Modelling of secure communication system for IoT enabled waste management system

Florian Akos Szabo

Computer Science and Engineering, master's level
2019

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering
This thesis is prepared as part of an European Erasmus Mundus programme PERCCOM - PERvasive Computing & COMmunications for sustainable development.

This thesis has been accepted by partner institutions of the consortium (cf. UDL-DAJ, n°1524, 2012 PERCCOM agreement). Successful defence of this thesis is obligatory for graduation with the following national diplomas:

- Master in Complex Systems Engineering (University of Lorraine)
- Master of Science in Technology (Lappeenranta University of Technology)
- Master in Pervasive Computing and Communications for Sustainable Development (Luleå University of Technology)
Urban expansion is a key driving force of our modern world. Increasing environmental footprint is an example issue that is directly caused by it. The city of St. Petersburg employs on average almost 500 garbage trucks on a daily basis and spends more than 1 million US Dollars every year to collect, process and manage waste. In order for megacities, such as St. Petersburg, to cope with its effects, new ideas are needed. This seems to be an obvious area in which technology can be used to improve current practices and help save resources. In this study, we investigate how the Internet of Things, blockchain and Quantum Key Distribution systems can be integrated to provide a safe and efficient method for improving the waste management process in the context of Smart City projects. Our implemented simulations in Mininet show that there are some clear challenges with regards to the adoption of blockchain technology in an IoT environment. However, the integration of quantum channels and the use of Quantum Key Distribution within the blockchain infrastructure shows good potential for balancing the advantages and disadvantages of blockchain. With the implemented simulations we demonstrate the superior capabilities of the Proof of Infrastructure blockchain solution, which can facilitate secure transactions within the waste management scenario.
ACKNOWLEDGMENTS

First and foremost I would like to say thanks for the generous support of my parents and my relatives. They encouraged me greatly before embarking on this 2 years long journey and I feel immense gratitude towards their love and support throughout this time.

Then I would like to thank the whole PERCCOM consortium and in particular Prof. Eric Rondeau, Prof Jari Porras, Prof. Andrei Rybin and Prof. Karl Andersson for making this program ([1, 2]) and guiding our steps through every semester. A big thank you also goes out to Caroline Schreppf, our very dear PERCCOM secretary who helped me tackle various administrative and other challenges. Merci beaucoup!

I would also like to thank Prof Oleg Sadov for helping me to stay on track during my time in Saint Petersburg. Our weekly meetings provided much-needed guidance and new insights for my whole master thesis project. I also thank Sylvain Kubler for his valuable input and comments during my final semester thesis work. Special thanks to Prof Andrei Rybin and Julia Semenova for helping me with all the necessary arrangements to make my stay in Saint Petersburg comfortable.

Last but not least I would like to thank all my classmates who shared this 2 years long journey with me. The memories of this master’s program will forever stay in my heart and mind.

Thank you all!

Florian Akos Szabo

June, 2019

Skellefteå, Sweden
**TABLE OF CONTENTS**

**ABSTRACT** 2

**ACKNOWLEDGMENTS** 3

**TABLE OF CONTENTS** 4

**LIST OF FIGURES** 6

**LIST OF TABLES** 7

**LIST OF ABBREVIATIONS** 8

1 INTRODUCTION 1

1.1 Background ................................. 1
1.2 The Saint Petersburg Use-Case ........................... 2
1.3 Research objectives ................................. 3
1.4 Delimitations ................................. 4
1.5 Structure of the thesis ................................. 5

