 to 2018

Portal	**2016**	**2017**	**2018**
IEEE	57	324	1,041
ACM	33	105	264
Springer	68	326	681

Table 1.2 – Number of papers using “blockchain and security” as query string - from 2016 to 2018

Portal	**2016**	**2017**	**2018**
IEEE	25	165	515
ACM	22	67	152
Springer	58	267	536

On the other hand, we inspect the blockchain applicability in the context of IoT. To perform that, we produced a new search string to see how many papers were published regarding “blockchain and IoT”. The results of this search are shown in Table 1.3. Similarly to blockchain and security search, the search using the “blockchain and IoT” string also returned few papers, even less than “blockchain and security”.

Table 1.3 – Number of papers using “blockchain and IoT” as query string - from 2016 to 2018

Portal	**2016**	**2017**	**2018**
IEEE	10	70	265
ACM	5	26	46
Springer	11	78	242

The final step was to understand how blockchain had been used to provide more security for IoT environments. Therefore, we produced a new string, *i.e.* “blockchain and security and IoT”, and applied it to the same search engines from the same digital libraries. As can be seen in Table 1.4, the number of papers that were published, regarding this combination, was very limited by 2016. Besides the low number of papers, most of them did not present any solution, but rather just discussed the application of blockchain to provide security in the context of IoT. Most of the solutions started to be published only by 2017 and 2018.

As can be seen in the previous tables, the number of research papers published in the last years has increased considerably regarding blockchain and its applicability for security in the IoT context. Despite that, blockchain technology still presents challenges in terms of scalability, ledger size and power-consuming, when applied to the IoT context, which leads to a broader research field. Hence several research questions were established and a hypothesis that will be validated in this thesis.

It is important to mention that the results presented in Tables 1.1, 1.2, 1.3 and 1.4 were collected on February 20, 2019. Naturally, at the beginning of this research, the number of papers in 2016 were less than the ones now present in those tables, since it takes

Table 1.4 – Number of papers for “blockchain and security and iot” string - from 2016 to 2018

Portal	**2016**	**2017**	**2018**
IEEE	4	46	164
ACM	1	18	37
Springer	9	65	210

time till all papers are stored in the digital libraries. Just as an example, by the beginning of January 2019 there were 941 papers regarding “blockchain” in the IEEE digital library, and by February 20, 2019, there were 1,041.

As the outcome from this review, we identified that the proposed blockchain must be able to adapt to the IoT context. As part of its adaptation, a change in the blockchain data model was performed in order to make it lightweight and still ensures data integrity and non-repudiation. It is important to highlight that during this research, it is assumed that the hardware is secure and trustworthy. This assumption guides research to define a permissioned blockchain, combining trusted and untrusted parties.

1.2 Hypothesis and Research Questions

Taking the main problems that current blockchain solutions present when applied to provide security for IoT contexts, this PhD thesis aims to investigate the following hypothesis:

A data model to provide a lightweight blockchain for devices in the Internet of Things can be applied in a less time-consuming solution than traditional blockchains, nonetheless, keeping the same security level.

Hence, the following research questions were established to support the validation of the set hypothesis:

1. *Which are the most common hardware available capable of run a blockchain?* Once this question is answered, it could lead to identifying the hardware that can run a blockchain, and this leads to the next question.

2. *What is the performance in constrained hardware to handle the algorithms needed for supporting a blockchain?* Answering this question would help to understand the overhead caused by the blockchain algorithms. Thus, this answer could point to scenarios where each different hardware could be used.

3. *How to change/adapt the current blockchain technology in order to become a lightweight solution capable of fitting in embedded hardware (such as constrained hardware IoT devices)?* The answer to this question should help to drive the blockchain architecture

in order to design a lightweight solution. This answer should help to identify what are the overhead bottlenecks in a blockchain and to find possible alternative solutions to keep the blockchain properties and, at the same time, to fit in a constrained devices IoT environment.

4. *What are the most common security threats that could compromise a blockchain?* This question aims to leverage the study of security risks for new technology. Like any new technology, a security analysis should be conducted in order to identify flaws and a possible workaround.

5. *How to propose an alternative data model for blockchain that keeps a high-security level?* Once the security threats are identified, the answer to this question should produce a data model that allows a blockchain to runs in a constrained environment and still keep the security aspects that were identified and analyzed.

1.3 Thesis publications

This thesis has produced the following research papers:

1. *A Decentralised Approach to Task Allocation Using Blockchain* [BMZB18]. Initially, this research aimed to identify the blockchain applicability in a different context from its traditional usage in Bitcoin. This paper proposed architecture for dynamic and decentralised allocation of tasks built on the idea of having communication and coordination in a multi-agent system through a private blockchain.

2. *Distributed access control on IoT ledger-based architecture* [LMNZ18]. This paper proposed an IoT ledger-based architecture to ensure access control on heterogeneous scenarios. This research applied conventional devices used on IoT networks, such as Arduino, Raspberry, and Orange Pi boards, and evaluated the solution performance. This paper presented the appendable block concept for blockchain.

3. *SpeedyChain: A framework for decoupling data from blockchain for smart cities* [MDS+18]. This paper is an extension from the “Distributed access control on IoT ledger-based architecture” paper. The new framework was applied to a smart city scenario, and the research was conducted in cooperation with the University of New South Wales (UNSW). In this paper, a blockchain framework that decouples the data stored in the transactions from the block header is proposed. This data structure allowed fast addition of data to the blocks using an expiration time for each block and a witness-based mechanism in order to create new blocks.

4. *Dependable IoT using blockchain-based technology* [ZNL+18]. This paper presented some discussion about the usage of blockchain technology in IoT environments and proposed a layer model of blockchains for IoT. Additionally, it presented an overview of the latest research regarding network architectures, consensus algorithms, data management, and applications.

5. *A reputation system based on blockchain to detect fake news* [LSL+18]. This paper presents a complete use case using the SpeedyChain blockchain framework. In this research, the blockchain acts as a back-end mechanism and is used to keep the votes and news historical information, thus working as input to allow the user reputation definition.

6. *A Lightweight Blockchain for Industrial IoT* [Lun18] (**to be submitted**). This paper will present a detailed security analysis pointing to the most common blockchain security threats and how these issues could affect SpeedyChain. Additionally, the framework is evaluated in an Industrial IoT scenario using a PBFT consensus algorithm.

7. *Blockchain Technologies for IoT* [DJD+20] (**to be submitted**). This research is a work in progress book chapter that analyses the blockchain applicability for IoT domain from different aspects such as, data model, security, consensus algorithms. This research has been performed together with UNSW, Data61, and PUCRS. Springer will publish this chapter.

All the above mentioned publications were related to using a framework that uses a data model that allows decoupling the payload from a blockchain, hence, allowing information to be appended to and at the same time to ensure that data is secured. This framework is called SpeedyChain and was registered at *Instituto Nacional de Propriedade Industrial* (INPI) under number: BR512018001343-0. This framework was applied to different scenarios, such as Smart Home/Offices, Smart Cities, Industrial IoT, and Fake News (see above publications). Additionally, during the current research, there was also a collaboration in research projects between our research group at PUCRS and companies installed at TecnoPUC.

1.4 Thesis Structure

The remainder of this thesis is organized as follows. Chapter 2 presents the fundamental concepts required to understand the subject of this research approaches. Next, Chapter 3 introduces literature research presenting the related work in which blockchain is applied in the IoT domain. Chapter 4 presents a security evaluation from different security threats that could somehow compromise a blockchain normal behaviour. Chapter 5 presents

the core of this document, which is the formal definition for a lightweight data structure for blockchains. Following, Chapter 6 presents the results in three different use cases where the proposed blockchain data structure is applied. Finally, Chapter 7 presents the final remarks on this thesis, and the hypothesis validation supported research questions answers and points for future direction to the current research.

2. BACKGROUND

This chapter introduces the terminologies and concepts used in this thesis. The subjects in this chapter present the blockchain technology and they briefly describe the working process of the Bitcoin blockchain (Bitcoin was chosen to present the main blockchain concepts due to its historical relevance). Also, this section contains challenges that blockchain present in IoT contexts.

2.1 Bitcoin blockchain

In 2008, Satoshi Nakamoto [Nak08] published the paper *Bitcoin: A peer-to-peer electronic cash system*, which presented a mechanism that enables a definition of a digital currency called Bitcoin. In the paper, they proposed the creation of a currency that presented the following features: decentralization, public and distributed ledger, no central authority trust, certification of the currency ownership, double spending control and data forging avoidance. Since its proposal, this cryptocurrency has increased its market share, and brought about the technology that makes Bitcoin works, *i.e.* blockchain.

Blockchain acts as the main technology in the Bitcoin cryptocurrency, ensuring that transactions are kept public and decentralized. Blockchain is also responsible for providing a control mechanism to avoid double-spending by controlling the coin ownership. Through the blockchain, a transaction ledger is created, storing all the transactions performed since the Bitcoin conception. Each transaction consists of a coin ownership transference, from the current owner to the new owner, *i.e.*, every time that someone executes a coin transference, a new transaction is created. It is important to notice that for every single transaction, in the Bitcoin blockchain, a coin exchange is performed.

Figure 2.1 shows how a transaction is defined in the Bitcoin blockchain. Each transaction is composed of three main fields groups, which are: inputs, outputs, and control. As part of transaction control, the fields *Version*, *#Inputs*, *#Outputs* (both fields indicate how many inputs and outputs are used to define the transaction), and *Locktime* (which indicates the earliest time when the transaction may be added to the blockchain). Besides that, there is a set of fields that describes the inputs and outputs that are in a transaction. The input definition is composed by *Previous Transaction Hash*, *Previous Output Index*, *Script Length*, *ScriptSig* (which is a sequence of instructions in order to ensure the destination address and to provide a proof of ownership), and *Sequence*. The output is defined by *Value*, *Script Length* and *Script Public Key*.

In the Bitcoin cryptocurrency, there is an algorithm that defines that a transaction fee must be paid when an amount of coins is exchanged between two parties. This fee

value is defined by the party that is sending the coins, and the more it is paid, the greater is the possibility of that transaction is inserted in the blockchain. This fee will generate a new transaction from the party that is sending the coins to the party that mines the transaction to a block. For example, imagine that party A wants to send 50 coins to party B, willing to pay 5 coins as transaction fee. Therefore, one transaction for the exchange of 50 coins between A and B is created. When party C creates a new block with that transaction, party C also includes a new transaction that transfers 5 coins from A to C in the created block.

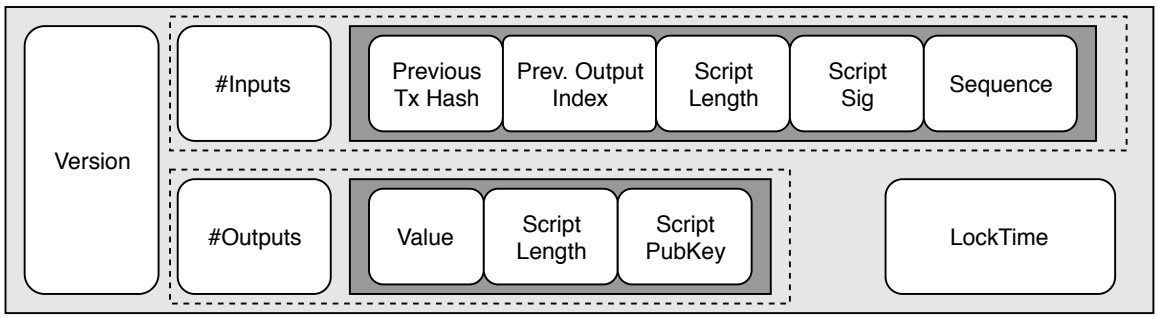

Figure 2.1 – Bitcoin transaction fields [Nak16]

Every time a transaction is created, it is sent to a pool of transactions, called mempool in the Bitcoin network. The mempool contains all transactions that were created, but not yet inserted into a block in the blockchain. The process of inserting a transaction into a block in the blockchain is performed by a party (peer) from the Bitcoin network. Basically, any peer can choose transactions[1] from the mempool to be inserted into the block. After that, the peer creates a Merkle tree based on the chosen transactions. Figure 2.2 shows an example of a Merkle tree that is created for four transaction: *Tx0*, *Tx1*, *Tx2* and *Tx3*. Basically, in a Merkle tree, each transaction is used to calculate a hash value to be inserted into the tree. For example, A = hash(Tx0) is inserted as a leaf node in the tree, B = hash(Tx1) is inserted as a leaf node in the tree, and so on. After that, a new hash is calculated based on the hashes of the leaves nodes, for example, E = hash(A|B)[2]. The next nodes are created based on the two children nodes, for example, the Merkle tree root = hash(E|F). Only the root hash is stored in the block that is a candidate to be inserted into the blockchain.

After that the candidate block is constructed, the peer has to calculate the hash value for that block. In order to calculate the hash value, the peer has to solve a mathematical puzzle. When the peer is solving the puzzle, it will be known as miner. Basically, this mathematical puzzle consists of finding a hash value that starts with a pre-defined number of zero bits (known as puzzle difficulty). For example, if the difficulty is set to 8 and the hash number consists of 16 bits, then a solution for the puzzle would be 0x00AB. Essentially, the data that is inserted in the block is fixed, *i.e.* transactions, Merkle tree root hash, Version, etc. (see Figure 2.3), however, the nonce field must be modified to solve the puzzle. Hence,

[1] On Feb. 06, 2019, the number of transactions that would be chosen by a peer varied from 2,000 to 2,500.
[2] Symbol '|' represents the concatenation of two values.

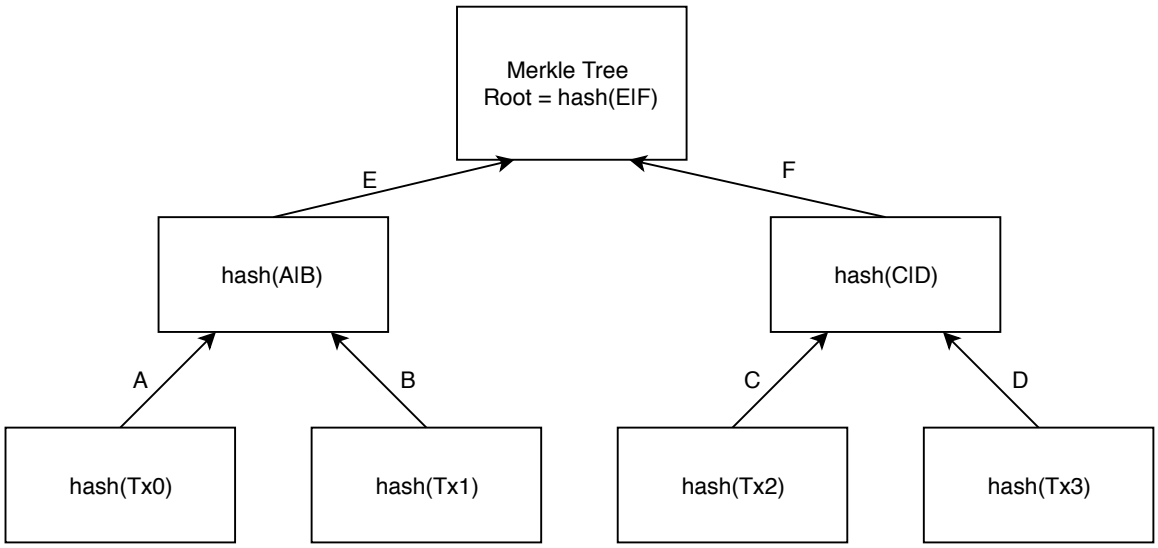

Figure 2.2 – Merkle tree sample

the miner has to find the nonce that is concatenated to the rest of the block and generates a hash value that starts with the pre-defined number of zero bits, the block is considered closed. This process is known as mining a block. Algorithm 2.1 shows how the mining process is performed.

Algorithm 2.1 Simplified mining process

```
1: while nonce < difficulty do
2:     if sha(sha(block+nonce)) < target then
3:         return nonce
4:     end if
5:     nonce++
6: end while
```

The puzzle difficulty is reevaluated every time 2,016 blocks have been inserted into the blockchain. Basically, the new difficulty is calculated by taking the old difficulty and multiplying that by 2,016 and again by 10. Ten represents the time that it is expected to a miner to solve the puzzle, *i.e*, in this case, ten minutes. After that, the result is divided by the time it took to mine the last 2,016 blocks. This value is recovered from the timestamp that is stored in the blocks (see Figure 2.3). Equation 2.1 shows the formula for the new difficulty.

$$new\_difficulty = \frac{old\_difficulty \times (2016\_blocks \times 10\_minutes)}{(the\_time\_took\_in\_minutes\_to\_mine\_the\_last\_2016\_blocks)} \quad (2.1)$$

The process of finding the nonce is part of what is called a consensus algorithm. In the Bitcoin case, and as explained in previous paragraphs, this consensus algorithm is called Proof-of-Work (PoW). This algorithm is part of algorithms known as lottery algorithms

[Bal19], since it rewards the peer first finds the nonce. It is called PoW because it requires a computing work to find the hash. Usually, the peer that has more computing power will work faster and, therefore, will find the nonce first. Despite finding the nonce and having a complete block to be inserted in the blockchain, it is not the peer that created the block that will insert that block in the blockchain. The peer will send the created block to other peers (usually six peers) that will verify whether the transactions are valid, the Merkle tree root hash is correct, and whether the found nonce meets the puzzle difficulty.

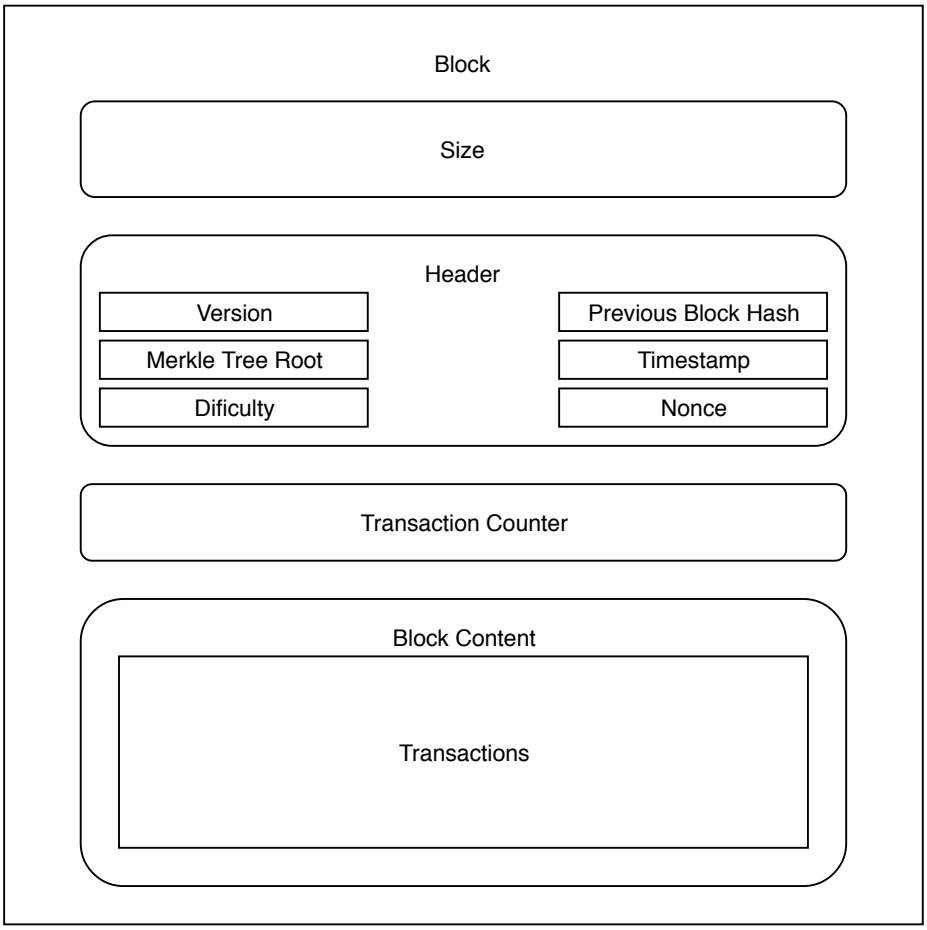

Figure 2.3 – Bitcoin block structure [Nak08]

The miners play a crucial role to keep the Bitcoin blockchain working. Hence, a reward is paid for miners that successfully are able to create a node that will be inserted into the blockchain. The reward is paid in Bitcoins[3] and the transactions fees. The reward is given to the miner through a CoinBase transaction, which, basically, consists of a different transaction that transfers the reward to the miner.

Every block is initially identified by a fixed four-byte value, *i.e.* 0xD9B4BEF9, to indicate where that block starts and follows the fields shown in Figure 2.3. The *Size* field (4 bytes) contains the block size, which can vary due to the number of transactions included in each block. The *Transaction Counter* field (1-9 bytes) refers to the number of transactions

[3]The reward in November 2018 was 12.5 Bitcoins

stored in the block. There is a subset of fields that composes the block header: *Version* (4 bytes), which is used to identify what rules set was applied to validate the specific block; *Previous Block Hash* (32 bytes), is used to create the link between the new block and its predecessor; *Merkle Tree Root* (32 bytes), is the value that represents the summary of all transactions in the block; *Timestamp* (4 bytes), current time that represents when the block was created; finally, *Difficulty* (4 bytes) and *Nonce* (4 bytes) as previously described.

In order to create a link between the blocks, a hash algorithm is used. The Bitcoin blockchain uses SHA-256 (see Algorithm 2.2) in order to create the block hash, and it includes this information in the next block, thus creating the chain concept. It is important to emphasize that once the chain is created, any change on the sequence, compromises the whole chain. The main characteristic of the SHA-256 algorithm is that given an input value, it always results in the same number of 256 bits, and its computation and validation operation (*i.e.* calculate the hash of a given value) is very cheap in computing terms. It is important to highlight that to find the original value from which the hash value was originated is computationally very expensive (almost impossible, given the current processing power).

Algorithm 2.2 Bitcoin block header hash definition

```
1: header_bin = version + prevBlockHash + rootHash + time + bits + nonce;
2: header_hash = hash('sha256', header_bin );
```

So far, we explained how the blocks are created and linked together to build the Bitcoin blockchain and to guarantee integrity. Non-repudiation and authentication are guaranteed using private-public cryptography (or asymmetric cryptography). Privacy is not the major concern from the Bitcoin cryptocurrency, but it can be achieved if the private and public keys are not associated to a person, hence, even though all transactions can be verified or seen, it is not, directly possible to associate those to a person. Usually, people use different key pair to different transactions in order to avoid being discovered.

The whole Bitcoin blockchain relies on the asymmetric Elliptic Curve Digital Signature Algorithm (ECDSA) [ZLS11]. The public key act as a user address, also called wallet. The private key should be kept secret, as it will be used in order to sign the transactions to allow a user to transfer some amount of coins from one wallet to another. The receiver needs to provide the public key.

Figure 2.4 shows how the transactions are chained in the Bitcoin blockchain. The transactions rely on the key par, while the public key is used in order to identify the user that receives the coins in the transaction, the private key is applied to sign the transaction. It is important to notice that despite all transactions belong to a sequence, *i.e.*, every new transaction stores the hash of the previous transaction, this sequence is not kept in a block, as the transactions are distributed among different blocks. As the transactions created are sent to mempool and only when a miner picks the transaction and solves the puzzle, that transaction will be persisted in a block.

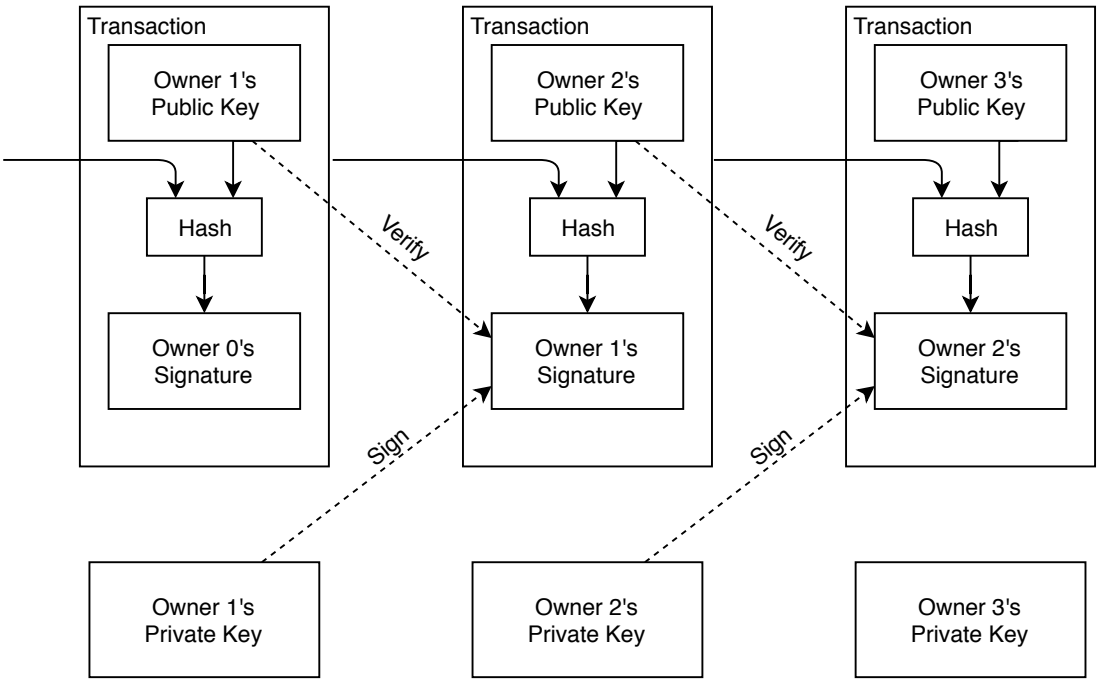

Figure 2.4 – Bitcoin transaction chain [Nak08]

The Bitcoin blockchain uses a peer-to-peer (P2P) network. Due to this architecture, each peer is responsible for keeping a blockchain copy, and this copy is updated by the connected nodes. In the original Bitcoin proposal, each peer should be connected to eight peers. This approach also makes the network resilient, since if any node fails, all other nodes can still run the application.

As mentioned before, the block consensus algorithm, PoW, is executed concurrently by several nodes in the network, and only one node "wins" the race to find the puzzle solution. However, this reward mechanism could not suit for scenarios where the computing processing power and storage are limited, such as in an IoT context. Hence different consensus algorithms have been proposed in the past years.

One sample of consensus algorithms is the *Proof-of-Stake* [WFN+16]. This algorithm has been considered for the Ethereum Foundation as an alternative to the PoW. This algorithm is based on the amount of coins (or anything) that each peer possesses. By design, the PoS allows the peer that holds the biggest amount of coins to perform the block validation. However, this approach presents a problem, related to the centralization, where the richest peer will get advantages when voting to validate blocks. In order to avoid this problem, researchers are proposing alternatives such as randomly selecting a node to vote and the coin age (once a node holds an amount of coins, this coin age is computed, and gives to its owner a per cent of chances to validate a new block). Several other consensus algorithms have been proposed: PBFT, DBFT, Tendermint, Algorant, etc.

Based on the blockchain technology definition introduced by Bitcoin presents the following properties: no central authority, auditability, data integrity, and resilience. All these

properties are supported by the different technologies applied in different contexts, for example, cryptocurrency [Nak08], healthcare [AEVL16], advertising [PKC+18], insurance [Nat16], copyright protection [KFW+15], energy [KYH+17], and others.

2.1.1 Mining pools

As previously described, the main Bitcoin blockchain consensus algorithm (PoW) rewards the node that solves a puzzle problem (the process encourage higher processing power). However, the required work to solve this puzzle is computationally expansive. According to Equation 2.1, the puzzle complexity is adjusted after every 2,016 blocks appended to the blockchain.

The puzzle complexity increase leads the miners to raise their processing power (hash rate) either. One of the most common approaches to boost the miners processing power is to group several nodes working together to solve the puzzle from the same block. This mining grouping is given the mining pool name.

The concept of creating a mining pool is presented in Figure 2.5, where different miners work together in order to solve the same puzzle. The pool concept is a node that chooses transactions from the mempool and creates a block that should be mined, and divides the work calculus among all the miners that are in the same pool. When the nonce is found for that block, the reward is divided among all the participants according to their contribution to solving the puzzle.

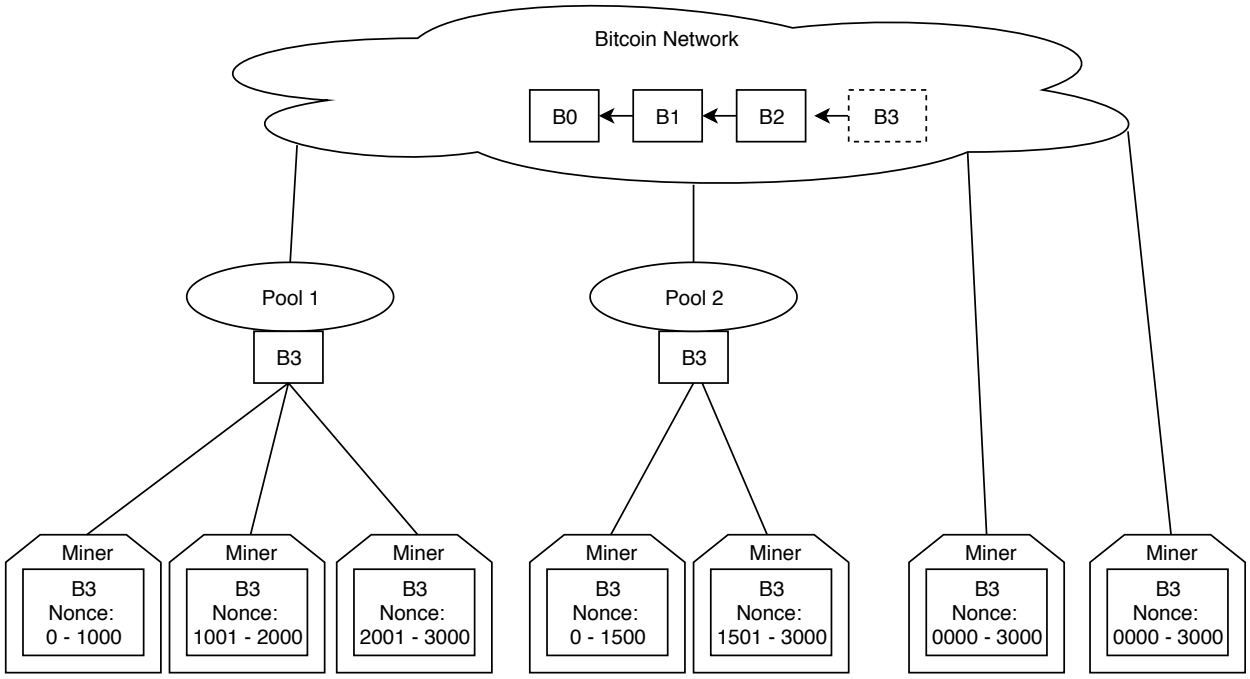

Figure 2.5 – Mining Pool

Nowadays the most processing power in the Bitcoin network is available at mining pools as shown in Figure 2.6. The four biggest mining pools (BTC.com, F2Pool, AntPool, SlushPool) are responsible for 48.2% of the Bitcoin network hash rate power.

As different consensus algorithms are being proposed, a new arrangement is proposed to increase the probability to win the consensus process. For example, Ethereum evaluated the Proof-of-Stake algorithm. This consensus algorithm rewards (increase the vote probability) nodes that hold the higher stake. Thus, to increase the chance, different nodes gather their stakes in order to improve their chance.

It is important to highlight that the consensus study is a wide research field. This field is not limited by the mining schema, but it is extended to different algorithms, rewards mechanism, etc.

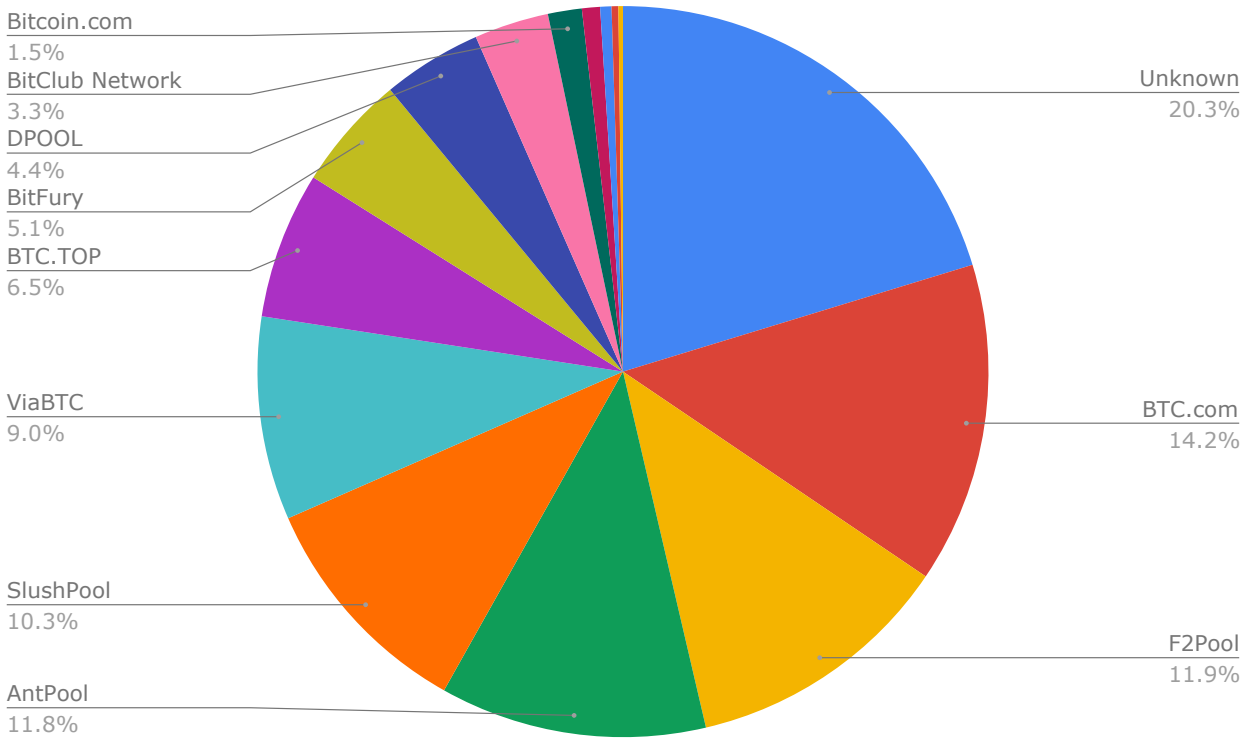

Figure 2.6 – Hashrate distribution among the largest mining pools (03/01/2019) [Blo19]

2.1.2 Smart Contracts

One of the main blockchain purposes is to keep the transaction between two parties, in a public and distributed ledger [Nak08]. However, after blockchain became a *de facto* technology, scripts were inserted in order to process and identify transactions (as presented before in the Bitcoin network), and Ethereum Network used the smart contract concept [CD16] to improve the blockchain usage. A smart contract consists of a piece of code

that can be executed (pretty much like a traditional programming language). However, this code will be stored in the blockchain and it runs in the nodes that are storing that blockchain. In a nutshell, a smart contract is a piece of code that is stored in the blockchain and presents the following properties: self-enforced, uniquely identified in the blockchain, and immutable. This contract is executed when some party sends a transaction taking as destination the smart contract address. Once it receives the call, all steps (operations) that were described in the contract are executed by any node in the network and produce a transaction of output.

Smart contract usage on the blockchain enables this technology to extends its functionality. The code definition capability that allows defining a behavior executed when the smart contract address is invoked, allows the blockchain to create an entirely new application model. Due to the value that smart contracts insert in the blockchain, to support this functionality is very important.

2.2 Blockchain Layers

As previously discussed, the blockchain is composed of different components. Among the components we could identify different options or implementations, *e.g*, the consensus in a blockchain could be a proof-of-work, proof-of-stake, proof-of-ownership, PBFT, etc. Based on the capability of choosing any of the components, a definition of different layers could help to understand the different concepts that could be used and how they impact a blockchain, we categorize blockchains into four layers as presented in Figure 2.7. Although we use the layers presented in Figure 2.7, some authors present different architecture layers [MAAN19].

The “Communication” layer represents how the nodes in the blockchain communicate and exchange information. This layer defines the communication protocols, P2P architectures, and network infrastructure used by a blockchain. It is important to note that this layer has interaction not only with the consensus algorithms, but also with how the data is distributed and how the applications are executed.

Additionally, the “Consensus” layer contains the process to validate the candidate blocks before inserting them into the ledger and broadcasting that to other peers. The consensus algorithm is required in the IoT context, especially for its characteristic, where the network is public, and there is no trust in the peers. Thus, it plays a crucial role in ensuring that each new block contains valid information, and any peer is able to verify the information in the blockchain. The unreliable peer environment, where a blockchain is being executed, could be considered in order to provide a solution to a common authentication problem, which is related to have a third party involved.

The “Data” layer presents how the information is structured in the blockchain. This layer specifies which are the adopted cryptography algorithms, how the data is stored, how

the access to this data is performed, and how the data is replicated. Additionally, there are some different approaches to the type of data that is stored in the blockchain. For example, Ethereum [Eth17] stores the current balance for all accounts, while Bitcoin [Nak08] maintains only the transactions between different users. The transaction organization in the Bitcoin blockchain is through a Merkle tree data structure. This is an example of definition performed in this blockchain level. During the consensus algorithm, some validations should be performed, most of them are related to the verification of signatures and hashes, and the accounts balance.

Moreover, there are different ways to use the blockchain. The "Application" layer defines the APIs for using the data from the blockchain. For example, there are different ways to access data, to use coins [Nak08], to generate tokens [FMMT18], to perform a distributed application [WLL+17], to use an identity management [LHH+18], and to execute smart contracts [ABC17].

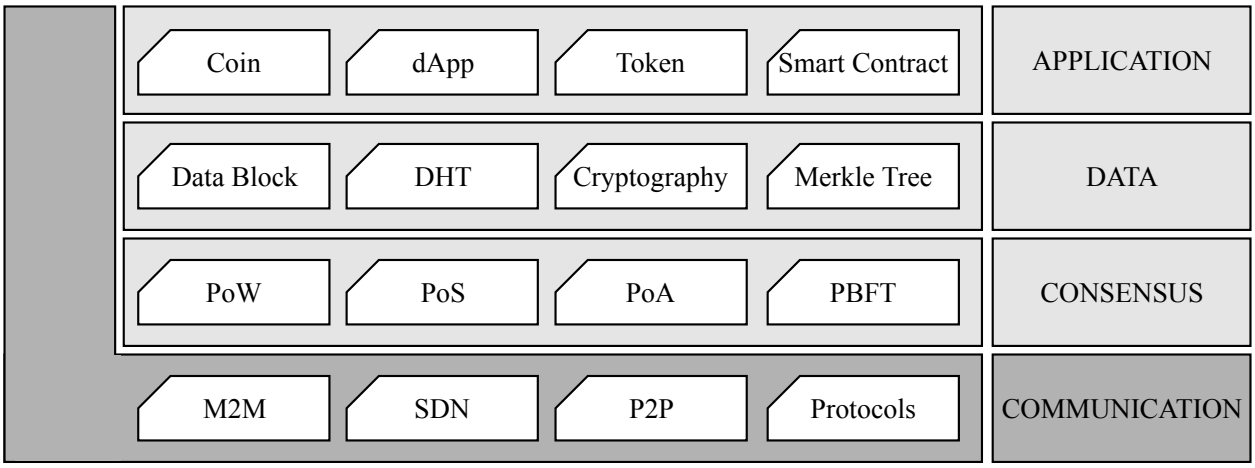

Figure 2.7 – Blockchain layers [ZNL+18]

2.3 Challenges of applying blockchain in IoT

The Internet of Things is a concept that has been employed and applied in different fields, as the "*thing*" term refers to anything. IoT can be applied to, for example, healthcare, home automation, environmental monitoring, transports, and industry.

The equipment, which is part of the IoT solutions, is composed of (usually) devices with lower processing power, for example, some of them are sensors that are able to gather information from the environment where they were deployed to, while some work as actuators that are responsible for interacting with, or modify the environment. All these devices are interconnected to work together. The devices setting is part of the solution architecture definition. This thesis follows the architecture presented by Gluhak *et al.* [GKN+11], which proposes an IoT testbed architecture.

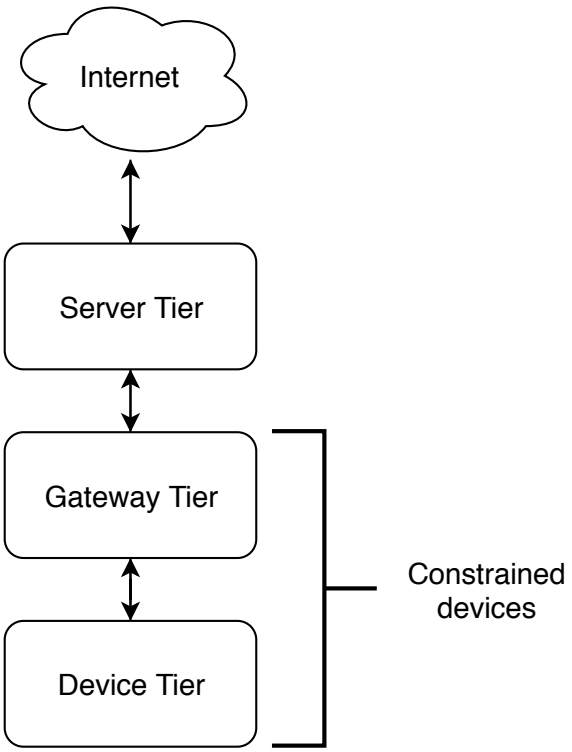

Figure 2.8 – IoT tiers architecture (adapted from [GKN+11])

The architecture applied for IoT solutions is divided into three different tiers, as shown in Figure 2.8. The first tier, called *Device Tier*, is composed of sensors and actuators. It is located at the edge of the architecture, being near to users, factories, cars, etc. *i.e.*, near to the environment where the solution will be acting. The second tier, called *Gateway Tier*, is composed of devices with more processing power than a Device Tier, however still with limited computing resources. This tier is responsible for providing some extra resources, allowing the solution to execute more complex tasks, or even to define a second connection level. The upper tier, called *Server Tier*, is composed of servers that are able to act as a service provider for the solution. Hence, the processing power is no longer a limitation, and it is able to interact with different gateways from the lower level.

Using a three-tier architecture allows the IoT solution to be conceived using heterogeneous hardware, in the same tier, or even distributed among tiers. It allows the gateway tier to act as a connection for the different devices from the lower tier. It will be responsible, also, for managing the produced data and to handle device access control [JGMP14].

Once the amount of data produced in the devices tier is notable, the gateway layer should work in order to handle the data. For example, a self-driven Google car, which is equipped with several different sensors types, is able to produce around 750 megabytes of data per second [Gro13]. Therefore, this brings about a problem related to the volume of data produced and how to find an efficient mechanism/solution to properly handle this data in terms of authentication (from which device is being produced) and security (data integrity and non-repudiation).

IoT is demanding more power processing and data handling at edge/fog level [SCZ+16]. Even with the cloud concept usage, there is still a need to act near to end-user in terms of data management. This need becomes real as some of these IoT applications might require short response time, handles private data, and some might produce a large amount of data [SCZ+16].

The blockchain technology adoption could be a challenge for IoT networks [CVM17]. As previously described, blockchain presents several benefits such as decentralization, resilience, data integrity, non-repudiation and relies on cryptography algorithms. Based on that, several different aspects could be enhanced in IoT solutions.

Due to its initial conception, blockchain was defined to work as a public ledger for cryptocurrency. Its applicability to the IoT context requires some extra research to evaluate different limitation and aspects. One of the first limitations is the hardware used to manage the blockchain. This brings the need to identify what is the minimum required hardware to manage a blockchain. To do so, it is important to take into account that big blockchains contain a large number of blocks and transactions, which will require more time to update peers, bandwidth and consequently, high resource demanding. Thus, a blockchain definition must be aware of these limitations and its architecture should allow controlling the block size as well to identify the cryptography algorithms.

The consensus algorithm commonly adopted for most blockchains, is the PoW, as described before. This process relies on node processing power. This algorithm is widely applied in blockchains in a way to trust in the network nodes. However, for the IoT domain, the PoW is not the best algorithm given the scenario, thus a further analysis in order to identify different algorithms, especially the ones that do not require high processing power.

A new data model applied for blockchain could, in part, help to improve the block size definition. This model especially helps when considering a structure to handle transactions and store them outside the blockchain.

3. RELATED WORK

The purpose of this chapter is to provide an overview of the state-of-the-art regarding existing commercial blockchains as well motivation and security needs, which could be supported by the blockchain applicability in IoT, smart cities and IIoT scenarios.

3.1 Blockchains

A large number of blockchains are being proposed after the Bitcoin blockchain proposal in 2008 [Nak08]. The blockchain development had increased especially after the beginning of 2013, which is when the Bitcoin currency value jumped from 10 USD to 600 USD in the middle of 2014 (a peak of 1,000 USD at the end of 2013). Due to the increase in the Bitcoin several other alternatives cryptocurrencies, and also blockchains, appeared in the market, for example, Litecoin, Namecoin, Peercoin, Monero, Dash, Dogecoin, Reddcoin, Vertcoin, Blackcoin, Feathercoin, Novacoin, etc. Currently[1],there are more than 2,000 different cryptocurrencies[2].

Table 3.1 – Blockchain comparison (data collected on 08/02/2019) [Bit19]

	Size (GB)	Length (blocks)	Access	Consensus	Usage
Bitcoin	237.32	562,189	Public	PoW	Monetary
Ethereum	178.93	7,194,465	Public/ Private	PoW	Transactions and Smart Contracts
Litecoin	21.88	1,576,133	Public	PoW	Monetary
Ripple	N/A	N/A	Public	Adapted PBFT	Monetary
BlackCoin	4.77	2,452,282	Public	PoS	Monetary
NameCoin	5.54	437,985	Public	PoW	Decentralized DNS
IOTA	8.8	111,808	Public	PoW	Monetary and information
Hyperledger	-	-	Permissioned	PBFT	Transactions Smart Contracts

The Bitcoin blockchain was the first blockchain proposed, and it is still one of the most stable implementations. That blockchain proposal is to support the Bitcoin cryptocurrency, this explains why most blockchain proposals are also supporting monetary application (online coins). However, it is important to notice that the blockchain technology is not limited

[1] February 08, 2019.
[2] https://coinmarketcap.com/all/views/all/

to be employed for the monetary domain, but it is extended to different domains. The Table 3.1 presents blockchains that are applied to different domains, such as Bitcoin, Litecoin, Ripple and Blackcoin are presented supporting monetary transactions, while Hyperledger, IOTA, NameCoin and Ethereum are examples of blockchain applied to a different purpose.

It is important to underline the blockchain coin application, as this was the technology genesis. It is defined as Blockchain 1.0, which is focused on cryptocurrency support (not limited to Bitcoin, but also Litcoin, Peercoin, etc.).

The blockchain evolved after that to Blockchain 2.0, in which the main milestone that could represent this change is the use of the smart contract concept in a blockchain. As described in Section 2.1.2, a smart contract is a piece of code that can be stored in the blockchain and can be executed by any node. The Ethereum blockchain is responsible for making smart contract popular, and for achieving a large number of users around the open source development community.

Smart contracts improve the blockchain usability and extend its functionality. This extension allows the evolution to Blockchain 3.0. The main change in Blockchain 3.0 is the development of DApps, *i.e.*, Decentralized Applications. These DApps present as the main characteristics its ability to use decentralized storage and communication. The Ethereum blockchain also acts as an enabler to improve the popularization of DApps.

Finally, Blockchain 4.0 is the new blockchain shift that is happening. This new proposal focuses on the blockchain applicability towards to Industry 4.0, which demands high trust and privacy aspects as well able to fit in an enterprise environment supporting different and legacy systems and applications. Scenarios such as health management institutions, supply chain management, IoT data collection, and many others.

3.2 Internet of Things

In the last few years, different solutions were proposed in the context of IoT networks. For example, some research focused on communication and management protocols [MSB+16], on distributed dissemination and processing of information [TMSB16], or on access control, especially on authentication and authorization, confidentiality, integrity and tamper-resistance [SRGCP15] [OMAA17] [AK14]. Although the solutions proposed by previous researchers presented some improvements to IoT networks, some open issues related to IoT security remain, for example, (*i*) use of existent protocols and services, or standardization of new ones, for security in IoT scenarios, especially for authentication; and, (*ii*) definition of architectures and models to ensure resilience and confidentiality through a heterogeneous environment. In this context, the blockchain technology presents a prominent solution that ensures confidentiality and resilience working as an authentication mechanism.

Huh *et al.* [HCK17] proposed a scenario using Ethereum and smart contracts to manage an IoT environment. After some experiments, some problems to run Ethereum on Raspberry Pi boards were discussed. The two major weakness of using Ethereum for IoT were: the time spent to update the blockchain (a problem related to the consensus algorithm) and the requirement of large storage size.

Ouaddah [OMAA17] research presented an evaluation considering the application of different access control mechanisms to the IoT context. The research considered criteria such as device heterogeneity, scalability and lightweight in order to identify the best solution for the IoT domain. The paper indicated as future direction the blockchain application in the IoT architecture to handle access management. Thus, proposing a lightweight consensus algorithm and blockchain storage strategy is crucial in order to apply the blockchain solution to the IoT context.