2 BACKGROUND AND RELATED WORK 6

2.1 Waste Management ................................. 6
2.1.1 What a Waste: 2.0 ................................. 6
2.1.2 The promise of the Smart City ........................... 8
2.2 Internet of Things ................................. 9
2.2.1 Overview ................................. 9
2.2.2 Security considerations ................................. 10
2.3 Blockchain technology ................................. 18
2.3.1 Overview ................................. 18
2.3.2 Brief introduction to Bitcoin ........................... 21
2.3.3 Nakamoto consensus mechanism ........................... 22
2.3.4 Previous PERCCOM thesis ........................... 27
2.3.5 A case for private blockchains ........................... 28
2.3.6 Challenges ahead ................................. 29
2.4 Quantum Computing ................................. 32
2.4.1 Overview ................................. 32
2.4.2 Traditional Key Distribution ........................... 33
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>UN Sustainable Development Goals</td>
<td>1</td>
</tr>
<tr>
<td>2.1</td>
<td>Internet of Things Overview</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Internet of Things security concerns.</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>IoT Architecture</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>Blockchain Components</td>
<td>19</td>
</tr>
<tr>
<td>2.5</td>
<td>Blockchain High-level Structure</td>
<td>21</td>
</tr>
<tr>
<td>2.6</td>
<td>Proof of Work puzzle</td>
<td>24</td>
</tr>
<tr>
<td>2.7</td>
<td>Blockchain Fork Example</td>
<td>26</td>
</tr>
<tr>
<td>2.8</td>
<td>Fork resolution mechanisms</td>
<td>27</td>
</tr>
<tr>
<td>2.9</td>
<td>Diffie-Hellman Key Exchange</td>
<td>33</td>
</tr>
<tr>
<td>2.10</td>
<td>No-Cloning Theorem for QKD</td>
<td>36</td>
</tr>
<tr>
<td>3.1</td>
<td>Overview of Design Science Methodology</td>
<td>39</td>
</tr>
<tr>
<td>3.2</td>
<td>Overall Architecture</td>
<td>41</td>
</tr>
<tr>
<td>3.3</td>
<td>High-level flow chart for waste management scenario</td>
<td>42</td>
</tr>
<tr>
<td>3.4</td>
<td>Class hierarchy for BAL Python project</td>
<td>43</td>
</tr>
<tr>
<td>3.5</td>
<td>BAL QTopo.py example</td>
<td>46</td>
</tr>
<tr>
<td>3.6</td>
<td>Mininet Tree topology example</td>
<td>47</td>
</tr>
<tr>
<td>3.7</td>
<td>Proof of Work scenario architecture</td>
<td>50</td>
</tr>
<tr>
<td>3.8</td>
<td>Proof of Infrastructure scenario architecture</td>
<td>52</td>
</tr>
<tr>
<td>4.1</td>
<td>PoW Difficulty’s effect on energy profile</td>
<td>55</td>
</tr>
<tr>
<td>4.2</td>
<td>PoI Energy Profile with different TX intensity</td>
<td>56</td>
</tr>
<tr>
<td>4.3</td>
<td>Power measurements of PoW and PoI</td>
<td>58</td>
</tr>
<tr>
<td>4.4</td>
<td>Delay versus throughput for PoW</td>
<td>59</td>
</tr>
<tr>
<td>4.5</td>
<td>Delay versus throughput for PoI</td>
<td>60</td>
</tr>
<tr>
<td>4.6</td>
<td>Blockchain metrics of PoW and PoI variants</td>
<td>60</td>
</tr>
<tr>
<td>4.7</td>
<td>Sustainability analysis via the Pentagon method.</td>
<td>62</td>
</tr>
<tr>
<td>A.1</td>
<td>Sequence diagram for Proof of Infrastructure scenario</td>
<td>74</td>
</tr>
</tbody>
</table>
LIST OF TABLES

2.1 Pros and Cons of QKD ................................................. 37
3.1 DSRM process in the context of this study. ......................... 40
LIST OF ABBREVIATIONS

BAL  Blockchain Alchemy Lab
CAD  Computer Aided Design
EU   European Union
FW   Firmware
GDPR General Data Protection Regulation
GHOST Greedy Heaviest Observed Subtree
GPRS General Packet Radio Services
GPS  General Positioning System
HDD  Hard Disk Drive
HTTPS Hyper-Text Transfer Protocol Secure
HW   Hardware
IDS  Intrusion Detection System
IoT  Internet of Things
IPS  Intrusion Prevention System
IPSec Internet Protocol Security
LoRaWAN Long Range Wide Area Network
MITM Man In The Middle
PoI  Proof of Infrastructure
PoW  Proof of Work
QIS  Quantum Information Science
RSA  Rivest Shamir Adleman
SDG  Sustainable Development Goal
SGB  Smart Garbage Bin
SGT  Smart Garbage Truck
SSD  Solid State Drive
SSH  Secure Shell
SW   Software
UN   United Nations
UTXO Unspent Transaction Output
WWW  World Wide Web
1 INTRODUCTION