Dorri *et al.* [DKJ17] proposed a lightweight blockchain architecture for IoT as an authorization mechanism to access data in Smart Homes. Basically, the devices with limited hardware are more susceptible to attacks, specially to: *Denial of Service* (*DoS*), *Modification Attack*, *Dropping Attack*, and *Appending Attack*. In order to mitigate these problems, the use of overlays was proposed. In that environment, computers are used to maintain a blockchain with information about the devices. Although simulations point to a reduction on devices' processing overhead and on the number of packets on the network, it did not discuss how the devices are authenticated nor how limited power devices could be used in the environment. The transaction inclusion mechanism, presented by Dorri, also follows the structure defined by the Bitcoin blockchain, where transactions are grouped and added into a block.

Boudguiga *et al.* [BBG+17] research was focused on the employment of blockchain to ensure updated information about IoT devices data and availability. The paper also presents some questions about different scenarios in which IoT is used, such as Smart Homes, Smart Grids, Industry 4.0, and Intelligent Transportation Systems. In order to cover these scenarios, the research proposed the use of two distinct infrastructures: one for blockchain devices in a MultiChain architecture (Blockchain-as-a-Service) and another for IoT devices. Nevertheless, there was no experimental evaluation of the proposed solution.

Furthermore, some papers discussed security in different layers of an IoT context [MYAZ15] [MLZZ16] [ASW17]. Jing [JVW+14], *e.g.*, proposed a three-layer architecture (Perception, Transportation and Application) and discussed security issues and challenges in each layer. Nonetheless, a solution that considers hardware restriction in each layer was not presented.

The use of blockchain has been a prominent solution to solve security issues on IoT networks, as indicated by the previously mentioned related work and summarized in Table 3.2. However, they did not consider the access management in IoT Networks composed by devices with different capabilities. Moreover, few researchers evaluated the performance to use cryptography and blockchain in a single architecture.

Table 3.2 – IoT related work general information and security demand

Paper	BC	Comments
Abomhara [AK14]	-	Envision the IoT security need and points to access control mechanism
Ahemd [ASW17]	-	Evaluates security challenges for IoT architectures, highlighting the hardware limitation for new solutions
Jing [JVW+14]	-	Analyze security problems for IoT in a 3 tiers architecture, pointing as solution lightweight able to handle heterogeneous data
Ma [MLZZ16]	-	Proposes an IoT network which considers crowdsensing for sensors
Mahmoud [MYAZ15]	-	Proposes the 3 layers architecture for IoT, considering security in each layer
Schonwalder [MSB+16]	-	Proposes a protocol for IoT network monitoring
Sicari [SRGCP15]	-	Survey which points that a solution considering security for heterogeneous devices should be considered
Tortonesi [TMSB16]	-	SDN at IoT edge in order to handle information
Boudguiga [BBG+17]	X	Uses a blockchain to keep IoT devices firmware updated
Dorri [DKJ17]	X	Overlay based blockchain to manage the device produced data
Huh [HCK17]	X	Proposes to use Ethereum Smart Contracts in order to manage IoT devices
Ouaddah [OMAA17]	X	Evaluate the usage of access control for IoT, considering traditional mechanism, and points BC as possible future usage

3.3 Smart Cities

Extending the IoT scenario, smart cities are a big challenge in terms of a dynamic environment. This scenario also presents several problems related to the usage of IoT devices to create a smart city environment. Problems related to security aspects from IoT applied to the smart city, are presented by Wray [Wra19]. She especially highlights an attack against Atlanta local services in 2018, which was powered by ransomware called “SamSam”. Not only security properties are needed in a smart city, but also the scale of this environment is larger than a single smart home, which means that the scenario has to support a high number of transaction, followed by a considerable amount of nodes that are moving around the city.

In most smart city scenarios, Intelligent Transportation Systems (ITS) are frequently used to provide connected vehicles with information about the current traffic situation as well as about road and weather conditions. Furthermore, ITS enable/support beneficial functions such as path planning or mechanisms to warn road users about traffic jams, approaching emergency vehicles, and dangerous roadworks. To do so, vehicles and roadside infrastruc-

ture units (RSIs), such as traffic lights, need to exchange data using Vehicle-to-Infrastructure (V2I) communication. One key issue in such ITS applications is the privacy of the involved users [KBL17]. This privacy is defined as the capability of keeping users data anonymous such that they cannot be linked to their real identity.

To tackle this issue, researchers have proposed solutions [KBL17], [HKL17], [HTC13], [BM13] for classical ITS functions, which mitigate most of the known privacy issues. However, the discussed solutions do not tackle the challenges of future smart cities, like supporting a decentralized trust model where multiple entities (SPs, RSIs, and vehicles) act together to share information.

In order to identify the privacy issues related to smart vehicles, Bloom *et al.* [BTRB17] targeted people's perception of the data collected by self-driving cars. They conducted a study with 302 participants and showed that people are not aware of the extensive amount of data collected by these cars including GPS data, images, speed and how these data are shared and used. Once the participants were made aware of this fact, they expressed concerns about how the data could be manipulated or who could have access to that data. Their findings suggested that the associated privacy concerns could have a negative impact on the acceptance of autonomous driving vehicles.

Privacy in smart cities and vehicular networks is an issue discussed in several works [PBH+08], [LLL+12], [HCL04], [HIJ+12]. Papadimitratos [PBH+08], for example, proposed an architecture that relies on a certification authority (CA) that is responsible for the identity management in its own region (like a city, district, county, etc.). Thus, each region or city has its own CA, and region-to-region cross-certification is used to allow a vehicle to move from one CA to another. The centralization of the CA is a bottleneck and a single point of failure, which motivates the need for a decentralized solution. To address the privacy issue, the authors proposed the creation of a pool of pseudonyms that are assigned to vehicles when they move between regions.

Blockchain technology provides a decentralized and resilient solution through a Peer-to-Peer (P2P) network and ensures data integrity by employing a hash of the data stored in the blockchain. Li *et al.* [LLC+18] proposed a framework that relies on a blockchain and a cryptocurrency, called CreditCoin, in order to motivate users to share information. However, the network latency to produce information and to notify the RSI is not evaluated.

Sharma *et al.* [SMP17] defined a blockchain-based architecture that relies on a vehicle manufacturer or a transit department to issue and to revoke permissions for all vehicles. Vehicles act as regular nodes that produce information that is stored in the blockchain, and special miner nodes that are managed by the manufacturer/road transport authority are responsible for handling all requests/responses from regular nodes. The miner is the node responsible for creating new blocks containing the vehicles transactions and updating the peers.

Table 3.3 – Smart cities related work aspects comparison

	IOTA [IOT18]	Dorri [DKJ17]	Sharma [SMP17]	Li [LLC+18]	SpeedyChain
Time to add transact.	Minutes	ms∼seconds	N/A	>40 ms	<10 ms
Architect.	P2P	Overlays	Vehicles and miner nodes	RSU/OBU	RSI - Vehicles
Hardware	Node:PC; Wallet: Own/ RaspPi	N/A	N/A	Simulated	Emulated
Block	Immutable	Immutable	Immutable	Immutable	Appendable/ Decoupled
Main Usage	Payment M2M	Smart Home/ Smart Cities	Smart Cities	Smart Cities	Smart Cities
Key Manag.	One key pair per device	One key pair per device	One key pair per device	One key pair per device	Expiration time for each PubKey. Only one active

Dorri *et al.* [DSKJ17] proposed a blockchain-based framework to address the security and privacy of smart vehicles. In their paper, they discuss multiple use cases, including remote software updates and flexible automotive insurance schemes. However, their article does not provide a detailed technical discussion of their framework and neither do they propose solutions for addressing the issues that we focus on in this thesis.

Table 3.3 provides a comparative summary of the key aspects of relevant related work. As previously described, IOTA [IOT18] is a blockchain proposed to perform Machine-to-Machine (M2M) payments using IoT devices, however, it uses Proof-of-Work (PoW) consensus, which requires a considerable time to append new information. Consequently, the hardware required for a full node in IOTA is high and not compatible with IoT devices (*e.g.*, a PC with at least 2GB of memory is recommended). Focusing on limited hardware, Dorri *et al.* [DKJ17], [DSKJ17] proposed a blockchain-based solution that achieves low latency for adding information to the blockchain, however, the proposal does not take into account the dynamism and privacy concerns that are important in smart city scenarios. While Li *et al.* [LLC+18] and Sharma *et al.* [SMP17] presented solutions using blockchain for smart cities, their approach incurs long delays for adding and retrieving information to and from the blockchain.

3.4 Industrial Internet of Things

Recently, some researchers have referenced the use of blockchain in the context of IIoT to improve security [TR17] [WDWL18]. However, none of them presents a discussion about how the existent blockchains could be adapted in this context. Specifically, they do

not discuss the impact of known security attacks and neither do they explore the impact of consensus algorithms or parameters such as blockchain size or the time to insert a new block into the blockchain.

Li *et al.* [LSFK17] presented a discussion and possible solutions to use blockchain in IIoT. They proposed a solution focusing on the "Communication" layer ([ZNL+18]) using P2P architecture that uses a mechanism called satellite chains, which use validator nodes to share information between these chains. Also, they propose an integration with Hyperledger Fabric [Cac16]. However, they do not present an evaluation of the performance results, nor security analysis of the proposed solution. Consequently, it is hard to evaluate in which scenario their work could be applied to.

Boudguiga *et al.* [BBG+17] focused on the "Application" layer of the blockchain, employing blockchain to perform access control in the context of IoT. Moreover, they present a discussion about the application of their proposal in different scenarios in which IoT is used, such as Smart Homes, Smart Grids, and Industry 4.0. They also presented an infrastructure based in a Blockchain-as-a-Service (BaaS) that is able o improve the application performance. However, their paper does not present practical experiments to support the evaluation, nor the blockchain data management is considered in the research.

Focusing on the architecture of the "Communication" layer, Dorri *et al.* [DKJ17] propose a solution where overlays control the access to data stored in a blockchain shared among different overlays. In this architecture, an overlay has enough computing power to maintain a blockchain and IoT devices are not exposed to common attacks such as Distributed Denial of Service (DDoS) and Dropping Attack [JLG+14].

In a similar architecture, Dorri *et al.* [DKJ19] extend the research proposing a different solution that could act in the "Data" layer, where they introduced the idea of removing data from a blockchain. In his proposal, the capability of removing a transaction from a blockchain is only allowed for the user who generates the transaction, and is important to highlight that this removing process, do not erase the transaction register (if this were possible, the blockchain integrity would be compromised), but it removes the data, and replace the data for a footprint. This solution fits for scenarios where the devices present low storage, as the summarized transaction is considerably smaller than a regular transaction. This research proposal is focused in the "Data" layer, as the transaction structure should be changed in order to allow the removal operation.

Table 3.4 presents a summary of the main characteristics present by the research, as well in which blockchain level the proposed solution is focused on.

Table 3.4 – Industrial IoT related work summary

	Blockchain Level	Comments
[TR17]	Data, Consensus	New blockchain architecture, which relies in the smart contract providing functions for store and process information.
[WDWL18]	Data, Application	Using blockchain as dual factor authentication for devices, evaluated the BC overhead in the devices.
[LSFK17]	Communication	Used the satellite chains which are mainly controller by a regulator node, which presents a central entity.
[BBG+17]	Application	Presents a solution based in BaaS, which act to perform the access control for IoT context.
[DKJ17]	Communication	Usage of overlays which controls the data access.
[DKJ19]	Data	Proposes a data structure allows removing information from a transaction.

3.5 Chapter Summary

This chapter presented a comparison of the most relevant blockchains. Most of them are part of the infrastructure for cryptocurrencies. Despite that, blockchain has evolved, and nowadays, they are applied to different areas.

Blockchain has just recently been used in IoT context and still presents some challenges before it can be widely adopted. Several papers proposed solutions that could be applied to IoT, however there is a lack of real constrained hardware evaluation, as well as a real implementation that considers this type of hardware. For example, blockchain applied to the smart cities scenarios has just started. This type of environment can be considered an IoT scenario, nonetheless, its main characteristic is the dynamic behavior, constant changes applied to peers, as well as an increased scale in comparison with a smart home, for example. Hence, different challenges emerge in this kind of environment.

Additionally, to support different contexts, the blockchain should allow its components to change in a way that fits the domain context where the blockchain is applied to. Traditional blockchain block structure, once added to the chain, cannot be changed, thus, a data model where the blocks, after added to the chain, could include new data could be an interesting new improvement for blockchains.

The current thesis was motivated by multiple different types of research that were present throughout this work. In this chapter, it was presented the most significant researches that contributed to define the blockchain for IoT devices proposed as part of this thesis. However, the main contributions (which are directly connected to the solution proposed) to the design and implementation of this research are presented in Table 3.5. In one hand, Bitcoin [Nak08] and IOTA [IOT18] are two commercial blockchain implementations. While Bitcoin was the first blockchain developed, IOTA was the first commercial distributed

ledger designed to manage IoT devices. In the other hand, Jing [JVW+14] and Dorri [DKJ17] are academic researches that raise the security concerns for IoT devices, and the latter presented a blockchain to address the device access control using a blockchain instance.

Table 3.5 – Related work summary

Research	**Relation**
Bitcoin [Nak08]	Introduces the blockchain concept, working on a P2P network, relying on hash function to create an immutable link between blocks and asymmetric cryptography to enforce the non-repudiation.
IOTA [IOT18]	This commercial distributed ledger over a P2P network, present an alternative data structure to manage information.
Jing [JVW+14]	Identify the security threats for IoT considering a 3 layers architecture, and highlight the security requirements for IoT networks.
Dorri [DKJ17]	Uses the blockchain solution to manage data produced by IoT devices.

4. SECURITY ASPECTS

Currently, security aspects are vital for any computing solution or application. Security is a field that has increased attention and importance on the whole life cycle of new computing solutions. This attention increase is partially motivated by recent attacks and vulnerabilities exploitation in computing systems [Woo16], [KBOS18], [New18], [Jga16], [Wra19].

There are multiple ways of handling security when defining a new computer solution. The most common approach applied in software development is to integrate security as part of the software development life-cycle. In order to get the best in software development, the security must be included by design in the technology since its conception. Thus, this leads to merging security practices and standards since the early stages of technology definition. This is what happens with blockchain technology, which is a very new technology and there is still time to incorporate security in this technology.

The blockchain technology is in early stages of definitions, research and adoption by the industry. Consequently, security is fundamental to improve it. As this technology is heavily based on existing technologies, such as P2P, cryptography algorithms, etc. as described in Section 2.1, it is important to review the security aspects of each technology in order to protect the blockchain from known security threats.

In order to identify issues related to security on blockchains, the current research performed a literature review using relevant research databases to summarize the possible attacks that could affect the blockchain technology. It is important to highlight that blockchain is widely applied to solve security problems from different domains, however, there are few researches that considers the security issues of this technology itself.

Based on the results presented in Table 1.2, achieved through query string *blockchain and security*, the papers were evaluated in order to filter and keep only those that present blockchain security issues. Thus a new search was defined: *blockchain and vulnerabilities*, and this query was executed again. Table 4.1 presents the number of paper that was achieved after reading the paper abstract, and removing all those that do not presents a vulnerability that affects the blockchain technology directly.

Table 4.1 – Number of papers that considers blockchain vulnerabilities

Portal	**Number of papers**
IEEE	16
ACM	7
Springer	36

It is important to emphasize that most of the papers found during the searches, presents the blockchain being applied to solve security problems in different domain areas. Thus for this evaluation, they were removed, and only papers that present threats or vulner-

abilities related to the blockchain technology were selected, and a discussion related to the findings is available in section 4.1.

4.1 Blockchain Attacks

This section presents the most common attacks that could be executed against a blockchain. For this evaluation, any attacks that somehow affects the regular/expected blockchain behavior were considered.

Double Spending/Race Attack

Double Spending attack [KAC12] is one of the first known attacks that are executed in the blockchain. This attack was identified in fast payment application for Bitcoin. Usually, each block in the Bitcoin blockchain takes around 10 minutes to be persisted. In order to propose an alternative to allowing the Bitcoin usage in scenarios such as a fast-food restaurant, the fast payment relies on a single (or no) confirmation.

The attack consists of a malicious user sending multiple transactions to reachable peers in order to spend the same coin more than once. Figure 4.1 presents a scenario where malicious users, which own 1 Bitcoin in their wallet, create two different transactions expending the same amount of coins. One transaction has the victim address, and the other transaction has a wallet address controlled by the attacker. When the transactions are sent to the Bitcoin network in two different blocks (**b3** in Figure 4.1) leading the chain to create a fork. Thus, the attacker uses more processing power to increase the chain in which the double spending transaction is (**Fork Chain B**), leading to the honest transaction falling in the short branch. According to the fork resolution algorithm, the short branch will be discarded, and the transaction is sent back to the *mempool*. In this attack, the transaction now is conflicted to a previous transaction where the attacker transfers the coins to another wallet, thus leading to discarding the transaction. The main goal of this attack is related to spending the same coin more than once, thus it compromises the Consensus, Data and Application layers (as presented previously in Figure 2.7 in Chapter 2).

The main attack vector exploited in this vulnerability is the fork resolution, and it can become attractive as the attacker has a financial advantage when performing the transaction collision. It is important to highlight that this attack is not commonly applied in the IoT domains that do not use coins exchange.

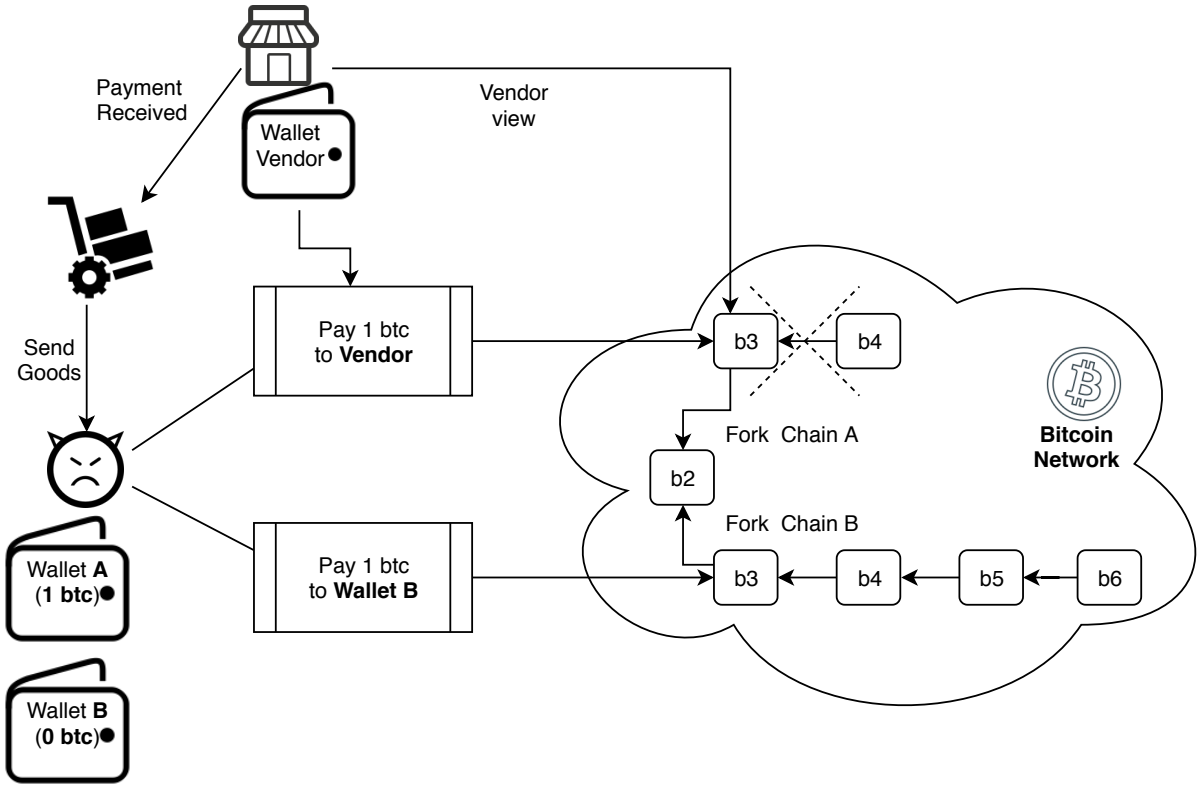

Figure 4.1 – Double Spending attack

Finney Attack

A variant from the Double Spending (with the same goal to spend twice the same coin) is called Finney attack [Fin11] and was presented by the Hal Finney user in the *Bitcointalk Forum*. In this attack, as shown in Figure 4.2, the malicious user creates a transaction Tx_A^B1 (Wallet A transfers 1 Bitcoin to Wallet B) and works in order to pre-mine (create a block privately). When this block is created, this malicious user can execute a transaction with an honest node (such as buying some product), creating a new transaction paying to the seller Tx_A^V1. As soon the seller transaction is mined into a new block (*b4*), the malicious users broadcast a pre-mined block (*b4'*), and waits for that block to be part of the longest branch length.

Thus this attack targets the same layers as the Double Spending operation. In blockchains in which the focus is in data handling, *i.e.*, there is no currency associated with, this attack is not profitable for malicious users.

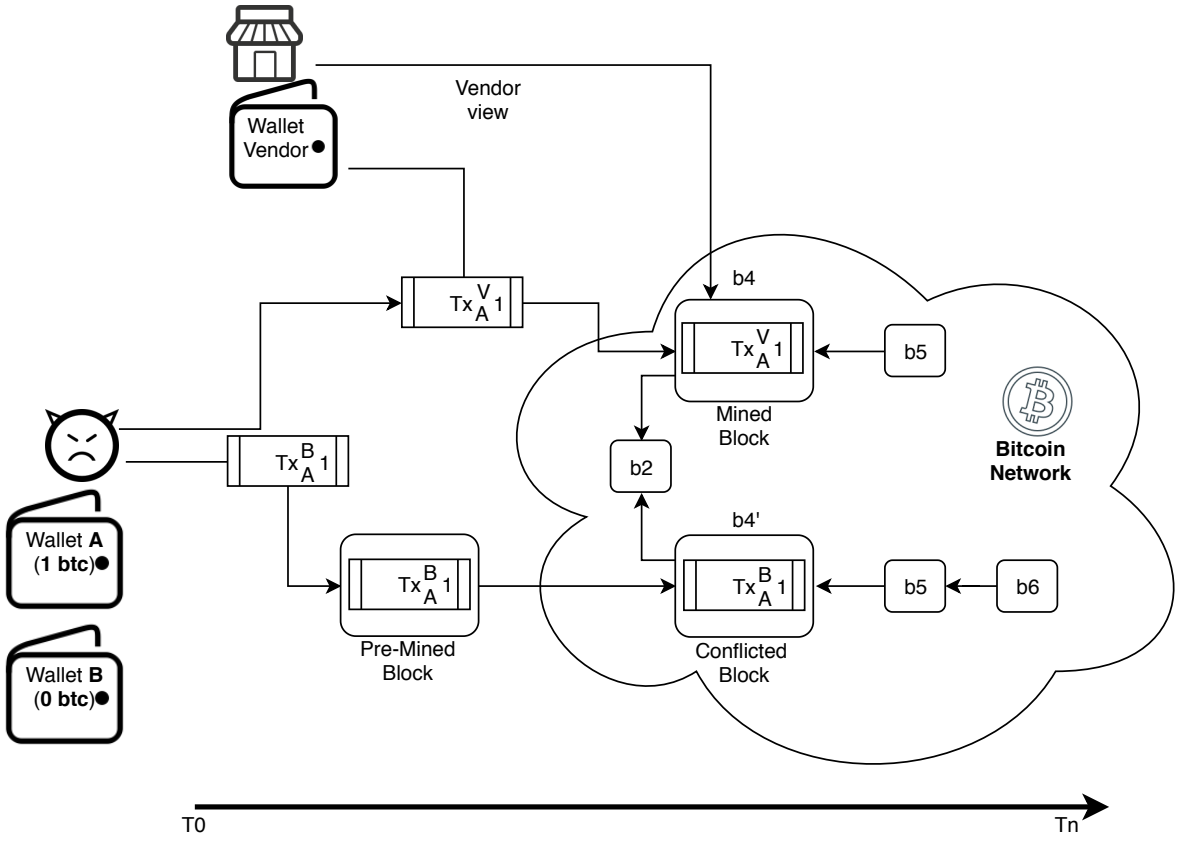

Figure 4.2 – Finney attack

Vector 76 Attack

The Vector 76 [Vec11] attack works similarly as the Double Spending and Finney attacks. However, the main target for these attacks is cryptocurrency exchanges. In this attack, shown in Figure 4.3, the malicious user pre-mines a block (*b4*) containing a transaction ($Tx_A^E 100$) in which the attacker performs a deposit in the exchange account. After that, this user creates a new block (*b5*) and with these two blocks (created privately) the attacker sends the blocks to the peers that are connected to the exchange. Thus, it will lead the main blockchain to a fork. However, before its fork resolution, the attacker sends a new transaction ($Tx_E^B 100$ mined in *b6*) to withdraw the money from the exchange. If the exchange receives the withdraw before the fork resolution, it will transfer the amount to an account B under the attacker control.