In this chapter, the background of the thesis is introduced, and illustrated by the use-case of the Smart Waste Management scenario in the city of Saint Petersburg, Russia. Furthermore, the research objectives and delimitations are articulated, followed by a short discussion on methodology and an overview of the thesis structure.

1.1 Background

The UN General Assembly, in 2015, established 17 targets, also known as Sustainable Development Goals or SDGs, which aim to identify the biggest challenges humanity faces in the present [3]. These SDGs cover a wide range of social and economic development issues including poverty, hunger, health, education, global warming, gender equality, water, sanitation, energy, urbanization, environment and social justice.

Each SDG contains a number of Targets and Indicators, which can provide crucially important data to help direct efforts aimed at solving issues within the context of individual goals.

In particular, SDG Number 11 concerns urban areas and big cities, with the specific goal of making them more inclusive, safe, resilient and sustainable. The Targets in SDG
11 include, with various Indicators within each for enabling progress tracking at a more granular level:

1. Safe and affordable housing
2. Affordable and sustainable transport systems
3. Inclusive and sustainable urbanization
4. Protect the world’s cultural and natural heritage
5. Reduce the adverse effects of natural disasters
6. Reduce the environmental impacts of cities
7. Provide access to safe and inclusive green and public spaces
8. Strong national and regional development planning
9. Implement policies for inclusion, resource efficiency and disaster risk reduction
10. Support least developed countries in sustainable and resilient building

As the title suggests, this thesis report reflects our efforts towards improving Waste Management in a Smart City scenario, so it is understood to be in the context of Target 6: Reduce the environmental impacts of cities.

1.2 The Saint Petersburg Use-Case

Nowadays it is commonly known fact that the driving forces behind urbanization are creating some of the biggest cities that ever existed on Earth. This process is undeniably putting a lot of pressure on city officials who are in charge of infrastructure and various services in these cities. According to a 2017 report by the UN on SDG progress [4], around 54% of Earth’s population (close to 4 billion people) have already lived in cities by 2015. Their projection is that by 2030 this is expected to increase to 5 Billion. This rapid urbanization has created some of the densest cities in the world, and such cities cannot achieve sustainability without an underlying infrastructure that delivers the basic utilities such as heat, electricity, and water.

As mentioned by the UN report on SDG progress, the safe removal and treatment of solid waste is a critical service. Uncollected or not properly handled waste can cause serious issues such as flooding, and various diseases. As identified by the authors in [5], IoT emerges as a viable solution for improving waste collection and management. The vari-
ous use-cases identified include smart route planning for waste collection or embedding sensors in garbage bins and collecting data in real-time about their fill level. Implementing such systems can offer good returns on investment by reducing the number of trucks necessary to perform waste collection which ultimately improves their utilization.

As an example, the city of St. Petersburg in Russia, which has a population of about 5 million, utilizes more than 450 garbage trucks to collect waste on a daily basis. This service costs the city more than 1 million US dollars and generates a lot of congestion and air pollution. In order to increase the efficiency of such a system and make progress towards Target 6 of SDG 11, IoT-enabled Smart Garbage Bins (SGB) have been proposed to be deployed in the city.

1.3 Research objectives

The main focus of this study is to investigate how the security of communication channels of such an IoT-enabled Smart Waste Management system can be ensured through the use of various technologies, such as Blockchain and Quantum Key Distribution.

Once a Smart Waste Management system gets deployed in a city, such as St. Petersburg, it becomes very important to make sure that this system is up and running without disruption. In particular, it becomes critical to ensure that it is resilient to malicious attacks and hackers so that its operation can be ensured.