This attack affects mainly the Consensus and Application layers of the blockchain (as presented previously in Figure 2.7 in Chapter 2).

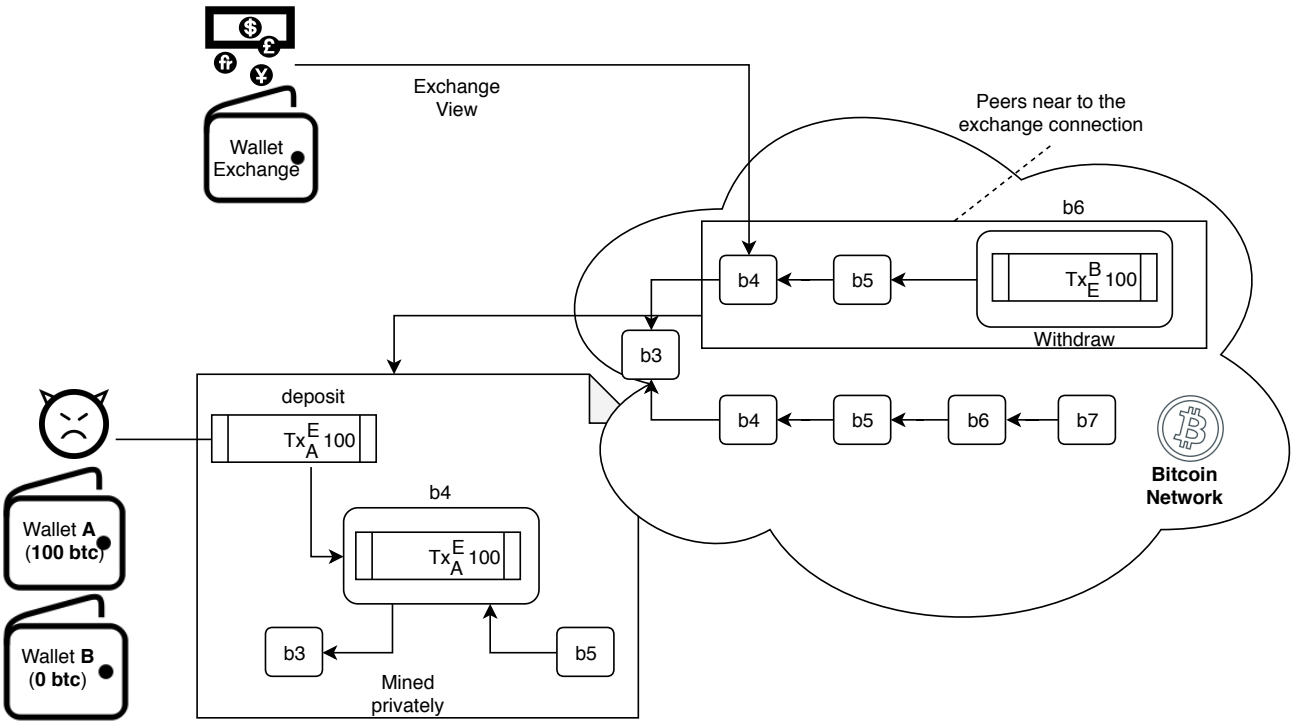

Figure 4.3 – Vector 76 attack

Alternative History Attack

This attack also exploits the Bitcoin blockchain fork resolution algorithm and uses the private mining technique.

In order to avoid attacks such as Double Spending and Finney attacks, the vendors only sell products after receiving confirmation from a number n of peers. Thus, the attacker, after sending a transaction ($Tx_A^V 1$), starts to mine privately keeping its own malicious blockchain fork, racing to keep the private copy longest than the main blockchain. Once the vendor sends the products, the attacker will publish the private blocks. To succeed in the attack, the malicious fork should be bigger than n blocks, and when the fork resolution algorithm act, the bigger chain will prevail, and the attack is completed.

However, the attacker should have more than 50% of the computing power of miners in the network. The PoW consensus algorithm and the fork resolution algorithm are the two factors that enable this attack to be executed, not only against a blockchain but also could affect IoT blockchains, which are traditionally hardware constrained. Consequently, it explores vulnerabilities in the Consensus layer to affect the Data layer of the blockchain.

51%/GoldFinger/Majority Attack

One of the most popular attacks for the Bitcoin blockchain is called 51% attack [GKW+16]. It basically consists of a malicious user controlling more than 50% of network processing power, thus this user can rewrite the network blocks according to their will. This attack in the alternative history attack ensures 100% success rate. This attack is focused on the Consensus layer, which is applicable for blockchain that is using the PoW consensus algorithm. Thus, for IoT security domain, the solution should not rely on any processing power mechanism, as the devices that compose the architecture are constrained in several different aspects such as, processing, memory, power, storage and network.

Selfish Mining Attack

A different approach of attack that focuses on the mining process and affects the Communication and Consensus layers is called Selfish Mining [ES14]. The main goal in this attack is to force honest miners to waste their mining effort, *i.e.*, honest miners will use computing power (and energy) without receiving rewards.

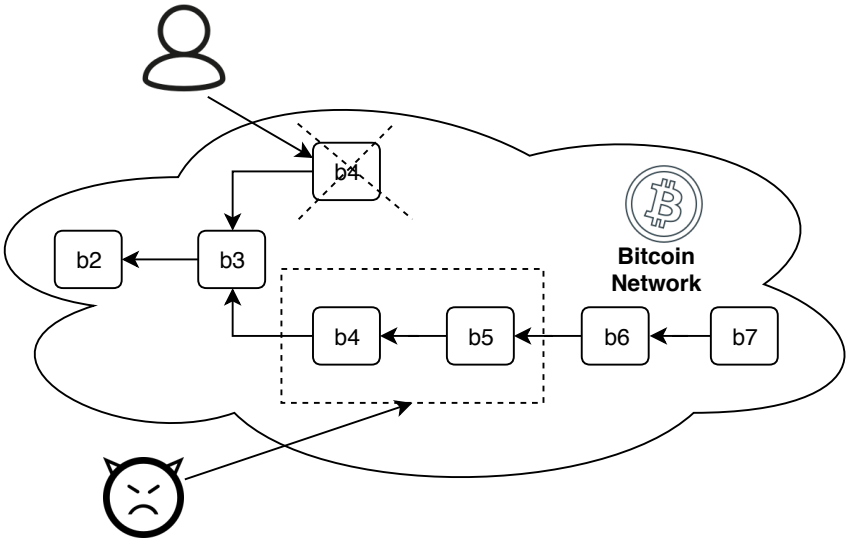

Figure 4.4 – Selfish mining attack

The attack consists of malicious users, or pool, mining privately at least two blocks (presented in Figure 4.4 as *b4* and *b5*). At the same time, these dishonest miners, keep checking the main network, and when they identify that a new block is included (*b4*) by an honest node, the malicious miners publish their two blocks (*b4* and *b5*). At this point, the fork resolution will switch to the longest chain leading to new blocks being added in this chain (*b6*

and *b7*). The honest node block (*b4*) is discarded, which will lead the honest miners to lose the processing to create the blocks.

Block-WithHolding Attack

The Block Withholding attack [Ros11] is focused on targeting mining pools. This attack explores the reward mechanism used in mining pools to define the amount of reward that each miner node will receive for the performed effort. The reward mechanism uses the partial PoW (PPoW), *i.e.*, the number of processed nonce and hashing mined before achieving the target hash for the block.

In order to perform the attack, a rogue miner, which is part of the pool, keeps sending partial PoW that is not able to solve the block puzzle. Through this behavior, it keeps injecting PPoW that are not able to solve a puzzle, and being rewarded for that. Any Full PoW (FPoW) that the node finds it discards.

In this case, the attack is executed using the consensus algorithm combined with the mining pool. As the mining process for constrained devices should be avoided, this attack is more common to blockchains that are rewarding the miners by its participation in solving mathematical puzzles.

Fork After WithHold Attack (FAW)

A variant from Block Withholding attack is called Fork After Withhold (FAW) attack [KKS+17]. The attack vector exploits the intentional fork creation, aiming to increase the profit by mining.

The attack flow is presented in Figure 4.5. The attack consists of a malicious miner joining a mining pool, splitting his processing power between an innocent miner and an infiltration miner. When this infiltrator finds an FPoW, they keep the block privately. As soon as this infiltrator identifies that other miner (which is not part of target pool) finds a block, they send the mined block, creating a fork in the blockchain. The attack becomes viable due to the PoW consensus algorithm and the fork mechanism.

Bribery Attack

A different approach to force a Double Spending is through a Bribery attack [Bon16]. The Bribery attack consists of malicious users temporarily obtaining the majority of mining

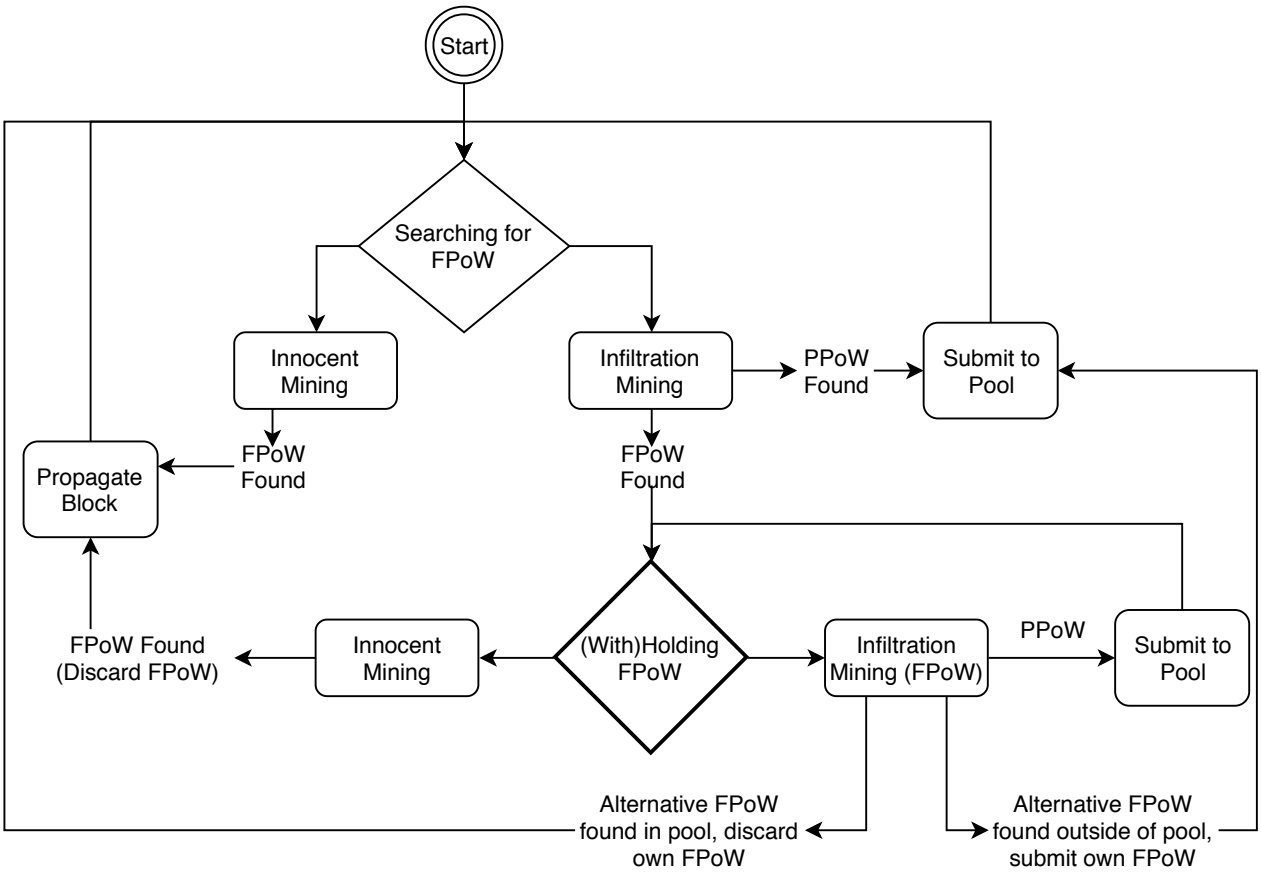

Figure 4.5 – Fork after withhold attack (Adapted from [Col18])

power renting computing resources, *e.g.*, virtual machines in a cloud service. As soon this power is achieved, the malicious user works to include a transaction in a block that will be mined.

Due to his majority in the processing power, this block is included in the blockchain, the user, using the majority of the network, creates a new block with a new transaction conflicting with the previous ones, and forces this new block to be included following the longest branch. It explores the fork resolution used by many blockchains that use PoW consensus algorithm, *e.g.*, Bitcoin and Ethereum.

Pool Hopping Attack

The Pool Hopping attack [Ros11] is focused on the profit that each pool is providing. In the attack scenario, malicious users participate in the mining process only when the possible reward for mining is high. When the malicious users identify that possible reward is low, they leave the mining pool to participate in other more attractive mining pools.

The attack succeeds when the miners identify that a victim pool is paying less than other pools. Thus, it leads miners to stop mining for the victim pool, and move to a profitable pool. This kind of attack only affects reward-based consensus algorithms.

Vulnerable Signature

The vulnerable signature attack was presented by Bos *et al.* [BHH+14]. In their research, they present that Bitcoin implementation presents poor signature randomness, which can allow a malicious user to steal money from a victim.

In order to prove the attack, they extracted 47,093,121 elliptic curve points from the signatures present in the Bitcoin blockchain. Based on those signatures, 158 unique public keys had used the same nonce value to generate more than one signature. This information allows to compute the user's private key and then compromise the Bitcoin wallet.

This attack points to a problem related to the cryptography algorithm implementation, thus a way to mitigate this vulnerability is to use implementations provided from known/good reputation security communities such as OWASP [OWA19] following NIST guidelines [NIS19].

Collision Attack

The hash function collision is possible but improbable. Nonetheless, Giechaskiel *et al.* [GCR16] evaluate its effect in case this attack becomes feasible.

As computer power is increasing, it is very likely that at some point in the future the known cryptography and hash algorithms become easy to attack. When this situation becomes a reality, *i.e.*, malicious users are able to find two different inputs that lead to the same output ($h(x) = m$ and $h(y) = m$), this attack could lead the Bitcoin blockchain to repudiate a performed payment as well as allows this malicious user to destroy coins.

Deanonymization

The identity/privacy is protected in a traditional public blockchain (*e.g.*, Bitcoin) through the asymmetric key mechanism. However, user privacy cannot be ensured only through this public key approach.

It has been shown that through the information gathered by the client connection, it is possible to track and to identify the client [BKP14], linking the public key to the IP ad-

dress where the transactions were generated. Once most information that is stored in the blockchain is public, the idea of keeping the privacy of the device that produced raises a discussion that should be considered, in order to identify the best strategy to classify which information should be exposed.

The most common vector exploited in this attack [BKP14] are the P2P network connection and the reuse of public keys (which are used to identify the wallet). The attack consists of 4 main steps to perform the deanonymization: the attacker retrieves a list of servers; from this list, the attacker selects a set of nodes from whom they want to reveal the identity; identify the entry nodes of clients connected to the network and finally, try to correlate the transactions appearing in the network with the set of entry nodes.

The identity management remains an open issue for blockchain systems, including for IoT. Thus, further research could be established in order to identify strategies and mitigation points for this attack. A very initial privacy-enhancing strategy is to avoid the reuse of the key. However, this solution brings a new problem, which is related to key management.

DDoS Attack

Attacks such as Distributed Denial of Service (DDoS) are widespread over the Internet. These attacks are focused on overloading a specific target in order to reduce (or deny) the capability to respond to legitimate requests [MZD14].

The blockchain usually is conceived to run in a P2P architecture, which means if a node is compromised through a DDoS, the network itself keeps working. Thus, in case of a DDoS being performed in the Bitcoin blockchain, it could reduce a mining pool capability, leading other pools/miners more likely to find a valid hash for a block [JLG+14].

This attack also needs attention in an IoT domain, as commonly there are devices with some hardware constraints (*e.g.*, energy, computing power, memory, storage, etc.). Therefore, it is easy/attractive for malicious users to perform a DDoS attack. Recently, compromising IoT nodes and exploiting them to launch DDoS attacks has also proven to be highly disruptive to the Internet [RBB17].

Transaction malleability

In 2014 an attack was responsible for suspending withdrawals from Mt. Gox Bitcoin exchange [Jef18]. This attack is called Transaction Malleability attack [ADMM15], which consists of the ability to duplicate transaction with a different identity to the two transactions.

This represents a problem once the Bitcoin public blockchain relies on this transaction identification to notify the coin exchange. This issue affects the blockchain Data layer.

This attack exploits a Bitcoin protocol flaw [LLZ+17]. It takes advantage of the algorithm applied to generate the transaction identifier. The transaction identifier is calculated through the transaction fields, such as input and output addresses, value, and cryptography signatures (as presented in Figure 2.4) and a hash function.

The malleability attack happens when a malicious user changes the ScriptSig field, which leads to creating a different transaction identifier. The attack will succeed when there is no transaction confirmation from other peers because when the transaction is confirmed, it becomes immutable, and the conflicting transaction should be discarded [LLZ+17].

Different approaches were proposed to solve this type of attack [LLZ+17] [ADMM13] [RAH+15]. Some propose a scheme that combines the transaction hash with the address balance in order to confirm and complete the transaction [LLZ+17]. Some define a protocol that should run on the Bitcoin network to eliminate the part of the transaction that could create the attack [ADMM13]. Last, some propose a form to eliminate the input script from the transaction hash calculus [RAH+15]. Despite this vulnerability and the proposed solutions, it is important to highlight that the attack does not affect or allow the attacker to change the amount of Bitcoins being exchanged or even change the user address (input or output).

Delay Routing Attack

The main goal of this attack is to cause a delay to deliver a block to a victim [AZV17]. During this attack, the malicious user causes a slow block propagation towards or from a set of nodes. This delay could reach 20 minutes without being detected. Thus, during this time, the victim is completely unaware of the most recent t executed transactions in the Bitcoin network.

The delay attack scenario is presented in Figure 4.6. Step 1: Node A and B send the same block to the victim (C); Step 2: Node C requests the block from node A. The attacker changes the content triggering the delivery of an older block from node A; Step 3: The older block is delivered; Step 4: Before 20 minutes after the original block request made by node C, the attacker triggers its delivery by modifying another message originated by C; Step 5: the block is delivered before 20 minutes timeout. The victim does not disconnect from node A [AZV17].

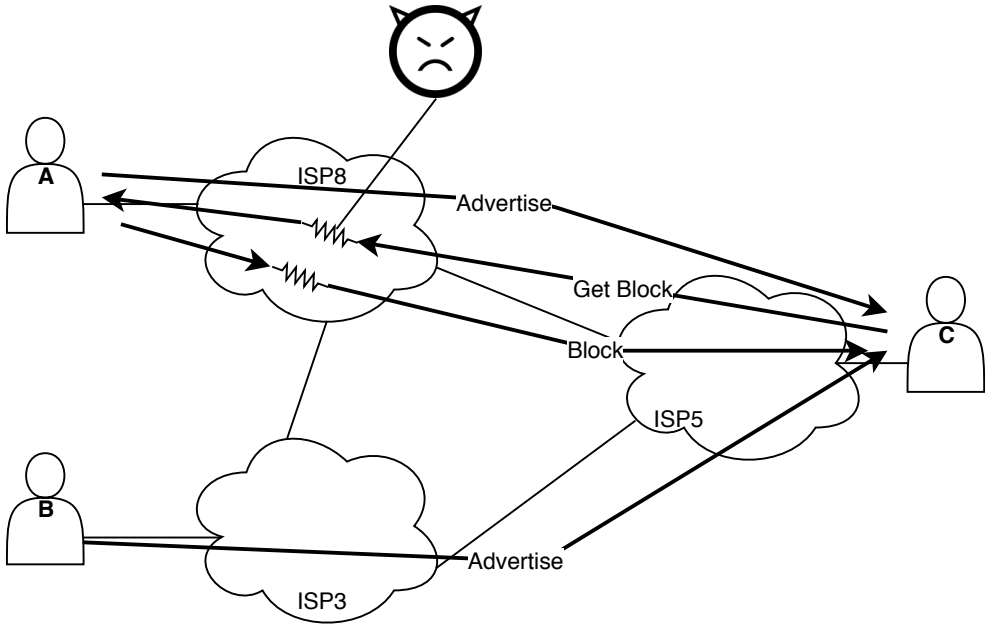

Figure 4.6 – AS8 acting as a rogue user intercepting part of the traffic and delay the delivery of a block for 20 minutes to a victim C (adapted from [AZV17])

Sybil Attack

The Sybil attack is possible due to the public P2P network [Dou02]. This attack is performed by a group of compromised nodes, which act together in order to compromise part of the Bitcoin network. Figure 4.7 presents the scenario where a victim node is connected to six malicious nodes. Thus this victim node will be influenced by the Sybil nodes (which are the majority) despite any information received by the honest node.

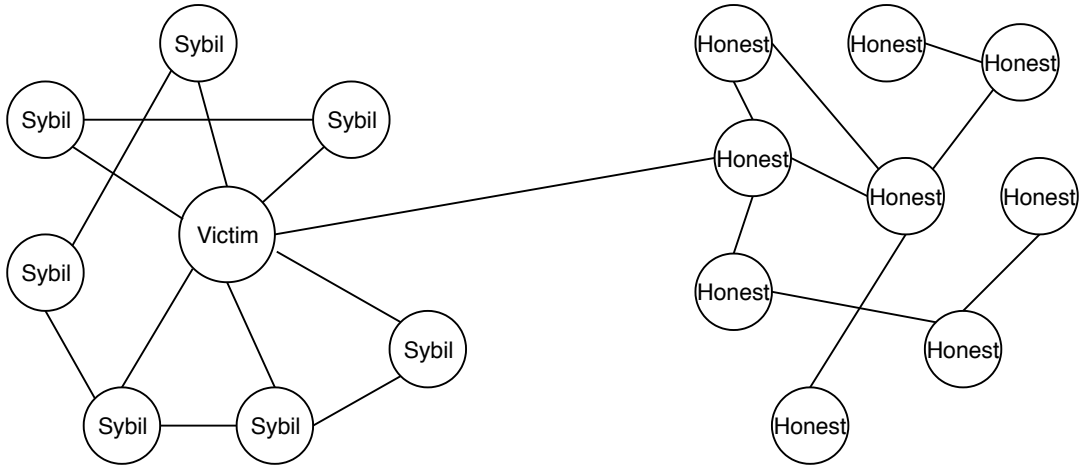

Figure 4.7 – Sybil attack

In this attack, the malicious users create/control multiple nodes/identities over the network, once a victim is connected to them, the attacker is able to influence the victim by controlling which block the target node will receive, as well as to hold blocks and transactions

performed by the victim. The main goal in this attack is to influence part of network nodes to act according to the Sybil behavior. This attack is performed in the Communication layer and depending on the Consensus layer could lead to different results.

Eclipse Attack

While the Sybil attack is focused on compromising the Bitcoin network, the Eclipse attack focus in compromising a single node. The Eclipse attack, proposed by Heilman [HKZG15], consists of a malicious user to monopolize a victim incoming and outgoing connections. Once this victim is isolated from the main blockchain network, the attacker can, for example, force the victim to waste computing power calculating old blocks hashes.

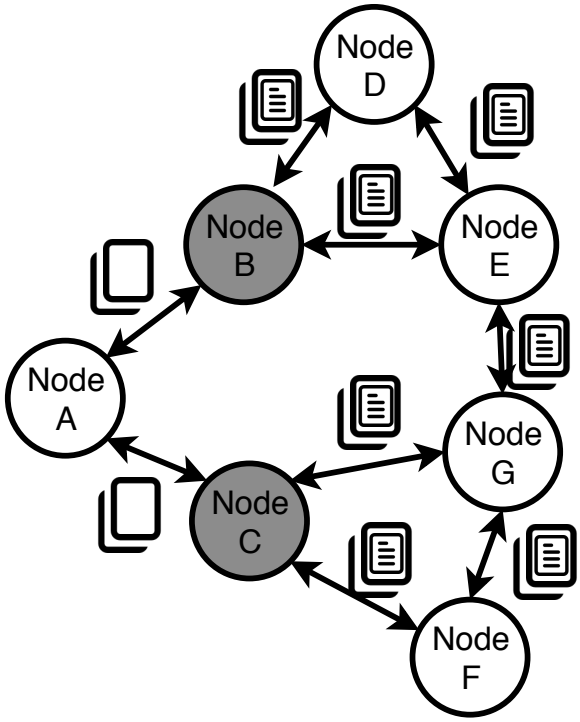

Figure 4.8 – Eclipse attack

Figure 4.8 presents an Eclipse attack scenario where node A is eclipsed by the malicious nodes B and C, thus leading node A to receive blocks and information filtered or even tempered by the malicious nodes. This attack also enables Double Spending and Selfish mining attacks. The attack is performed in the communication layer.

Timejacking Attack

Boverman [Bov18] presented the timejacking attack, and it exploits the block timestamp algorithm in the Bitcoin blockchain network.

Each Bitcoin node maintains internally a counter that represents the network time. This is based on the median time of a node's peers which is sent in the version message when peers connect. The network time counter reverts to the system time however if the median time differs by more than 70 minutes from the system time [Bov18].

The attack takes places when a dishonest user, connects multiple peers and reports inaccurate timestamps, which could slow down or speed up the node's network time counter. In case of this attack, an adversary speeds up the clocks of a majority of network mining resources while slowing down the target's clock. Based on the Bitcoin blockchain algorithm, the difference between the compromised node and the honest nodes could reach 140 minutes [Bov18].

This attack could be used jointly with the Eclipse attack and enables the dishonest user to perform a double spending attack, as soon as the attacker is able to exhaust the processing power and slow down the transaction confirmation rate.

Refund Attack

The refund attack is directed related to the Bitcoin payment protocol BIP70, which is used by Coinbase and BitPay [MSH17]. MacCorry's research [MSH17] presents two different refund attack: Silkroad Trader, which exploits an authentication vulnerability in the payment protocol, and Marketplace Trader, which exploits the refund policies of existing payment processors.

The Silkroad Trader attack relies on a vulnerability that the customer can authenticate that messages originate from the merchant, but not vice-versa. Through this attack, a customer can route payments to an illicit trader via a merchant and then deny the involvement [MSH17].

The Marketplace Trader attack aims to compromise the refund policies of Coinbase and BitPay. Both systems accept the refund address over e-mail, allowing a malicious trader to use the reputation of a trusted merchant to persuade customers to fall victim to a phishing attack [MSH17].

Balance Attack

The balance attack targets to the PoW algorithm [NG16]. The attack consists of delaying network communications between multiple subgroups of nodes with balanced mining power.

Natoli *et al.* [NG16] research focused on evaluating the trade-off between how many mining power is needed (considering the network delay) to send fake information, thus leading to enables the double-spending attack. The evaluation was performed in the Ethereum blockchain network.

Other Vulnerabilities

Despite all identified attacks against the blockchain technology, some other attack vectors could be exploited. Among all other attack vectors, we list issues related to the blockchain virtual machine, such as Ethereum Virtual Machine [HSR+18]. Also issues related to the bad codding when implementing a Smart Contract [ABC17], which is executed in the blockchain, however for this researchs this issue is part of a different category of threats.

In the same way, attacks resulting in bugs found in the implementation or related to bad practice by the users in private key or even wallet handling are known issues. However, it is not part of the scope in this research to identify all possible bugs and bad practices.

4.2 Chapter summary

This chapter presented a discussion on known attacks that could be performed on blockchains. As a new technology blockchain also presents some known security issues *e.g.* DDoS, Eclipse, Sybil attacks. Furthermore, it presents new possible scenarios and threats such as Double Spending, 51%, Transaction Malleability, etc.

The attacks were classified according to the blockchain layer model defined in Figure 2.7. This classification was performed according to the target component and blockchain module compromised. Table 4.2 summarizes each attack characteristics and the corresponding blockchain layer. As can be seen in the table, most attacks focus on the consensus and communication layers.

Finally, security evaluation is an important step to be considered for using blockchain in new solutions, despite the domain where the solution will be applied.

Table 4.2 – Blockchain vulnerability summary

Threat	**Layer**	**Characteristics**
51%	Consensus	Focused on the consensus algorithm that relies on processing power.
Alternative History	Application Data Consensus	Focused on the consensus fork and confirmation algorithm, aiming to generate a longer chain to generate a conflict.
Balance	Consensus Communication	Focused on the consensus and introducing a delay in the network.
Block WithHolding	Consensus	Focused on the consensus algorithm and the mining pools reward mechanism.
Bribery	Consensus	Focused on compromise the consensus algorithm through renting computer power.
Collision	Application Data	Focused on compromise the hash algorithm.
DDoS	Communication	Focused on compromising a peer in the network, make it unavailable.
Deanony-mization	Data Communication	Focused on the link the blockchain address with a user.
Delay Routing	Communication	Focused on compromise the information propagation in the network, lead a peer to an out-of-date state.
Double Spending	Application Data Consensus	Focused on the consensus fork algorithm, aiming to create conflicting blocks.
Eclipse	Communication	Focused on control the connections of a victim to manipulate the information sent and received.
Finney	Application Data Consensus	Focused on the consensus fork algorithm, pre-minig blocks to generate conflict.
Fork After WithHolding	Consensus	Focused on the consensus algorithm and the mining pool definition combined with fork algorithm.
Pool Hopping	Application Consensus	Focused on compromise the consensus and aim to mining pool to decrease the revenue.
Refund	Application	Focused on Bitcoin network payment protocol, to increase the refund.
Selfish Mining	Consensus Communication	Focused on the consensus algorithm and the aims to mining pools.
Sybil	Consensus Communication	Focused on control different peers in the network and use these peers to influence the network behaviour.
Timejacking	Application Communication	Focused on manipulating peer timestamp.
Transaction Malleability	Data	Focused on compromise the generated transaction creating a collision.
Vector 76	Application Data Consensus	Focused on the consensus confirmation algorithm, aiming to generate block conflict against cryptocurrencies exchanges companies.
Vulnerable Signature	Application Data	Focused on compromise the cryptography algorithms applied.

5. SPEEDYCHAIN FRAMEWORK DEFINITION

This chapter presents the blockchain definition in order to attend the device constraints and to ensure security. The proposed solution, called SpeedyChain, which is a permissioned blockchain, specifies a modified data model that allows transactions to be appended in different blocks at the same time. As part of the definition, the IoT devices should be structured following a three-tier architecture, which allows SpeedyChain to run in a gateway level (gateways are the components responsible for managing different devices), able to receive and to handle the data produced from different devices in different levels. The main operations will be described in this chapter.

The solution architecture is presented in Figure 5.1, and this solution is composed of devices that are responsible for generating or receiving information (usually constrained hardware). These devices are connected to gateways that are responsible for managing the information through the blockchain usage. Gateways are responsible for adding new blocks to the blockchain and appending new transactions into existing blocks.

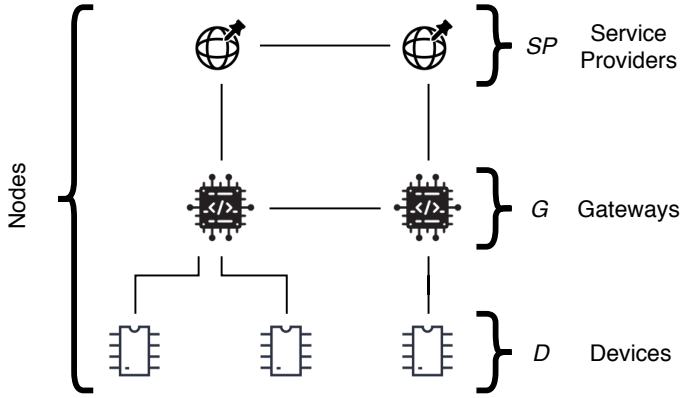

Figure 5.1 – Architecture components

Each device that takes part in the solution is represented by one block in the blockchain, *i.e.*, information produced by one device will be stored in the block associated with that device. The information validation is performed at the gateway level, and once the data is valid, the gateway is responsible for appending this information in the device's block, and for sending that information to the neighbouring gateways.

5.1 Architecture

Let $N = \{N_1, ..., N_n\}$ be the set of n nodes in the system with public-private key pairs (NPK_i, NSK_i). Also, consider that these nodes can have different roles in the architecture. Consequently, this system is composed by d devices, where $D = \{D_1, ..., D_d\}$, that usually

produce information and could be controlled remotely; g gateways, where $G = \{G_1, ..., G_g\}$, that manage the access to information in a blockchain; not limited to this, different kind of nodes are supported such as s service providers $SP = \{SP_1, ..., SP_s\}$. Therefore, $N_i = \{D, G, SP\}$. Assume that all nodes in N can use the same cryptography algorithms. Moreover, every NPK_i should be different and accessible by any participant in this system. Also, assume that a key pair (public and secret keys) from a device will be represented as (DPK_j, DSK_j) and a key pair from gateway will be represented as (GPK_h, GSK_h). Consider that each device in D (Devices Level) should be connected to a gateway in G (Gateway Level) through different (wired or wireless) network devices (Network Level). Additionally, the gateways are responsible to manage the device access and provide an API that allows to manage the blockchain. A generic IoT architecture containing all nodes, connections and its structure is presented in Figure 5.1.

5.2 Blockchain Definition

Based on the architecture presented in the Figure 5.1, the blockchain will be maintained by gateways in G. To ensure that every participant can access any NPK_i (*e.g.*, DPK_j or GPK_h) and information stored in a Gateway was not tampered with, let a blockchain $B = \{B_1, ..., B_b\}$ be a set of b blocks. Each B_k has a pair of different information (BH_k, BL_k), where BH_k is responsible to maintain the block header of B_k and the BL_k stores the block ledger, *i.e.*, the set of transactions of B_k as shown in details in Figure 5.2.

Therefore, BH_k is composed by ($HashBH_{k-1}$, k, Exp_k, $Time_k$, Pol_k, NPK_i), where

$$HashBH_{\text{k-1}} = \begin{cases} 0 & \text{, when } k = 1 \\ \text{hash digest of } BH_{k-1} & \text{, when } k \geq 2 \end{cases}$$

where hash digest is obtained through a hash function, *i.e.*, $HashBH_{k-1}$ contains the hash digest of previous block header (or zero when it is the first block); k is equal to the index of the block B_k in the blockchain; $Time_k$ is the timestamp from when the block was generated; Exp_k presents the threshold time to insert a new transaction in its block ledger, for example, after this time a device should create a new key pair (NPK, NSK) and submit a new block; Pol_k presents the access policy that the device has to attend; and NPK_j is the node public key. It is important to mention that every node - independent of its type - should have a block in B, composed of at least a block header, and every NPK should be available in the blockchain.

Let BL_k=$\{T_1, ..., T_t\}$ be the set of t transactions on the block ledger of the block B_k. T_m is composed by ($HashT_{m-1}$, m, $SigG_m$, $Info_m$), where

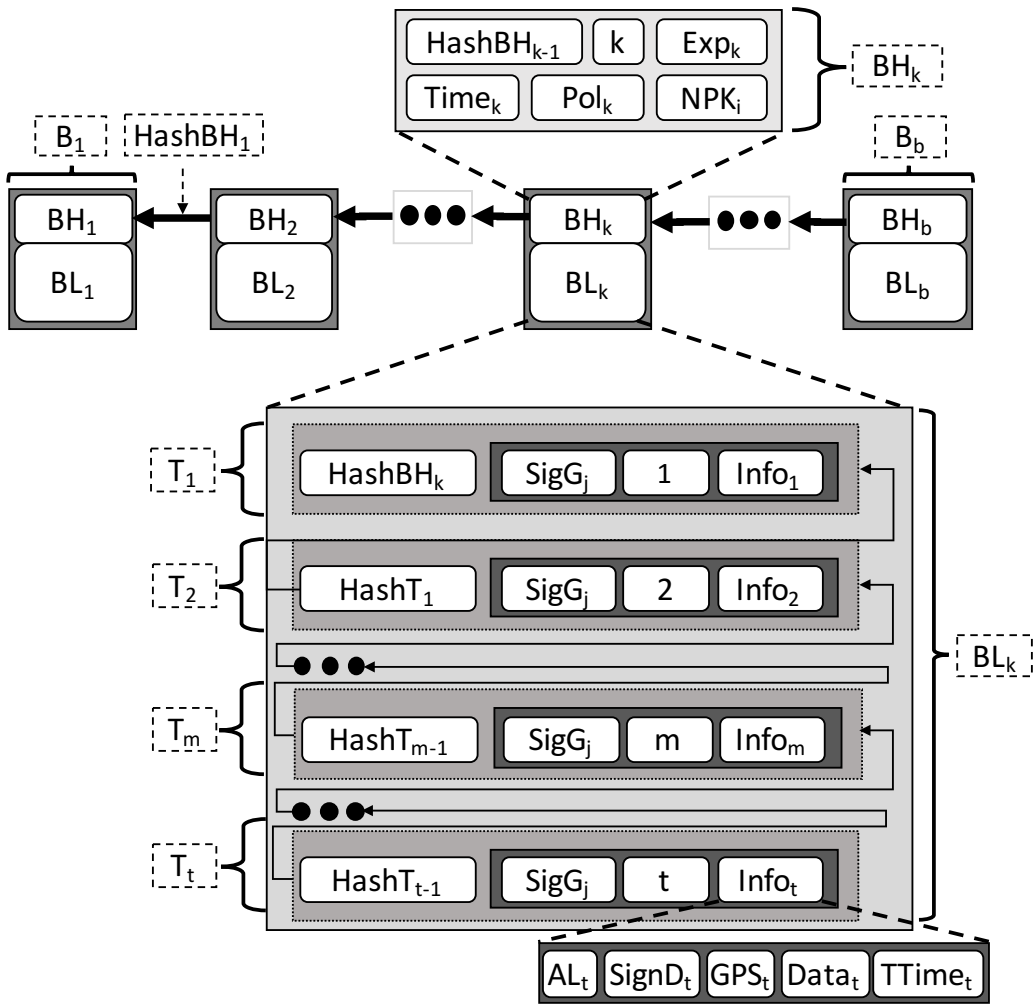

Figure 5.2 – SpeedyChain components

$$HashT_{m-1} = \begin{cases} \text{hash digest of } BH_k & \text{, when } m = 1 \\ \text{hash digest of } T_{m-1} & \text{, when } m \geq 2 \end{cases}$$

where the $HashT_{m-1}$ contains the hash of the previous transaction (or the hash of its block header when it is the first transaction of the block ledger); m is equal to the index of the transaction T_m in the block ledger BL_k, so $IT_m=m$; $SigG_m$ represents the result of the cryptography using the GPK_h to sign $Info_m$, used to public verification.

The $Info_m$ can be different for each type of node. Devices provide a set of information ($SigD_m$, AL_m,GPS_m, $Data_m$, $TTime_m$), where AL_m is the access level required to access the information from outside of the blockchain that is defined by the device D_j, while the $SigD_m$ represents the signature of (AL_m, GPS_m, $Data_m$, and $TTime_m$) using DPK_j, where GPS_m represents the global position of the device (when it is available), while $Data_m$ is the data collected/set from/to device D_j and $TTime_m$ is the timestamp when the $Data_m$ was generated/set. It is important to note that $Data_m$ could be formatted differently depending on the device. For example, it could store a single read of a sensor (an integer type) or a set of information, encrypted or not, depending on the configuration established in the API level. It

is important to highlight that to ensure the information ownership, the $SigD_m$ field, must be present in the $Info_t$.

5.3 SpeedyChain Main Operations

The main operations performed in SpeedyChain are: appending blocks, appending transactions, key update and consensus algorithm.

The appending blocks operation follows the structure presented in Bitcoin blockchain by appending new blocks at the end of the chain. The significant contribution presented in SpeedyChain is the appendable block concept, which takes places by appending transactions into a block. While traditional blockchains are focused on grouping the transactions from a mempool and thus creating a new block, SpeedyChain process the transactions as they are received. Each transaction after being validated is appended into an existing block, providing a unique property of appendable blocks. Additionally, SpeedyChain supports a key update mechanism that defines a time while a block accepts new transactions; once this limit is reached, it enforces the devices to update their keys. Finally, SpeedyChain was designed in a modular architecture, thus allowing the seamless application of any consensus implementation.

These operations are detailed in the next subsections.

5.3.1 Appending blocks

Insertion of a new block B_k in blockchain B is started by a gateway (present in Gateway level) with the objective to include a new node (N_i) public key (NPK_i). This algorithm is performed every time that a node N_i requests a connection and its Public Key (NPK_i) is not present in the blockchain (line 1 in Algorithm 5.1).

After verifying that an NPK_i is not present in the blockchain, the gateway should send this new public key to perform a consensus to insert the new block (line 2). It is important to note that the consensus is performed by a leader elected in the blockchain (see Section 5.3.4).

Algorithm 5.1 Insertion of new blocks in SpeedyChain

Require: Connection request and requester NPK_i
1: **if** NPK_i is not present in any BH_j **then**
2: **sendBlockToConsensus**(NPK_i)
3: **end if**

5.3.2 Appending transaction

Every time a node N_i produces new information $Info_m$ to be inserted in the blockchain, it has to communicate to a gateway to append the transaction to its block ledger BL_i. This operation is performed only if the node public key (NPK_i) is present in a block header BH_i from blockchain B (line 1 in Algorithm 5.2). When, a gateway receives a new information $Info_m$, the digital signature $SigD_m$ present in $Info_m$ should be validated (lines 2 and 3) using the public key NPK_i.

Algorithm 5.2 Appending new transactions into the block ledger

Require: $Info_m$ and device NPK_i

```
 1: if NPK_i is present in any BH_j then
 2:     result ← verifySign(NPK_i, Info_m)
 3:     if result is true then
 4:         b ← blockIndex(B, NPK_i)
 5:         t ← lastTransaction(BL_b)
 6:         HashT_m-1 ← hash(T_t)
 7:         m ← t + 1
 8:         SigG_m ← sign(GSK_h, Info_m)
 9:         T_m ← {HashT_m-1, m, SigG_m, Info_m}
10:         broadcast(T_m, BH_b)
11:     end if
12: end if
```

After the validation of the signature, the gateway performs the encapsulation of the new transaction, setting: the hash of the previous transaction $HashT_{m-1}$ (line 6), the index of the transaction (based on the last transaction) m (line 7), and the digital signature from the gateway that is processing the transaction $SigG_m$ (line 8) using its secret key GSK_h.

After that, the gateway creates the new transaction T_m (line 9), and the transaction can be broadcast to the other gateways (line 10).

5.3.3 Key Update

Anytime that a gateway receives a transaction with its timestamp $TTime_m$ with a higher value than the expiration time present in the origin node N_i expiration time Exp_k the gateway will proceed a with key update algorithm (Algorithm 5.3). Also, the node N_i can send to the gateway a request to update its public key NPK_i'.

In both situations, a gateway will request to node N_i, its new public key NPK_i' (line 1 in the Algorithm 5.3). After the key validation (*e.g.*, if the key is not already in the blockchain),

the gateway will append a new block into the blockchain with the new NPK_i' from node N_i (line 3).

In order to append a new block, a gateway will use Algorithm 5.1 presented previously. Consequently, each node will receive a new block with the new public key NPK_i' of the node N_i.

Algorithm 5.3 Algorithm for key update

Require: $TTime_m \geq Exp_k$ or requested by node N_i
1: $NPK_i' \leftarrow$ **requestNewKey**(NPK_i)
2: **if** NPK_i' is valid **then**
3: **appendBlock**(NPK_i') {see Algorithm 5.1}
4: **end if**

5.3.4 Consensus

SpeedyChain was improved to allow the adoption of different Consensus algorithms. Before discussing different consensus algorithms, first we need to present what is a valid block or transaction. For a transaction to be considered valid, it should have a NPK_i that is already in the blockchain, a valid signature (based on the data transmitted and NPK_i), and a $TTime_m$ lower than its Exp_k (present in the block header) to ensure that no transactions are inserted in an expired block. Moreover, to ensure that a block header is valid: (*i*) the gateways should agree that a new node NPK_i can be part of the blockchain B; (*ii*) the access policy Pol_k for this node NPK_i should be defined; (*iii*) the Exp_k should be calculated to avoid a large block in size. We assume that this validation is performed by the gateways through predefined rules, e.g, signature validation.

Currently, there are different consensus algorithms used by blockchains, such as Proof-of-Work (PoW), Proof-of-Stake (PoS), Proof-of-Space, Proof-of-Burn and Practical Byzantine Fault-Tolerance (PBFT). Furthermore, it is not possible to define a single solution that will perform better than others for any scenario.

Two consensus algorithms for block insertion are available - but not limited to them - in the proposed solution: (*i*) validation based on a specific number of witnesses, where every block should be signed by at least a predefined number of witnesses; and (*ii*) adapted PBFT algorithm, where more than 2/3 of the active gateways should validate and sign the block. Both consensus algorithms could be summarized in Algorithm 5.4.

In order to encapsulate the new block B_k, every information from the block header BH_k is set, such as the hash of the previous block header BH_b (line 2), block index k (line 3), the timestamp using the time of block creation $Time_k$ (line 4), an expiration time Exp_k to control the validity of the block (line 5), and the access policy Pol_k that the new node is submitted to (line 6 in Algorithm 5.4). It is important to note that both Exp_k and Pol_k are defined in API

Algorithm 5.4 Generic consensus algorithm

Require: receive a NPK_i to perform consensus

1: $b \leftarrow$ **lastIndex**(B)
2: $HashBH_{k\text{-}1} \leftarrow$ **hash**(BH_b)
3: $k \leftarrow b + 1$
4: $Time_k \leftarrow$ **getTime()**
5: $Exp_k \leftarrow$ **defineExp()**
6: $Pol_k \leftarrow$ **setPolicy()**
7: $BH_k \leftarrow \{HashBH_{k\text{-}1}, k, Time_k, Exp_k, Pol_k, NPK_i\}$
8: *consensusResponses* $\leftarrow$ **performConsensus**(BH_k)
9: **if** *consensusResponses* $>$ *minimumResponses* **then**
10: **broadcast**(BH_k)
11: **end if**

level. After the block header is created, the consensus is performed (line 7). It is important to note that the consensus is performed only by gateway nodes. After the consensus is performed and it receives more than the minimum responses for each consensus algorithm, the new block is broadcast to the peers (line 10).

A simplified version of the consensus algorithm was applied. This simplified version uses both PBFT and Witness-based consensus algorithms that present good options based on security aspects. It is important to state that SpeedyChain consensus definition is a vast research topic. Due to this thesis research limitation, the consensus is still a work in progress in order to identify the best algorithm, or even support multiple algorithms.

5.4 Security Analysis

Chapter 4 presented a list of security threats that could affect the blockchain technology. Some of those threats can compromise SpeedyChain operation (see Table 5.1). SpeedyChain can cope with the following attacks: Alternative History, Balance, Block With-Holding, Bribery, Double Spending, 51%, Finney, Fork After WithHolding, Pool Hopping, Refund, Selfish Mining, Timejacking, Transaction Malleability, and Vector 76.

Among the principal vulnerabilities that can still compromise SpeedyChain are: Collision, Delay Routing, DDoS, Deanonymization, Eclipse, Sybil, and Vulnerable Signature.

Vulnerable Signature is a threat that is not exclusive to SpeedyChain, but is an exploit that focus on cryptography algorithm implementation. As possible countermeasures, it is recommended to use cryptography algorithm implementation that follows NIST [NIS19] standards and guidelines. The **Collision** attack also focused on exploiting the cryptography algorithms, in particular, the hash collision. Possible mitigation for this attack is to use hash algorithms loose coupled to the core blockchain, in a way that changing the algorithm does not affect the blockchain operation.

Focused on different blockchain layers, attacks such as **Deanonymization** also could succeed when executed against SpeedyChain solutions. This attack is performed based on the connection executed between peers. Thus, possible mitigation on this issue is to define different layers in the architecture, and through the layers avoid the information to be tracked to its device source. The **DDoS** works in the communication layer, and it is a widespread attack nowadays for network applications. SpeedyChain also could be affected by this attack. As possible mitigation, the solution should avoid malicious network traffic that aims to consume target resources, among the possible mitigation, the firewall and IDS configuration with specific rules to block the package. However, SpeedyChain architecture allows a device to change the gateway, if one of them stops responding, sending data to any gateway available, thus mitigating a DDoS attack.

The communication layer still presents a significant threat for SpeedyChain. In this context, **Delay Routing** attack could also affect a SpeedyChain solution. Possible mitigation for this attack is to use the connection between device and gateways architecture, which decreases the possibility of an attacker to intercept this communication channel. The architecture proposed, also helps in the **Eclipse** and **Sybil** attacks mitigation, as the model reduces the possibility of an attacker to intercept the communication. However, it is important to highlight that the connection between different gateways also could be impacted by these attacks.