The following research questions aim to serve as milestones towards a successful implementation of this study:

Ø Q1: How can the communication patterns of an IoT enabled Waste Management system be modeled?
Ø Q2: How can Blockchain technology be useful in order to make this model more secure?
Ø Q3: Can QKD be integrated with blockchain to further enhance the communication security of such a system?

In order to answer these questions a number of objectives are defined below, which
are later on reflected in the methodology:

- State of the art survey: investigate the available literature on IoT, Blockchain and Quantum Key Distribution, with a special focus on the Waste Management scenario
- Define simulation scenarios involving Blockchain and QKD.
- Implement and Evaluate the implemented scenarios.
- Draw conclusions about the feasibility of different technologies.

1.4 Delimitations

Security is, of course, a very broad term that can encompass a lot of different scenarios and technologies. This study focuses on modeling secure communications, to enable the proper functioning of the system, as it was designed. As such, there are certain aspects of security which are out of scope for this study:

- security of collected data throughout the operation of the system
- network security of the SWM infrastructure
- host security of the servers of the SWM infrastructure
- perimeter security of physical devices and objects
1.5 Structure of the thesis

The structure of this thesis is comprised of 5 chapters which are organized as follows:

- **Chapter 1** provides an overview, research goals, motivations, objectives, delimitations, and description of the methodology.

- **Chapter 2** presents the background and state-of-the-art literature on the Internet of Things, Blockchain and Quantum Key Distribution.

- **Chapter 3** introduces the details of the design and implementation of the framework with an emphasis on Mininet architectures and overall topology. This chapter includes most of the technical implementation details.

- **Chapter 4** presents the results of the study by focusing on the different metrics and blockchain characteristics of the simulations in Mininet. Furthermore, this chapter includes a discussion of the results.

- Finally, the study concludes in **Chapter 5** with a discussion and recommendations for future work.
2 BACKGROUND AND RELATED WORK

This chapter starts with a general overview of the waste management problem and places it into the context of Smart Cities. Further on, the chapter is divided into three subsections, each dealing with different aspects of the study, (a): Internet of Things and its security implications, (b) Blockchain technology and its underlying consensus mechanisms, and (c) Quantum Key Distribution. In each of these subsections, existing literature will be investigated to study the characteristics, use-cases, and applications of the corresponding technology. With this literature review, I aim to study available resources that relate to these technologies and establish a body of knowledge on which my thesis project work can be based.

2.1 Waste Management

This subsection will cover solid waste management as a use-case for a modern IoT-enabled smart city scenario. Literature is reviewed to understand the various approaches and efforts that have been pursued.

2.1.1 What a Waste: 2.0

A report on waste from the World Bank Group [6] claims that waste generation is projected to outpace population growth, which means there will be an ever-growing need to manage waste more efficiently. Considering the projections that urbanization is expected to accelerate and drive more and more people to dense cities, it becomes crucial to design a system that is able to cope with the problems that arise if waste management is not implemented properly.

The World Bank report emphasizes that waste affects everyone, but poorly managed waste has a substantially greater negative impact on the most vulnerable portions of society, those people living in slums and low-income neighborhoods of big cities around the world.
The report estimates that approximately 1.6 billion tonnes of CO$_2$ emissions were generated from solid waste management practices around the globe in 2016. This accounted for about 5% of all emissions. By 2050, they estimate this will increase to 2.6 billion tonnes of CO$_2$ annually. This highlights the critical importance of making our waste management practices as efficient as possible so that the emissions can be minimal if not eliminated completely.

The most common ways of waste collection involve a truck with several workers driving door-to-door to collect waste from individual dwellings. This is improved marginally when there are known collection centers in neighborhoods where people take their trash and the collection trucks come only to this location.

Technology and data play a crucial role in many areas of life to help us make better decisions and waste management is no exception. The flow of this data and the technology behind the infrastructure, however, can vary greatly. In more affluent neighborhoods a push notification on the smartphones of people may be sufficient to disperse information about an incoming collection truck, however, in poor and illiterate districts, there may be a need to deploy radio-transmission units and loudspeakers that can communicate the same information to local residents.

The World Bank compiled a list of available information that can be collected and utilized in various ways to improve the efficiency of current practices.