In addition to the presented attacks, SpeedyChain could also be susceptible to a malicious gateway attack. The compromised gateway can discard blocks and transactions received either from other gateways or from devices. This threat mitigation involves the consensus algorithm choice and definition. The current SpeedyChain implementation foresees the consensus applied by the gateways when creating new blocks. This operation requires the consensus among multiple gateways before inserting a new block and update blockchain copy in their peers. A lightweight consensus algorithm should be evaluated by controlling transaction insertion.

The consensus algorithm choice allows to mitigate the malicious gateway vulnerability and also improves the blockchain security. As presented in Table 4.2, most of the attacks are focused on consensus algorithms. SpeedyChain applies the consensus algorithm at the gateway level when a new block creation is requested. Due to the permissioned characteristic of SpeedyChain, it was implemented the PBFT consensus algorithm and it was introduced a Witness based consensus. Thus, this architectural decision avoids attacks that target the blockchain consensus layer. Attacks such as **Double Spending**, **Finney** and **Alternative History** are not threats for SpeedyChain once the consensus algorithm does not allow the main blockchain to be forked. Likewise, **51%** attack is possible to be performed in blockchains that use consensus-based in processing power, as our proposal is not using this algorithm, this attack is mitigated. Despite that, even if one of these attacks is performed, the transaction keeps a sequential timestamp, and if it is in an incorrect order, it will

be discarded. Additionally, only the device that owns the private key will be able to produce information to the blocks, this approach will mitigate the fork with conflicted transactions.

The **Vector 76** attack also exploits the conflicting blocks that might be generated in the consensus algorithms. Its main goal is to explore financial gains from cryptocurrency exchanges. As SpeedyChain does not use a token (or cryptocurrency), this attack, or the Double Spending attack, do not affect SpeedyChain. The **Transaction Malleability**, **Timejacking** and **Refund** attacks are very specific to exploiting protocols and business model definition that is applied in the Bitcoin blockchain.

Finally, it is important to highlight that the evaluated attacks to SpeedyChain target the gateways, which is the node responsible for running the blockchain. It was also assumed that the hardware is secure, as the proposed solution is focused on handling the data in order to keep its integrity and non-repudiation. In the same way, the communication channel security is not part of this research, and in order to protect the information exchanged between nodes (gateways and devices) cryptography algorithms were applied. The key management for these algorithms is the responsibility of each node and is assumed they can handle it properly to ensure its security.

Table 5.1 – SpeedyChain attacks analysis

Threat	**Affects SpeedyChain**
51%	No
Alternative History	No
Balance	No
Block WithHolding	No
Bribery	No
Double Spending	No
Finney	No
Fork After WithHolding	No
Pool Hopping	No
Refund	No
Selfish Mining	No
Timejacking	No
Transaction Malleability	No
Vector 76	No
Collision	Yes
DDoS	Yes
Deanonymization	Yes
Delay Routing	Yes
Eclipse	Yes
Sybil	Yes
Vulnerable Signature	Yes

5.5 Chapter summary

In this chapter, an alternative data structure for blockchains was presented. The main achievements in this proposal are to keep the data secured by the asymmetric cryptography algorithm, using a digital signature, which ensures the non-repudiation property.

Due to this data structure characteristic, it is possible to highlight that the block is divided into two different parts: Block header, which is immutable and contains all information needed to ensure the node that is sending information; Block ledger, which contains all transactions, and at anytime (limited by the expiration timeout) the node can send information, allowing to append data into an existing block. At the same time, it is possible to ensure that the gateways could work in parallel, appending data in different blocks at same time.

The proposed definition is consensus agnostic, *i.e.*, it should support different algorithms. However, it is important to highlight that once a consensus algorithm is defined, it should be used on that blockchain instance. In order to evaluate SpeedyChain, the PFBT and Witness consensus algorithm were applied, as these algorithms mitigate known attacks that are focused on the blockchain consensus layer.

6. EVALUATION

This chapter presents the experiments performed with the blockchain prototype implementation named SpeedyChain. These experiments aim to evaluate the performance of the developed prototype, and measure the common operations performed in the blockchain (adding blocks and appending transactions).

The experiments are divided into three different scenarios, which are Smart Homes, Smart Cities and Industrial IoT. The Smart Home experiment execution aims to evaluate how the most common cryptography and hash algorithms perform in standard hardware. The Smart Cities experiment aims to evaluate the blockchain data model in an emulated environment, gathering data related to the time to manage data using blockchain. In the last scenario, the goal is to increase the scalability in an emulated environment and use a different consensus algorithm.

6.1 Smart Home

An important aspect of implementing a blockchain in an IoT multi-tier architecture is to verify its applicability on a specific scenario. Furthermore, it is crucial to consider how constrained devices can handle the solution. Additionally, in the proposed architecture, the cryptography and hash functions play a crucial role in the blockchain application. Thus, its evaluation could indicate to the hardware that fits better the requirements. As examples of constrained devices, Arduino (*e.g.*, simple applications, such as to monitor sensors) and Raspberry Pi (*e.g.*, managing IP Cameras) boards are widely used to control devices over the Internet.

For this analysis, the following devices were chosen: Arduino UNO, a micro-controller board based on Atmel ATMega 328P (16 MHz clock and 32KB of programmable memory); Arduino Leonardo micro-controller board based on Atmel ATMega32u4 (16 MHz clock and 32KB of programmable memory); Arduino Mega 2560 micro-controller board based on Atmel ATMega2560 (16 MHz clock and 256KB of programmable memory); Raspberry Pi 2 B Boards (900MHz quad-core ARM Cortex-A7 CPU and 1GB of RAM memory); Orange Pi Zero (1.2GHz ARM Cortex-A7 CPU and 256MB for RAM memory); and regular PC (Intel®Core™i3 M350@2.27GHz, 8GB SODIMM DDR3 Ram, 120GB SSD, running Linux Ubuntu 14.04), which was chosen to establish a performance baseline.

First, some experiments were performed with the RSA algorithm - often used for key exchange. This is important to verify how devices can handle this algorithm that is known to be time consuming. After that, an evaluation was performed to know how these devices can handle the SHA256 algorithm - used on many blockchains, such as Bitcoin, to create

block and transaction hashes. After that, we performed an evaluation on how the boards can handle AES symmetric algorithm (less time consuming than RSA), commonly used to build a secure communication. Also, some experiments were performed to verify how the boards would handle both cryptography and hash algorithms - for example, to send encrypted and hashed data. For both RSA and AES cryptography algorithms, due to hardware constraints, we used predefined fixed keys. The results presented in Table 6.1 show the median value for 10 samples presenting a standard deviation smaller than 0.004ms.

Table 6.1 – Performance evaluation of constrained devices with RSA, AES256 and SHA256

	Arduino Uno	**Arduino Leonardo**	**Arduino Mega**	**Raspberry Pi 2**	**Orange Pi Zero**	**PC**
RSA Encrypt	15.0ms	15.1ms	15.4ms	0.4ms	0.36ms	0.07ms
RSA Decrypt	9966.1ms	10020.2ms	10177.9ms	0.5ms	0.5ms	0.1ms
SHA256	22.3ms	22.2ms	22.3ms	0.16ms	0.18ms	0.03ms
RSA(TEXT +SHA256)	63.8ms	58.3ms	64.6ms	1.1ms	0.8ms	0.14ms
AES Encrypt	6.5ms	6.6ms	6.6ms	0.07ms	0.07ms	0.01ms
AES Decrypt	25.9ms	26.1ms	26.0ms	0.06ms	0.07ms	0.01ms
AES + SHA256	32.6ms	33.0ms	32.7ms	0.25ms	0.3ms	0.03ms

Based on the results presented in Table 6.1, we identified that even Arduino, which has limited memory and processing power resources, was able to run the RSA algorithm. However, it takes a considerable amount of time to get the text ciphered and deciphered, when compared to Raspberry Pi 2, Orange Pi Zero or PC. For example, text deciphering RSA using Arduino took around 10,000 ms, while the same text deciphering using Raspberry Pi 2 or Orange Pi Zero took only 0.5 ms. This difference becomes smaller when the SHA256 hash algorithm is executed. In that case, the difference reduces to 22 ms. Thus, taking this results into account, Raspberry Pi 2 and Orange Pi Zero were chosen to host the blockchain (gateway), while Arduino will only be used to manage devices (sensors and actuators). Also, it is important to notice that both Arduino Mega, Leonardo and UNO had similar performance to handle cryptography algorithms.

Since Raspberry Pi 2 and Orange Pi Zero will be playing the gateway role in the proposed architecture, their performance was compared to regular PC in order to establish a time parameter of hosting the blockchain. Table 6.2 presents the information for running the blockchain in a Raspberry Pi 2, Orange Pi Zero and in a regular PC (as described previously). Two operations were executed: (*i*) AES key generation, which consists of the operation when a device is beginning a communication to the gateways and, at this point, the gateway will generate an AES key and cipher this random key using the device RSA public key fetched from the block header in the blockchain; and (*ii*) appending information to an existing device block, where the gateway receives a package containing information

and the device signature; after that, the gateway, using its own RSA private key, signs the package and appends it to the block in the blockchain.

The results (evaluated in both situations) had better performance when operations are executed in the PC than in Raspberry Pi 2 or Orange Pi Zero. However, as the IoT architecture proposed considers the use of constrained hardware (in terms of processing capabilities and power consumption), Raspberry Pi and Orange Pi showed acceptable results in terms of processing time.

Table 6.2 – Performance for connecting and appending new block in a blockchain

		Average Time	**Sample Standard Deviation**	**Sample Confidence**
AES Key Generation	*Orange Pi Zero*	2.78 ms	0.08 ms	0.02 ms
	Raspberry Pi 2	3.59 ms	0.49 ms	0.30 ms
	PC	0.86 ms	0.15 ms	0.09 ms
Append Block Data	*Orange Pi Zero*	45.7 ms	0.69 ms	0.43 ms
	Raspberry Pi 2	20.99 ms	0.50 ms	0.31 ms
	PC	3.26 ms	0.67 ms	0.42 ms

Analyzing the results from Raspberry Pi 2 and Orange Pi Zero, we noticed that the operation to sign and append new information to a block is more time consuming than the key generation and encryption using AES. We calculated the confidence level considering an alpha value of 95%, which produced the values shown in Table 6.2. Thus, the time value to append new information into a block in the blockchain has an average time of 45.7ms on Orange Pi Zero and 20.99 ms on Raspberry Pi 2. Also, considering the confidence interval value, it will ensure that 95% of samples are in the range from 45.27 to 46.13ms on Orange Pi Zero and 20.69 to 21.31 ms on Raspberry Pi 2. Moreover, the AES key generation also ensures that 95% of samples are in the range from 2.76 to 2.80 ms on Orange Pi Zero and 3.29 to 3.89 ms on Raspberry Pi 2.

Once the hardware evaluated using the times as baseline, and thus choose an architecture for a possible IoT scenario, we considered, in this evaluation, a three-floor building environment with lighting controlled by smart devices. Each floor is managed by a gateway (*e.g.* Raspberry Pi 2 B or Orange Pi Zero). Also, each room has luminosity sensors, dimmers and relay managed by one or more devices (*e.g.* Arduino boards). Thus, the IoT infrastructure employed in this evaluation is composed by 3 gateways - one Raspberry Pi 2 and two Orange Pi Zero, all of them with Raspbian OS - and devices (Arduino UNO) to measure and control the lightning. The proposed architecture is show in Figure 6.1.

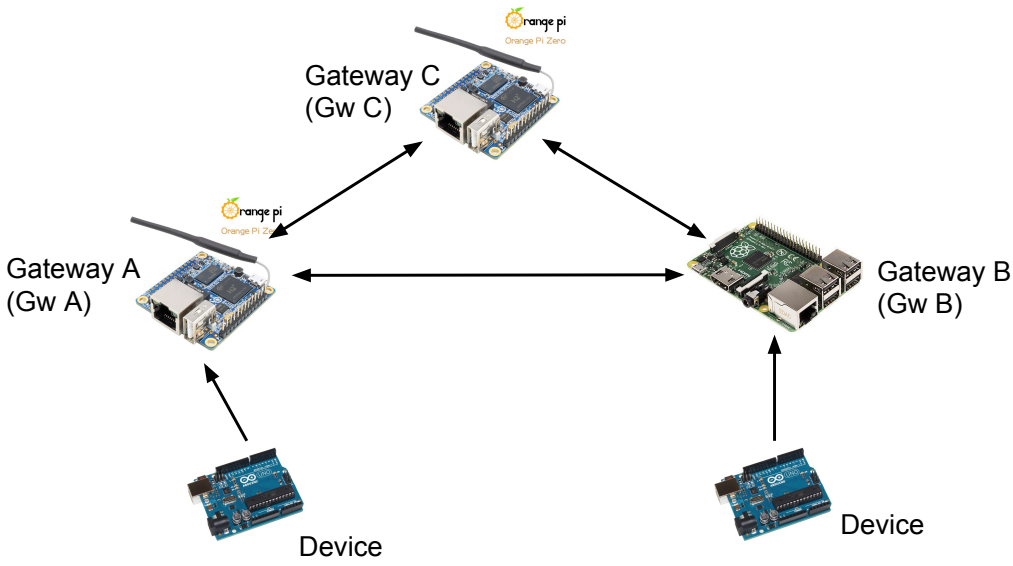

Figure 6.1 – Smart Home hardware architecture

In this scenario, we assumed that each device was registered previously by an administrator. Consequently, each Block Header (with the device public key) in the blockchain was already created when the experiments started. After the key exchange procedure (to use an AES Key generated by the gateway), a device sends 100 data updates, in a rate of one update per second, to the corresponding gateway. Each update is appended to the corresponding block in the blockchain and propagated to the other gateways. The experiment was repeated 10 times and we present the median time for each block in sequence.

Figure 6.2 presents the median time to append the data received from the device to generate both gateway signature and the hash of the previous block, append it to its block and send it to other peers. As can be observed in Figure 6.2, gateway *Gw A* takes from 45 to 70 ms to append and send the transaction generated to the other gateways. It is important to highlight that some collected time presents a time deviation (transactions between #50 and #60) mainly due to the operating system that is running in the IoT boards.

Additionally, the time to append a transaction (to the blockchain) into gateways *Gw B* and *Gw C* was measured (gateways that are not directly connected to the device that sends information). It is important to mention that only the time spent after the other gateways received the transaction is considered, *i.e.*, the time spent to send a transaction onto the network is not considered. As can be observed in Figure 6.3, the time to append a transaction on *Gw B* goes from 0.4 to 0.8 ms and on *Gw C* from 0.4 to 0.83ms.

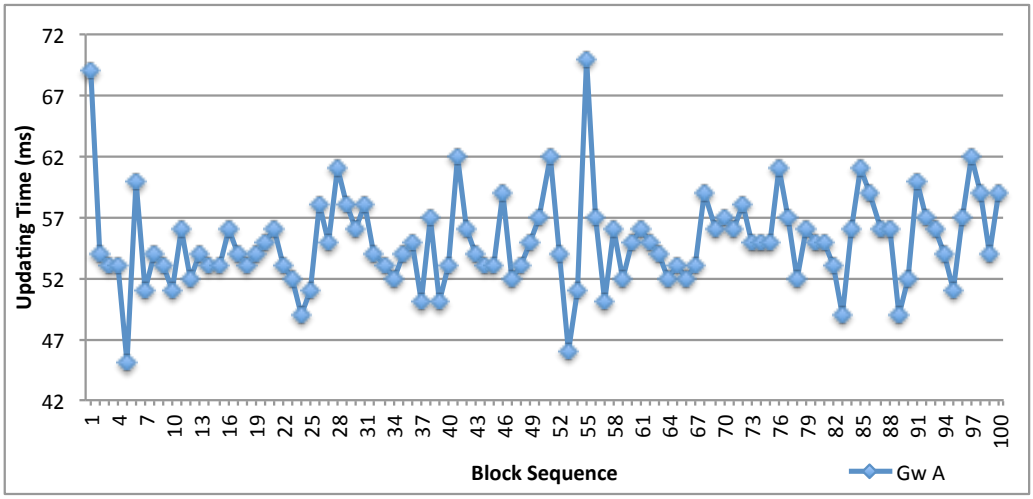

Figure 6.2 – Performance for appending and sending information to the gateways

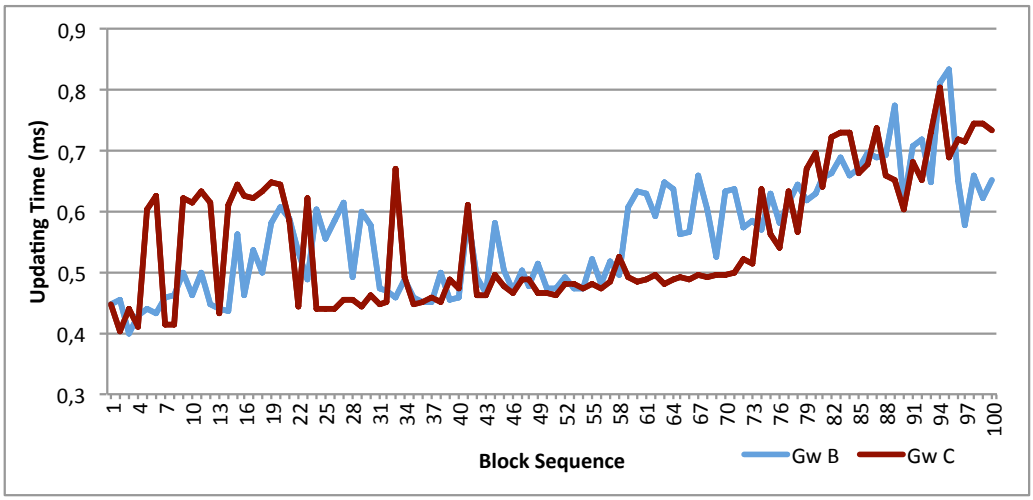

Figure 6.3 – Performance for appending new transaction on a block

Considering the evaluated scenario, *Gw A* (a gateway that controlled the device) takes around 5 to 7% of the time between updates to generate a transaction, to append it to block and to send it to other gateways. Furthermore, *Gw B* and *Gw C* take less than 0.1% of the time between updates to append information into the block. Consequently, the proposed blockchain presented promising performance results. However, it is important to evaluate the solution with a higher number of devices and gateways.

6.2 Smart City

In order to extend SpeedyChain evaluation, a second use case scenario was considered to run the experiments. This scenario is a smart city environment, in which the previous evaluation was extended and time to add a new block was evaluated. SpeedyChain includes smart city infrastructure (*e.g.*, traffic lights), smart vehicles, RSIs, and SPs

as shown in Figure 6.4. The blockchain management, *i.e.*, verification of transactions and blocks, and appending new blocks, are performed by participants that have more available resources and a closer commitment to the smart city, *e.g.*, RSIs and SPs.

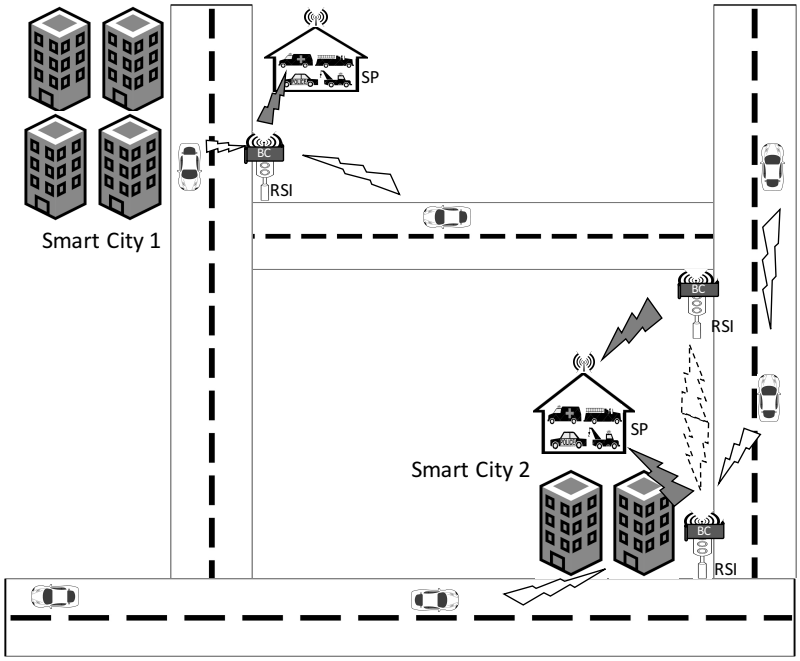

Figure 6.4 – Smart City evaluated architecture

The results were achieved in an emulated scenario to evaluate the performance of our approach. We emulated the scenario using the CORE network emulator [ADHK08], running in a VM in VirtualBox with 4 processors and *8GB* of *RAM* and the host Intel *i7@2.8Ghz* and *16GB* of *RAM*. The emulated scenario consists of 15 nodes representing RSIs that are interconnected with other RSIs and keep the blockchain updated. Thus, an RSI will be responsible for receiving connection requests from the vehicles, validating them, creating new blocks, and broadcasting these blocks to the other RSIs. Vehicles are responsible for establishing a connection to an RSI and, once connected, produce data and send that to the RSI.

In case of the vehicle moving among different RSIs, the proposed solution is not considerably affected, since once the block header is created in the blockchain, any RSI will be able to validate and append new transactions.

To evaluate the scalability of the proposed method, we vary the number of transactions generated by the vehicles, and study the solution performance with 10, 100, and 1,000 transactions in the network. We also vary the number of participating vehicles in the application and study the performance in three scenarios with 50, 100, and 650 vehicles. Overall we run 9 different scenarios to study the performance of each metric. This allowed evaluating the proposed framework performance when managing transactions and blocks in the smart city scenario.

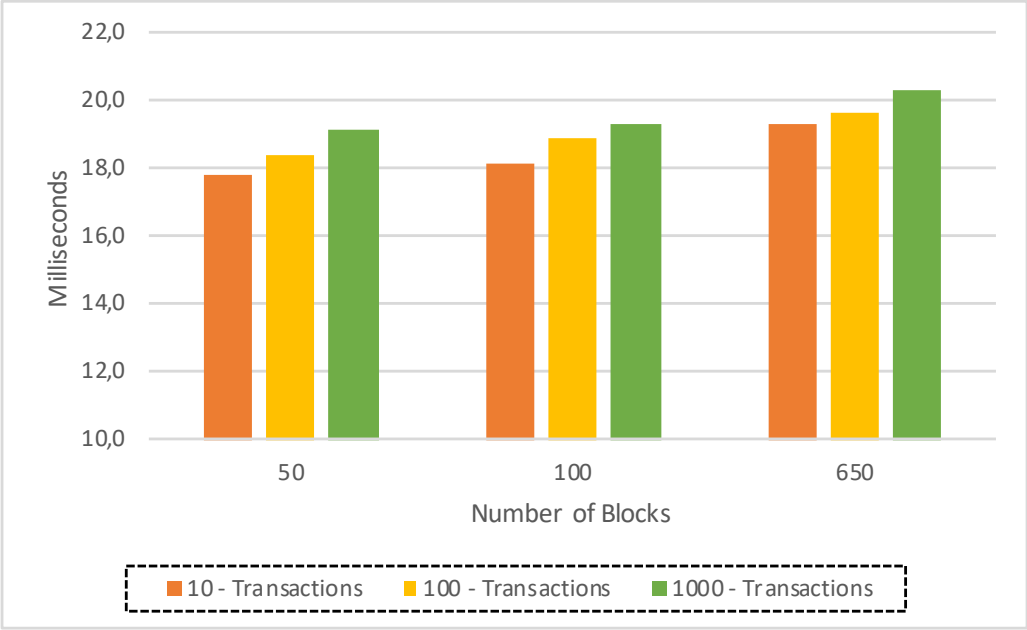

Figure 6.5 – Required time in an RSI to add a new block to the blockchain.

We first evaluated the processing time taken by an RSI to add a new block into the blockchain. The process of adding a new block includes: *i*) receiving a connection request at the RSI, *ii*) validating the vehicle request, *iii*) identifying the need for a new block and creating it, and *iv*) updating the RSI peers. Figure 6.5 summarizes the emulation results. As expected, the processing time for adding a new block to the blockchain increases as the number of blocks increases, since there are more transactions and blocks to be validated by the RSI. It can also be seen that the number of transactions also directly affects the processing time (Figure 6.6). The reason is that a higher number of transactions, in the block ledger, takes longer to be validated. However, processing time overhead for transactions is less significant than for new block creation. Note that the block creation operation will only be executed when a vehicle connects to an RSI the first time (or when it changes its key pair).

A block is created only when a vehicle joins the network. However, once its block is added to the blockchain, the vehicle is allowed to generate transactions. The next metric that we evaluated is the time taken by RSIs to process received transactions, *i.e.*, verify data signature, check if the block is not expired, append the transaction and notify all RSIs regarding the update. Figure 6.6 plots this metric. In this scenario, it was noticed that for a blockchain size of 50, the time increases 3.63% when increasing the number of transactions from 10 to 100. In a blockchain with 100 blocks, the transaction creation time increases 4.62%. However, for a blockchain size of 650 blocks, the time to validate and append new transactions increases by 36.97% from 10 to 1,000 transactions, which points to a linear time increase, as expected. Based on the collected samples, assuming a confidence level

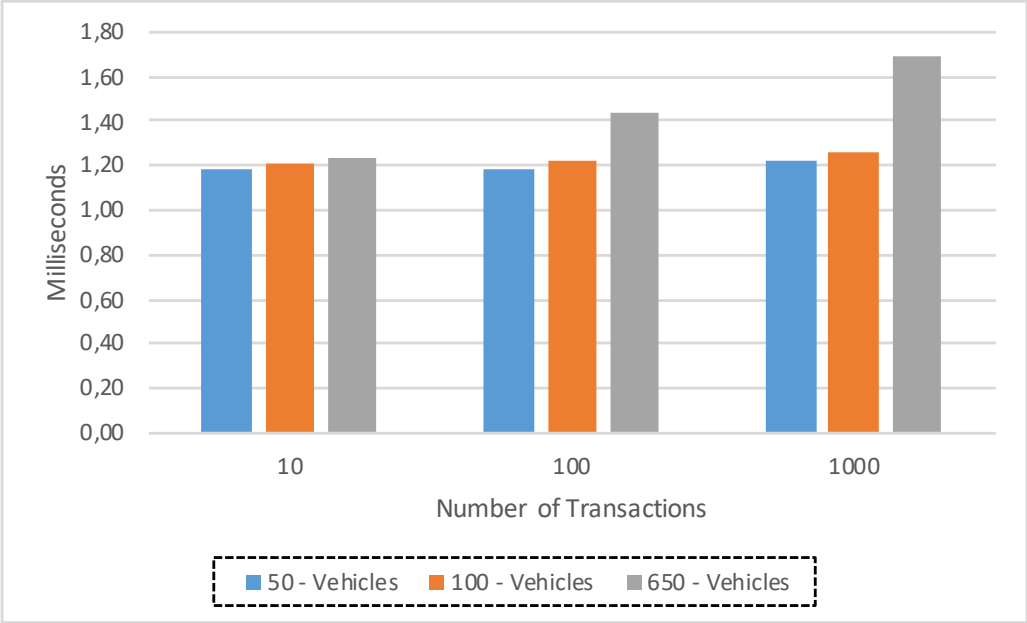

Figure 6.6 – Required processing time to add new transaction to the blockchain.

of 95%, the interval varies from 0.0013 ms to 0.0126 ms, *i.e.* a very small variance on the data.

Each transaction is created by the vehicle and sent to the RSI. The receiving RSI validates the transaction to ensure the integrity and trust in the data coming from the vehicle. Once all validation is performed, the RSI updates its local blockchain copy and sends it to other RSIs. Each RSI that receives the new transaction validates the received data before appending this new transaction to a local copy of the blockchain. The time required to execute this update operation increases as more transactions are stored in each block, and this behavior is shown in the tests presented in Figure 6.7. Considering a blockchain with 650 blocks, and changing the number of transactions from 10 to 1,000, the time to validate and update the blockchain increases by 103.08%, which is justified by having 100 times more transactions to be processed. Considering a confidence level of 95% for the samples, the error probability is at 0.003 ms to 10 transactions and 0.002 ms to 1,000 transactions.

The time that a single peer takes to validate and to update its blockchain, with a new block created by another RSI, is shown in Figure 6.8. This value varies from 0.021 ms to 0.025 ms for a blockchain with 50 blocks and 10 transactions to 650 blocks and 1,000 transactions, respectively.

Considering the evaluated blockchain sizes in terms of transactions and amount of blocks, even considering the scenario of a blockchain with 650 blocks sending 1,000 transactions, the values are 20.33 ms to validate and to create a new block and 1.69 ms to create and to validate a new transaction. The validation time for this scenario represents 7.7% of the operation total time. It represents a good improvement in terms of time to add a new transaction in comparison to the Bitcoin blockchain.

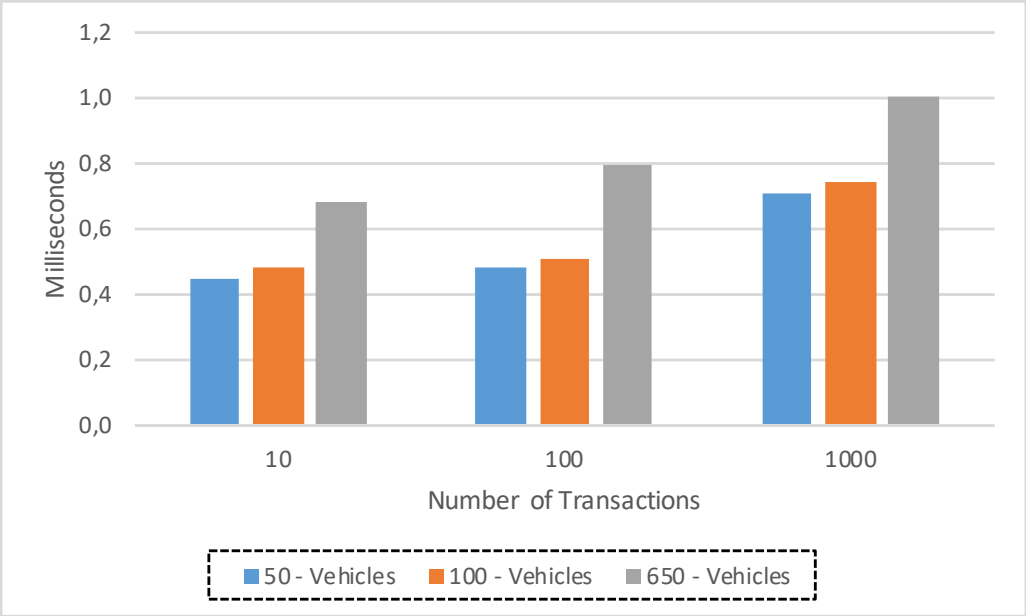

Figure 6.7 – Time (ms) to update peer's blockchain with received transactions

Table 6.3 – Time taken by vehicles to calculate the Merkle tree root.

# of Transactions	Time to calculate the Merkle tree
10	0.162 ms
100	0.857 ms
1000	7.995 ms

The time taken to calculate the Merkle tree in a vehicle was measured and is presented in Table 6.3. As this operation is performed on the transactions, we consider three values for the number of transactions in the block ledger that are 10, 100 and 1,000, and evaluated the time to generate the Merkle tree for each set. The time for a block with 10 transactions was 0.126 milliseconds, while for 1,000 transactions this time increases to 7.99 milliseconds. As expected, the time to calculate the Merkle tree increases as there are more transactions within a block. These results present a sublinear time increase, and, it is possible to estimate the key update interval used to define the block expiration time.

6.3 Industrial Internet of Things

The last use case defines in order to evaluate the performance of SpeedyChain is the IIoT scenarios, the CORE emulator platform [ADHK08] was used on the experiments to have consistent protocols and communication times spent in the Network layer. The evaluation was executed on a VMware Fusion 8.5.10 with 6 processors and *12GB* of *RAM* on an Intel *i7@2.8Ghz* and *16GB* of *RAM*. We performed the evaluation using 10 gateways,

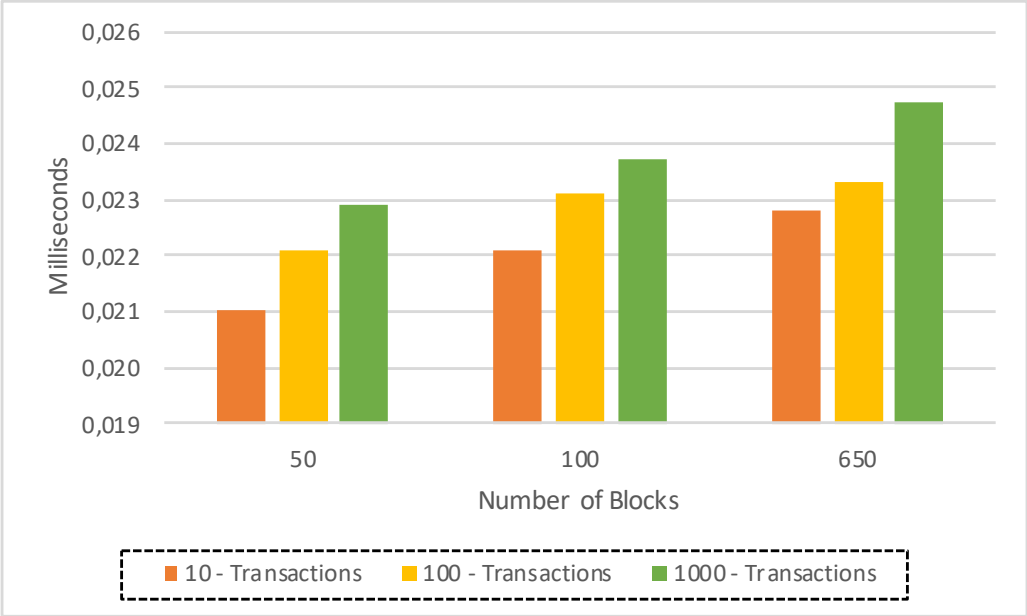

Figure 6.8 – Time (ms) to update peer's blockchain with received blocks

where each gateway runs in a container based-virtualized machine; in 9 different scenarios (as presented in Table 6.4) using 100, 500 and 1000 devices connected through theses gateways (10, 50 and 100 per gateway) and 100, 500 and 1,000 transactions per device (*e.g.*, 1,000,000 transactions in Scenario I). All times presented in Table 6.4 represent the median time considering the whole execution in all gateways.

The witness based consensus was used as a baseline in terms of time to append blocks and information. As expected, it can be observed in Table 6.4 that varying the consensus algorithm has an impact in the performance in the task to make the consensus of a block (used to insert block header with the public key of each device). For example, in Scenario A, witness based consensus takes 58.20ms to achieve the consensus of a block against 102.82ms using PBFT and in Scenario I (scenario with the highest number of devices and transactions), witness based takes 79.22ms against 160.35ms using PBFT (more than twice the time). However, Witness-based consensus is more likely to be affected by different attacks (*e.g.*, Eclipse and Sybil attacks) in comparison to PBFT.

In the other blockchain operations - for instance, time to add a new block in the leader gateway (a gateway that started the consensus), as well as the time to update the blockchain, to append a new transaction in a gateway (where devices are connected to) and to update the blockchain with the new transaction - presented few or no impact using both consensus algorithms. However, the number of transactions and nodes influenced in the processing time to append a transaction in the most demanding scenario (Scenario I) takes less than 7ms to both append the transaction (4.28ms in Witness-based and 4.55ms in PBFT) and to update a new transaction in the other gateways (2.33ms in Witness-based and 2.39ms in PBFT).

Table 6.4 – Performance evaluation considering different consensus

	A	B	C	D	E	F	G	H	I
Devices per Gw	10	10	10	50	50	50	100	100	100
Transactions per Device	100	500	1,000	100	500	1,000	100	500	1,000
Total of Devices' Blocks	100	100	100	500	500	500	1,000	1,000	1,000
Total of Transactions	10,000	50,000	100,000	50,000	250,000	500,000	100,000	500,000	1,000,000
Block Consensus (Witness)	58.20ms	64.01ms	65.25ms	64.51ms	71.02ms	71.73ms	69.13ms	72.47ms	79.22ms
Block Consensus (PBFT)	102.82ms	119.53ms	121.68ms	121.98ms	126.56ms	132.37ms	129.14ms	136.86ms	160.35ms
Add Block in Leader (Wit.)	3.72ms	3.56ms	4.42ms	4.66ms	4.82ms	5.81ms	5.33ms	5.95ms	6.28ms
Add Block in Leader (PBFT)	3.40ms	4.45ms	5.16ms	4.21ms	4.87ms	5.88ms	5.29ms	5.93ms	6.52ms
Update Blockchain w/ Block (Wit.)	0.22ms	0.22ms	0.23ms	0.22ms	0.23ms	0.23ms	0.23ms	0.24ms	0.25ms
Update Blockchain w/ Block (PBFT)	0.22ms	0.22ms	0.23ms	0.23ms	0.23ms	0.26ms	0.24ms	0.24ms	0.27ms
Append Transaction in Gw. (Wit.)	2.66ms	2.82ms	2.91ms	3.24ms	3.49ms	3.54ms	3.89ms	4.29ms	4.28ms
Append Transaction in Gw. (PBFT)	2.69ms	2.80ms	2.90ms	3.30ms	3.46ms	4.00ms	3.96ms	4.16ms	4.55ms
Update Blockchain w/ Trans. (Wit.)	0.94ms	1.18ms	1.48ms	1.30ms	1.58ms	1.89ms	1.73ms	2.11ms	2.33ms
Update Blockchain w/ Trans. (PBFT)	0.94ms	1.17ms	1.47ms	1.31ms	1.55ms	2.03ms	1.73ms	2.03ms	2.39ms

Additionally, it can be observed that growing the number of transactions (overload of processing in gateways) has more impact than the number of devices that a gateway is handling. For example, scenario D has half of the transactions and 5 times more nodes than C, but takes almost the same time to reach the consensus for a block. Differently, scenario F has half of the nodes and 5 times more transactions than scenario G, resulting in F spending around 3% more time to achieve the consensus than G. Figure 6.9 presents a comparison of the time to achieve a consensus of a block in different scenarios.

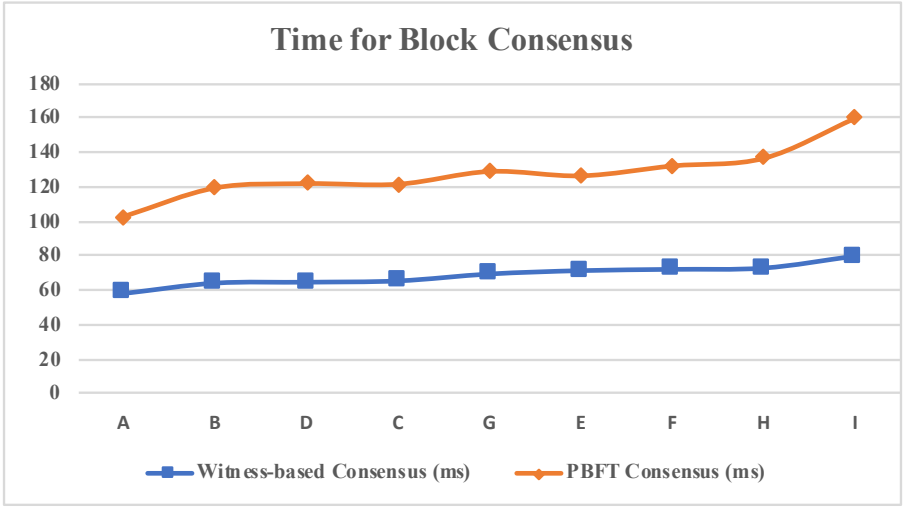

Figure 6.9 – Time for block consensus

As a comparison, Bitcoin network has around 10,000 [Bit18] active nodes in a 24-hour slice, consequently, the experiment in Scenario I represents approximately 10% of the Bitcoin network (which is current blockchain standard). As a comparison, Bitcoin has more than 150,000 confirmed transactions per day [CELR18] with a peak of 490,644 confirmed transactions in a day [Blo], which means that the evaluation in Scenario I, at least represents more than twice the transactions in the Bitcoin blockchain in a day. A more effective comparison could be made with IOTA [IOT18] - a blockchain developed for IoT - which has around 8.7 transactions per second [IOT] (data collected in August of 2018). This means around 750,000 transactions processed in a day (around 75% of the transactions processed in Scenario I). Also, it represents that the IOTA transaction processing time is around 115ms. Consequently, the transactions processing time in SpeedyChain represents less than 6% of the time that is spent in IOTA - 115ms in IOTA and 7ms in SpeedyChain (4.55ms to append a transaction in a gateway summed with 2.39ms to update the entire blockchain using PBFT).

This evaluated scenario presents good results in the emulated IIoT scenarios with a different number of devices and transactions. It is important to note that the code that implements SpeedyChain was developed using the Python programming language and a set of libraries. The code is available at GitHub[1] and could be used to replicate the experiments.

6.4 Discussion

Based on the experiments, we can show that the proposed blockchain data model is capable of handling the produced data including a small overhead for the blockchain.

As presented, the first scenario (smart home) was the research foundation, as it draws a bottom line to show that the most common cryptography and hash algorithms are able to run in constrained devices. This experiment also acts as a validation to, based on the algorithms execution time results, indicate and define which device should be used in each layer in an IoT architecture. Once the scenario was defined, the experiments were executed using real hardware, which shows the blockchain viability for the proposed scenario.

Once the SpeedyChain viability was evaluated using real hardware, the smart city scenario works to evaluate it in a larger scenario, and extending the function to support a consensus algorithm. In order to increase the number of nodes, this scenario was executed in a network emulator tool, which allowed to gather the blockchain operation time in a similar way as the real hardware. During the execution of this scenario, we noticed that the consensus algorithm needed to be better evaluated, since it plays an important role in the blockchain block addition.

[1] https://github.com/regio/r2ac

Finally, we ran an IIoT scenario, in which our main goal was to increase the number of nodes, and identify the consensus overhead when a new block is added to the blockchain. The results are still supporting that the SpeedyChain applicability fits for an increased scenario, as the blockchain processing time is still in milliseconds. Additionally, the results show that further investigation is needed in order to improve consensus performance without compromising solution security.

7. CONCLUSION

This work presented a permissioned lightweight blockchain data model for IoT or constrained devices. It was analyzed the most common security threats and vulnerabilities in blockchain in order to help to propose a solution that could fit the IoT scenario. It was evaluated the existing cryptography algorithm in real hardware, allowing to map the hardware to each layer in an IoT architecture. Finally, SpeedyChain was evaluated in different scenarios, changing consensus algorithm and scaling in order to gather results, after that the results show a promising solution that fits different IoT scenarios.

7.1 Hypothesis Foundation

Through the current research and use cases scenarios, it was possible to validate the research hypothesis (presented in Chapter 1). After running the use cases presented in Chapter 6, the SpeedyChain presented promising results. The data gathered during the executions supports the hypothesis, firstly showing standard constrained device can execute the most common security algorithms that are prerequisite to run a blockchain. Based on that, it leads us to define a lightweight solution that will run in these devices generating the minimum overhead for the blockchain handling. In order to achieve the lightweight, a change in the traditional blockchain data model was proposed, this change enables the block in the blockchain to allow appending data and given the blockchain the ability to allow different nodes including the transaction in different blocks at the same time. The proposed blockchain was evaluated in terms of security in Section 4, showing that the approach should be taken into account in order to avoid/mitigate most common blockchain attacks.

The defined research questions were followed and answered during the current thesis, in terms of supporting and validating the research hypothesis.

1. *What is the minimum hardware requirement to run a blockchain?*

   It was presented in Section 6.1 that common constrained device can execute the required cryptography algorithms that are needed to support a blockchain execution.

2. *What is the performance in constrained hardware to handle the algorithms need for supporting a blockchain?*

   Once identified that the constrained devices could handle the algorithms, it was evaluated in Section 6.1 the time that each device takes to perform these operations. Based on this research question, we could map that some devices presented time for decryption a message using asymmetric cryptography takes around 10 seconds (see Table 6.1). Thus, although all devices evaluated able to run all algorithms, due to this

performance, an IoT three tiers architecture identified during the literature review was chosen to run the proposed solution. In the evaluated scenario device such as Arduino boards are running to manage the device level, while more powerful board such as Raspberry and Orange Pi are better candidates to manage the blockchain.

3. *How to change/adapt the blockchain technology in order to become a lightweight solution capable of fits in embedded hardware (such as IoT devices)?*

   The answer to this research question leads to evaluate the possible bottlenecks that a traditional blockchain could presents. At this moment it was identified two main concerns in Section 2.3, which are (*i*) the consensus algorithm, as most of the blockchains are using the PoW consensus algorithm, and as presented in Section 2 this algorithm relies on the processing power. Thus for an IoT blockchain proposal, this algorithm should be switched, for this research it was considered two different options that are Witness based and PBFT, as these algorithms are vote based; (*ii*) the data structure that is defined for blockchains lead to a sequential block creation, which could lead to other bottleneck related to the nodes competing to include a block of transactions. This block ledger in some scenarios, due to the amount of produced data, could reach a considerable high size for storing this data structure. To work around this blockchain limitation, the current research is proposing a data structure that allows to decouple the payload (that is responsible for demanding the most storage space), and allow this ledger to be stored off-chain.

4. *What are the most common security threats that could compromise a blockchain?*

   As the blockchain technology brings by design the security aspects, is imperative to identify that are the most common security threats that could compromise different aspects of a blockchain, in order to subvert its behaviour, which could lead the blockchain to an inconsistent state. It was conducted a literature review for the purpose to identify these threats, which points to a set of threats evaluated in Section 4.1. Based on the analysis, the main components (but not limited) that are exploited to compromise the blockchain technology are the PoW consensus algorithms and the fork resolution algorithm.

5. *How to proposes an alternative data structure keeping the security?*

   After identifying the security threats and the bottleneck present in most common blockchain implementations, this research evaluated an alternative data structure that is still able to ensure the data security and integrity applying the most traditional symmetric and asymmetric cryptography algorithms and at the same time allows to improve the time take to handle new transactions. In order to support the required demand, it was proposed a data structure for the blockchain blocks that allows the header is not dependent on the transactions (as happen in most common blockchains, through the

merkle three usage), but the proposed data structure the first transaction depends on the block header and the following transactions depends on the previous as presented in Section 5.3.2. Through this proposal, the data structure enables the block to accept new transaction even after a new block header being added to the blockchain, and still allows to multiple nodes, append transactions in different blocks at the same time.

7.2 Future Directions

The increasing number of devices connected to the Internet and networks, still presents a challenge, especially in term of security and scalability aspects. The present work shows a step toward proposing solutions capable of handling these demands, however, it is important to highlight this research field still needs attention and can be extended.

We could list some research opportunities that can improve this research, such as, to evaluate/propose different consensus algorithms to fit the hardware limitation presented in most common IoT devices. This evaluation should compare the processing power needs and response time, for example. A research field is also presented in order to evaluate the network usage, in terms of latency and number of messages, for example. Linked with these topics, it should also take into account the security aspects as well as threats that could be exploited in consensus and network.

An important aspect is once the information becomes available in the blockchain, to define an access control mechanism able to restrict the data access.

REFERENCES

[ABC17] Atzei, N.; Bartoletti, M.; Cimoli, T. “A survey of attacks on ethereum smart contracts sok”. In: Proceedings of the International Conference on Principles of Security and Trust (POST), 2017, pp. 164–186.

[ADHK08] Ahrenholz, J.; Danilov, C.; Henderson, T. R.; Kim, J. H. “Core: A real-time network emulator”. In: Proceedings of the IEEE Military Communications Conference (MILCOM), 2008, pp. 1–7.

[ADMM13] Andrychowicz, M.; Dziembowski, S.; Malinowski, D.; Mazurek, L. “How to deal with malleability of bitcoin transactions”. In: Proceedings of the Financial Cryptography and Data Security: FC International Workshops, 2013, pp. 1–6.

[ADMM15] Andrychowicz, M.; Dziembowski, S.; Malinowski, D.; Mazurek, Ł. “On the malleability of bitcoin transactions”. In: Proceedings of the Financial Cryptography and Data Security, 2015, pp. 1–18.

[AEVL16] Azaria, A.; Ekblaw, A.; Vieira, T.; Lippman, A. “Medrec: Using blockchain for medical data access and permission management”. In: Proceedings of the International Conference on Open and Big Data (OBD), 2016, pp. 25–30.

[AIM10] Atzori, L.; Iera, A.; Morabito, G. “The internet of things: A survey”, *Computer Networks*, vol. 54–15, Jun 2010, pp. 2787 – 2805.

[AK14] Abomhara, M.; Køien, G. M. “Security and privacy in the internet of things: Current status and open issues”. In: Proceedings of the International Conference on Privacy and Security in Mobile Systems (PRISMS), 2014, pp. 1–8.

[AMV16] Arseni, S. C.; Mitoi, M.; Vulpe, A. “Pass-iot: A platform for studying security, privacy and trust in iot”. In: Proceeding of the International Conference on Communications (COMM), 2016, pp. 261–266.

[Ash09] Ashton, K. “That ”Internet of Things” Thing”. Source: http://www.rfidjournal.com/articles/view?4986, Feb 2019.

[ASW17] Ahemd, M. M.; Shah, M. A.; Wahid, A. “Iot security: A layered approach for attacks defenses”. In: Proceedings of the International Conference on Communication Technologies (ComTech), 2017, pp. 104–110.

[AWHJ15] Arias, O.; Wurm, J.; Hoang, K.; Jin, Y. “Privacy and security in internet of things and wearable devices”, *IEEE Transactions on Multi-Scale Computing Systems*, vol. 1–2, Apr 2015, pp. 99–109.

[AZV17] Apostolaki, M.; Zohar, A.; Vanbever, L. “Hijacking bitcoin: Routing attacks on cryptocurrencies”. In: Proceeding of the IEEE Symposium on Security and Privacy (S&P), 2017, pp. 375–392.

[Bal19] Baliga, A. “Understanding Blockchain Consensus Models”. Source: https://pdfs.semanticscholar.org/da8a/37b10bc1521a4d3de925d7ebc44bb606d740.pdf, Feb 2019.

[BBG+17] Boudguiga, A.; Bouzerna, N.; Granboulan, L.; Olivereau, A.; Quesnel, F.; Roger, A.; Sirdey, R. “Towards better availability and accountability for iot updates by means of a blockchain”. In: Proceedings of the IEEE European Symposium on Security and Privacy Workshops (EuroS PW), 2017, pp. 50–58.

[BHH+14] Bos, J. W.; Halderman, J. A.; Heninger, N.; Moore, J.; Naehrig, M.; Wustrow, E. “Elliptic curve cryptography in practice”. In: Proceedings of the Financial Cryptography and Data Security, 2014, pp. 157–175.

[BHvOS12] Bonneau, J.; Herley, C.; v. Oorschot, P. C.; Stajano, F. “The quest to replace passwords: A framework for comparative evaluation of web authentication schemes”. In: Proceedings of the IEEE Symposium on Security and Privacy (S&P), 2012, pp. 553–567.

[Bit18] BitNodes. “Global bitcoin nodes distribution”. Source: https://bitnodes.earn.com/, Aug 2018.

[Bit19] BitInfoCharts. “Cryptocurrency statistics”. Source: https://bitinfocharts.com/index_v.html, Feb 2019.

[BKP14] Biryukov, A.; Khovratovich, D.; Pustogarov, I. “Deanonymisation of clients in bitcoin p2p network”. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2014, pp. 15–29.

[Blo] Blockchain.com. “Confirmed transactions per day in bitcoin”. Source: https://www.blockchain.com/charts/n-transactions, Aug 2018.

[Blo19] Blockchain.com. “Bitcoin hashrate distribution”. Source: https://www.blockchain.com/pools, Jan 2019.

[BM13] Biswas, S.; Mišić, J. “A cross-layer approach to privacy-preserving authentication in wave-enabled vanets”, *IEEE Transactions on Vehicular Technology*, vol. 62–5, Jun 2013, pp. 2182–2192.

[BMZB18] Basegio, T. L.; Michelin, R. A.; Zorzo, A. F.; Bordini, R. H. “A decentralised approach to task allocation using blockchain”. In: Proceedings of the Engineering Multi-Agent Systems, 2018, pp. 75–91.

[Bon16] Bonneau, J. “Why buy when you can rent?” In: Proceedings of the Financial Cryptography and Data Security, Clark, J.; Meiklejohn, S.; Ryan, P. Y.; Wallach, D.; Brenner, M.; Rohloff, K. (Editors), 2016, pp. 19–26.

[Bov18] Boverman, A. “Timejacking & bitcoin - the global time agreement puzzle”. Source: http://culubas.blogspot.com/2011/05/timejacking-bitcoin_802.html, Nov 2018.

[BTRB17] Bloom, C.; Tan, J.; Ramjohn, J.; Bauer, L. “Self-driving cars and data collection: Privacy perceptions of networked autonomous vehicles”. In: Proceedings of the Symposium on Usable Privacy and Security (SOUPS), 2017, pp. 357–375.

[Cac16] Cachin, C. “Architecture of the hyperledger blockchain fabric”. In: Proceedings of the Workshop on Distributed Cryptocurrencies and Consensus Ledgers, 2016, pp. 1–6.

[CD16] Christidis, K.; Devetsikiotis, M. “Blockchains and Smart Contracts for the Internet of Things”, *IEEE Access*, vol. 4, May 2016, pp. 2292–2303.

[CELR18] Conti, M.; E, S. K.; Lal, C.; Ruj, S. “A survey on security and privacy issues of bitcoin”, *IEEE Communications Surveys Tutorials*, vol. 20–4, May 2018, pp. 1–10.

[Col18] Colyer, A. “New bitcoin / pow blockchain fork-after-withholding attack favours the larger player in a mutual attack game”. Source: https://twitter.com/adriancolyer/status/938812855910043649, Dec 2018.

[CVM17] Conoscenti, M.; Vetrò, A.; Martin, J. C. D. “Peer to peer for privacy and decentralization in the internet of things”. In: Proceedings of the IEEE/ACM International Conference on Software Engineering Companion (ICSE-C), 2017, pp. 288–290.

[CZ16] Chiang, M.; Zhang, T. “Fog and iot: An overview of research opportunities”, *IEEE Internet of Things Journal*, vol. 3–6, Dec 2016, pp. 854–864.

[DJD+20] Dedeoglu, V.; Jurdak, R.; Dorri, A.; Lunardi, R. C.; Michelin, R. A.; Zorzo, A. F.; Kanhere, S. S. “Blockchain Technologies for IoT”. Singapore: Springer Singapore, 2020, chap. 3, pp. 55–89.

[DKJ17] Dorri, A.; Kanhere, S. S.; Jurdak, R. “Towards an Optimized BlockChain for IoT”. In: Proceedings of the Second International Conference on Internet-of-Things Design and Implementation, 2017, pp. 173–178.

[DKJ19] Dorri, A.; Kanhere, S. S.; Jurdak, R. “Mof-bc: A memory optimized and flexible blockchain for large scale networks”, *Future Generation Computer Systems*, vol. 92, Mar 2019, pp. 357–373.

[Dou02] Douceur, J. R. “The sybil attack”. In: Proceedings of the First International Workshop on Peer-to-Peer Systems, 2002, pp. 251–260.

[DSKJ17] Dorri, A.; Steger, M.; Kanhere, S. S.; Jurdak, R. “Blockchain: A distributed solution to automotive security and privacy”, *IEEE Communications Magazine*, vol. 55–12, Dec 2017, pp. 119–125.

[ES14] Eyal, I.; Sirer, E. G. “Majority is not enough: Bitcoin mining is vulnerable”. In: Proceedings of the Financial Cryptography and Data Security, 2014, pp. 436–454.

[Eth17] Ethereum. “Ethereum project”. Source: https://www.ethereum.org/, Jan 2017.

[Fin11] Finney, H. “Best practice for fast transaction acceptance - how high is the risk?” Source: https://bitcointalk.org/index.php?topic=3441.msg48384#msg48384, Dec 2018.

[FMMT18] Fenu, G.; Marchesi, L.; Marchesi, M.; Tonelli, R. “The ico phenomenon and its relationships with ethereum smart contract environment”. In: Proceedings of the International Workshop on Blockchain Oriented Software Engineering (IWBOSE), 2018, pp. 26–32.

[GCR16] Giechaskiel, I.; Cremers, C.; Rasmussen, K. B. “On bitcoin security in the presence of broken cryptographic primitives”. In: Proceedings of the European Symposium on Research in Computer Security (ESORICS), 2016, pp. 201–222.

[GKN+11] Gluhak, A.; Krco, S.; Nati, M.; Pfisterer, D.; Mitton, N.; Razafindralambo, T. “A survey on facilities for experimental internet of things research”, *IEEE Communications Magazine*, vol. 49–11, Nov 2011, pp. 58–67.

[GKW+16] Gervais, A.; Karame, G. O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. “On the security and performance of proof of work blockchains”. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 3–16.

[Gro13] Gross, B. “Google’s self driving car gathers nearly 1 gb/sec”. Source: https://www.linkedin.com/pulse/20130502024505-94747-google-s-self-driving-car-gathers-nearly-1-gb-per-second/, Jan 2019.

[HCK17] Huh, S.; Cho, S.; Kim, S. “Managing iot devices using blockchain platform”. In: Proceedings of the International Conference on Advanced Communication Technology (ICACT), 2017, pp. 464–467.

[HCL04] Hubaux, J. P.; Capkun, S.; Luo, J. “The security and privacy of smart vehicles”, *IEEE Security Privacy (S&P)*, vol. 2–3, May 2004, pp. 49–55.

[HIJ+12] Hoh, B.; Iwuchukwu, T.; Jacobson, Q.; Work, D.; Bayen, A. M.; Herring, R.; Herrera, J. C.; Gruteser, M.; Annavaram, M.; Ban, J. “Enhancing privacy and accuracy in probe vehicle-based traffic monitoring via virtual trip lines”, *IEEE Transactions on Mobile Computing*, vol. 11–5, May 2012, pp. 849–864.

[HKL17] Haidar, F.; Kaiser, A.; Lonc, B. “On the performance evaluation of vehicular pki protocol for v2x communications security”. In: Proceedings of the IEEE Vehicular Technology Conference (VTC-Fall), 2017, pp. 1–5.

[HKZG15] Heilman, E.; Kendler, A.; Zohar, A.; Goldberg, S. “Eclipse attacks on bitcoin’s peer-to-peer network”. In: Proceedings of the USENIX Security Symposium, 2015, pp. 129–144.

[HSR+18] Hildenbrandt, E.; Saxena, M.; Rodrigues, N.; Zhu, X.; Daian, P.; Guth, D.; Moore, B.; Park, D.; Zhang, Y.; Stefanescu, A.; Rosu, G. “Kevm: A complete formal semantics of the ethereum virtual machine”. In: Proceedings of the IEEE Computer Security Foundations Symposium (CSF), 2018, pp. 204–217.

[HTC13] Hao, Y.; Tang, J.; Cheng, Y. “Secure cooperative data downloading in vehicular ad hoc networks”, *IEEE Journal on Selected Areas in Communications*, vol. 31–9, Sep 2013, pp. 523–537.

[IOT] IOTA. “Iota search - transactions overview”. Source: https://iotasear.ch/live-transactions, Aug 2018.

[IOT18] IOTA. “Iota - next generation blockchain”. Source: https://iota.org/,2010, Mar 2018.

[Jef18] Jeffries, A. “Inside the bizarre upside-down bankruptcy of mt.gox”. Source: https://www.theverge.com/2018/3/22/17151430/bankruptcy-mt-gox-liabilities-bitcoin, Jun 2018.

[Jga16] Jgamblin. “Leaked mirai source code for research/ioc development purposes”. Source: https://github.com/jgamblin/Mirai-Source-Code, Dez 2016.

[JGMP14] Jin, J.; Gubbi, J.; Marusic, S.; Palaniswami, M. “An information framework for creating a smart city through internet of things”, *IEEE Internet of Things Journal*, vol. 1–2, Apr 2014, pp. 112–121.

[JLG+14] Johnson, B.; Laszka, A.; Grossklags, J.; Vasek, M.; Moore, T. “Game-theoretic analysis of ddos attacks against bitcoin mining pools”. In: Proceedings of the Financial Cryptography and Data Security, 2014, pp. 72–86.

[JVW+14] Jing, Q.; Vasilakos, A. V.; Wan, J.; Lu, J.; Qiu, D. “Security of the internet of things: Perspectives and challenges”, *Wireless Networks*, vol. 20–8, Nov 2014, pp. 2481–2501.

[KAC12] Karame, G. O.; Androulaki, E.; Capkun, S. “Double-spending fast payments in bitcoin”. In: Proceedings of the ACM Conference on Computer and Communications Security, 2012, pp. 906–917.

[Kas19] Kaspersky. “New iot-malware grew three-fold in h1 2018”. Source: https://www.kaspersky.com/about/press-releases/2018_new-iot-malware-grew-three-fold-in-h1-2018, Feb 2019.

[KBL17] Kountche, D. A.; Bonnin, J. M.; Labiod, H. “The problem of privacy in cooperative intelligent transportation systems (c-its)”. In: Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM), 2017, pp. 482–486.

[KBOS18] Kumar, M. S.; Ben-Othman, J.; Srinivasagan, K. G. “An investigation on wannacry ransomware and its detection”. In: Proceedings of the IEEE Symposium on Computers and Communications (ISCC), 2018, pp. 1–6.

[KFW+15] Kishigami, J.; Fujimura, S.; Watanabe, H.; Nakadaira, A.; Akutsu, A. “The blockchain-based digital content distribution system”. In: Proceedings of the IEEE International Conference on Big Data and Cloud Computing, 2015, pp. 187–190.

[KKS+17] Kwon, Y.; Kim, D.; Son, Y.; Vasserman, E.; Kim, Y. “Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin”. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 195–209.

[KSM13] Kelly, S. D. T.; Suryadevara, N. K.; Mukhopadhyay, S. C. “Towards the implementation of iot for environmental condition monitoring in homes”, *IEEE Sensors Journal*, vol. 13–10, Oct 2013, pp. 3846–3853.

[KYH+17] Kang, J.; Yu, R.; Huang, X.; Maharjan, S.; Zhang, Y.; Hossain, E. “Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains”, *IEEE Transactions on Industrial Informatics*, vol. 13–6, Dec 2017, pp. 3154–3164.

[LHH+18] Lin, C.; He, D.; Huang, X.; Khan, M. K.; Choo, K. K. R. “A new transitively closed undirected graph authentication scheme for blockchain-based identity management systems”, *IEEE Access*, vol. 6, May 2018, pp. 203–212.

[LLC+18] Li, L.; Liu, J.; Cheng, L.; Qiu, S.; Wang, W.; Zhang, X.; Zhang, Z. “Creditcoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles”, *IEEE Transactions on Intelligent Transportation Systems*, vol. PP–99, 2018, pp. 1–17.

[LLL+12] Lu, R.; Lin, X.; Luan, T. H.; Liang, X.; Shen, X. “Pseudonym changing at social spots: An effective strategy for location privacy in vanets”, *IEEE Transactions on Vehicular Technology*, vol. 61–1, Jan 2012, pp. 86–96.

[LLZ+17] Liu, Y.; Liu, X.; Zhang, L.; Tang, C.; Kang, H. “An efficient strategy to eliminate malleability of bitcoin transaction”. In: Proceedings of the International Conference on Systems and Informatics (ICSAI), 2017, pp. 960–964.

[LMNZ18] Lunardi, R. C.; Michelin, R. A.; Neu, C. V.; Zorzo, A. F. “Distributed access control on iot ledger-based architecture”. In: Proceedings of the IEEE/IFIP Network Operations and Management Symposium, 2018, pp. 1–7.

[LSFK17] Li, W.; Sforzin, A.; Fedorov, S.; Karame, G. O. “Towards scalable and private industrial blockchains”. In: Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts, 2017, pp. 9–14.

[LSL+18] Lippert, B. H.; Scheltzke, B. F.; Lunardi, R. C.; Michelin, R. A.; Zorzo, A. F. “Sistema de reputação baseado em blockchain para detecção de fake news”. In: 2018 Workshop Regional de Segurança da Informação e de Sistemas Computacionais (WRSeg), 2018, pp. 1–6.

[Lun18] Lunardi, R. C.; Michelin, R. A. N. C. V. Z. A. F. “A lightweight blockchain for industrial iot - submitted”, *IEEE Transactions on Industrial Informatics*, vol. 1, Dec 2018, pp. 1–6.

[LZSL17] Liang, X.; Zhao, J.; Shetty, S.; Li, D. “Towards data assurance and resilience in iot using blockchain”. In: Proceedings of the IEEE Military Communications Conference (MILCOM), 2017, pp. 261–266.

[MA18] Murthy, M. N.; AjaySaiKiran, P. “A smart office automation system using raspberry pi (model-b)”. In: Proceedings of the International Conference on Current Trends towards Converging Technologies (ICCTCT), 2018, pp. 1–5.

[MAAN19] Makhdoom, I.; Abolhasan, M.; Abbas, H.; Ni, W. “Blockchain’s adoption in iot: The challenges, and a way forward”, *Journal of Network and Computer Applications*, vol. 125, Jan 2019, pp. 251 – 279.

[MDS+18] Michelin, R. A.; Dorri, A.; Steger, M.; Lunardi, R. C.; Kanhere, S. S.; Jurdak, R.; Zorzo, A. F. “Speedychain: A framework for decoupling data from blockchain for

smart cities". In: Proceedings of the International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2018, pp. 145–154.

[MLZZ16] Ma, H.; Liu, L.; Zhou, A.; Zhao, D. "On networking of internet of things: Explorations and challenges", *IEEE Internet of Things Journal*, vol. 3–4, Aug 2016, pp. 441–452.

[MSB+16] Mayzaud, A.; Sehgal, A.; Badonnel, R.; Chrisment, I.; Schönwälder, J. "Using the RPL protocol for supporting passive monitoring in the Internet of Things". In: Proceedings of the IEEE/IFIP Network Operations and Management Symposium, 2016, pp. 366–374.

[MSH17] McCorry, P.; Shahandashti, S. F.; Hao, F. "Refund attacks on bitcoin's payment protocol". In: Proceedings of the Financial Cryptography and Data Security, 2017, pp. 581–599.

[MYAZ15] Mahmoud, R.; Yousuf, T.; Aloul, F.; Zualkernan, I. "Internet of things (iot) security: Current status, challenges and prospective measures". In: 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), 2015, pp. 336–341.

[MZD14] Michelin, R. A.; Zorzo, A. F.; De Rose, C. A. "Mitigating dos to authenticated cloud rest apis". In: Proceedings of the International Conference for Internet Technology and Secured Transactions (ICITST), 2014, pp. 106–111.

[Nak08] Nakamoto, S. "Bitcoin: A peer-to-peer electronic cash system". Source: http://bitcoin.org/bitcoin.pdf, Nov 2015.

[Nak16] Nakamoto, S. "Bitcoin Core Integration/Staging Tree". Source: https://github.com/bitcoin/bitcoin, Nov 2016.

[Nat16] Nath, I. "Data exchange platform to fight insurance fraud on blockchain". In: Proccedings of the IEEE International Conference on Data Mining Workshops (ICDMW), 2016, pp. 821–825.

[New18] Newman, L. H. "The under armour hack was even worse than it had to be". Source: https://www.wired.com/story/under-armour-myfitnesspal-hack-password-hashing/, Dez 2018.

[NG16] Natoli, C.; Gramoli, V. "The balance attack against proof-of-work blockchains: The R3 testbed as an example". Source: http://arxiv.org/abs/1612.09426, Nov 2018.

[NIS19] NIST. “Cryptographic standards and guidelines”. Source: https://csrc.nist.gov/Projects/Cryptographic-Standards-and-Guidelines, Feb 2019.

[OMAA17] Ouaddah, A.; Mousannif, H.; Abou Elkalam, A.; Ait Ouahman, A. “Access Control in the Internet of Things: Big Challenges and New Opportunities”, *Computer Networks*, vol. 112, 2017, pp. 237–262.

[OWA19] OWASP. “The owasp foundation”. Source: www.owasp.org, Jan 2019.

[PBH+08] Papadimitratos, P.; Buttyan, L.; Holczer, T.; Schoch, E.; Freudiger, J.; Raya, M.; Ma, Z.; Kargl, F.; Kung, A.; Hubaux, J. P. “Secure vehicular communication systems: Design and architecture”, *IEEE Communications Magazine*, vol. 46–11, Nov 2008, pp. 100–109.

[PKC+18] Pärssinen, M.; Kotila, M.; Cuevas Rumin, R.; Phansalkar, A.; Manner, J. “Is blockchain ready to revolutionize online advertising?”, *IEEE Access*, vol. 6, Jan 2018, pp. 54884–54899.

[RAH+15] Rajput, U.; Abbas, F.; Hussain, R.; Eun, H.; Oh, H. “A simple yet efficient approach to combat transaction malleability in bitcoin”. In: Proceedings of the Information Security Applications, Rhee, K.-H.; Yi, J. H. (Editors), 2015, pp. 27–37.

[RBB17] Ray, S.; Basak, A.; Bhunia, S. “Patching the internet of things”, *IEEE Spectrum*, vol. 54–11, Nov 2017, pp. 30–35.

[RNL11] Roman, R.; Najera, P.; Lopez, J. “Securing the internet of things”, *Computer*, vol. 44–9, Sep 2011, pp. 51–58.

[Ros11] Rosenfeld, M. “Analysis of bitcoin pooled mining reward systems”. Source: http://arxiv.org/abs/1112.4980, Dec 2018.

[Sar19] Sarang, R. “Trending: Iot malware attacks of 2018”. Source: https://securingtomorrow.mcafee.com/consumer/mobile-and-iot-security/top-trending-iot-malware-attacks-of-2018/, Feb 2019.

[SCZ+16] Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. “Edge computing: Vision and challenges”, *IEEE Internet of Things Journal*, vol. 3–5, Oct 2016, pp. 637–646.

[SI16] Sanda, T.; Inaba, H. “Proposal of new authentication method in wi-fi access using bitcoin 2.0”. In: Proceedings of the IEEE Global Conference on Consumer Electronics, 2016, pp. 1–5.

[SMP17] Sharma, P. K.; Moon, S. Y.; Park, J. H. “Block-vn: A distributed blockchain based vehicular network architecture in smart city”, *Journal of Information Processing Systems*, vol. 13–1, Dec 2017, pp. 184–195.

[SRGCP15] Sicari, S.; Rizzardi, A.; Grieco, L. A.; Coen-Porisini, A. “Security, privacy and trust in Internet of Things: The road ahead”, *Computer Networks*, vol. 76, Jan 2015, pp. 146–164.

[SS16] Singh, S.; Singh, N. “Blockchain: Future of financial and cyber security”. In: Proceedings of the International Conference on Contemporary Computing and Informatics (IC3I), 2016, pp. 463–467.

[TMSB16] Tortonesi, M.; Michaelis, J.; Suri, N.; Baker, M. “Software-defined and value-based information processing and dissemination in IoT applications”. In: Proceedings of the IEEE/IFIP Network Operations and Management Symposium, 2016, pp. 789–793.

[TR17] Teslya, N.; Ryabchikov, I. “Blockchain-based platform architecture for industrial IoT”. In: Proceedings of the Conference of Open Innovations Association, 2017, pp. 321–329.

[Vec11] Vector76. “Fake bitcoins?” Source: https://bitcointalk.org/index.php?topic=36788.msg463391#msg463391, Jul 2018.

[WDWL18] Wu, L.; Du, X.; Wang, W.; Lin, B. “An Out-of-band Authentication Scheme for Internet of Things Using Blockchain Technology”. In: Proceedings of the International Conference on Computing, Networking and Communications (ICNC), 2018, pp. 769–773.

[WFN+16] Watanabe, H.; Fujimura, S.; Nakadaira, A.; Miyazaki, Y.; Akutsu, A.; Kishigami, J. “Blockchain contract: Securing a blockchain applied to smart contracts”. In: Proceedings of the International Conference on Consumer Electronics (ICCE), 2016, pp. 467–468.

[WLL+17] Wang, X.; Li, K.; Li, H.; Li, Y.; Liang, Z. “Consortiumdns: A distributed domain name service based on consortium chain”. In: Proceedings of the International Conference on High Performance Computing and Communications, 2017, pp. 617–620.

[Woo16] Woolf, N. “Ddos attack that disrupted internet”. Source: https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet, Dez 2016.

[Wra19] Wray, S. “Is the iot a trojan horse for smart cities?” Source: https://www.smartcitiesworld.net/opinions/opinions/is-the-iot-a-trojan-horse-for-smart-cities, Feb 2019.

[Xu11] Xu, L. D. “Enterprise systems: State-of-the-art and future trends”, *IEEE Transactions on Industrial Informatics*, vol. 7–4, Nov 2011, pp. 630–640.

[ZBC+14] Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. “Internet of Things for Smart Cities”, *IEEE Internet of Things Journal*, vol. 1–1, Feb 2014, pp. 22–32.

[ZJ18] Zhang, K.; Jacobsen, H. “Towards dependable, scalable, and pervasive distributed ledgers with blockchains”. In: Proceedings of the International Conference on Distributed Computing Systems (ICDCS), 2018, pp. 1337–1346.

[ZLS11] Zhang, Q.; Li, Z.; Song, C. “The improvement of digital signature algorithm based on elliptic curve cryptography”. In: Proceedings of the International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011, pp. 1689–1691.

[ZLWS18] Zhao, S.; Li, Y.; Wang, B.; Su, H. “Research on the blockchain-based integrated demand response resources transaction scheme”. In: Proceedings of the International Power Electronics Conference (IPEC-), 2018, pp. 795–802.

[ZNL+18] Zorzo, A. F.; Nunes, H. C.; Lunardi, R. C.; Michelin, R. A.; Kanhere, S. S. “Dependable iot using blockchain-based technology”. In: Proceedings of the Latin-American Symposium on Dependable Computing (LADC), 2018, pp. 1–6.