- Real-time vehicle locations and routes
- Weight of disposed waste per location
- Emissions of vehicles and landfills
- Records on transactions (B2B, user)
- Household waste collection history
- Citizen feedback
- Inventory of facilities and equipment
2.1.2 The promise of the Smart City

After the initial search for literature on this topic, it seems clear that there is already a considerable amount of articles and papers on the intersection of 3 standalone fields in research: (a) Internet of Things, (b) Smart City and (c) Waste Management. In this short subsection, a few of these works are summarized to give a brief overview of current and past efforts in these fields.

The majority of proposals involve some kind of sensor devices being installed into/onto garbage bins with various types of communication capabilities so that they can report various data, such as fill level, and signal to the central waste operator that it should be emptied soon. The operators usually deploy a cloud or fog-based management system which can collect and digest the measurement data and use this for route planning of waste collection trucks among other things.

A survey of various models of smart waste management in contemporary literature is reported in [7] from 2017. The paper presents a clear overview of the topic and establishes a structured taxonomy that creates a clear understanding of categories, components, and functions in the context of waste management via IoT in smart cities.

Their taxonomy is then used to provide a survey of literature up until 2015, in which they identify various categories of papers. Some focus only on the use of actuators and WSNs, while others use various kinds of sensors, such as RFID, capacity, pressure, weight temperature, humidity, and chemical sensors, etc.

Authors in [8] in 2016 proposed to use a Raspberry Pi mounted on a traditional garbage bin and equipped with an active RFID sensor, while the authors in [9] focused on the communication technology and chose LoRaWAN as their means of transport. This helps them to achieve ultra low-cost deployments and low-power operation for their scenario.

In the paper [10] the authors propose to take a citizen-centric approach to combine GPS/GPRS technology with sensor-equipped garbage bins to provide the necessary services to the involved parties. In [11] the authors propose to combine waste level monitoring via ultrasonic sensors and machine learning algorithms processing the collected data to
predict future patterns and optimize waste collection.

2.2 Internet of Things

2.2.1 Overview

The Internet of Things is often said to be the inevitable consequence of Moore’s Law throughout many years of development, manifesting in an ever-increasing density of transistors on silicon chips and integrated circuits. This eventually means that overall computing power per the same area of a microchip doubles roughly every two years. As a result of the size of microchips dramatically decreases for the same amount of computational power, which means that embedding microcontrollers in small objects become increasingly possible. This fact, coupled with advances in battery and storage technology, is what enables the Internet of Things to transform industries and various practices. The ultimate goal of IoT is to build an intelligent world based on analysis of data collected from deployed IoT devices [12].

Figure 2.1: Internet of Things Overview. (Source: Intel Inc.)

The IoT phenomena has been an emerging area in IT for many years [13], but attention has increased drastically over the last 5-10 years. This is thanks to IoT’s promise to be the key enabler for many new types of scenarios and applications. IoT can increase
the utility of objects of the physical world by augmenting them with computational and communication capabilities.

In recent years IoT has received a great deal of attention thanks to all the potential use-cases it can be adapted to, such as environmental monitoring, smart agriculture, home automation, autonomous vehicles, building efficiency management, industrial automation, smart cities, security and surveillance, wearables, health monitoring, and smart grid.

When it comes to the Waste Management scenario that is the basis of this study, the benefits of integrating IoT can be substantial. Organizing the resources necessary to collect and manage waste on such a big scale takes a great deal of planning and consumes lots of resources. Using IoT technology to make this whole process more efficient has been studied previously in [5], where the authors conclude that equipping trucks and garbage bins with IoT sensors in order to collect data can help optimize processes and save resources. Improving the waste handling process to make it more resource efficient can help us achieve more sustainable living conditions in big cities, and thus contribute towards Goal 11 of the SDGs.

However, if such an IoT-enabled system gets compromised and the service gets interrupted, the consequences can be considerable. It poses a direct threat to the safety and health of humans by disrupting waste collection, which can lead to garbage piling up in unhealthy amounts on city streets. Therefore it is very important to design such a system in a way that is resistant to malicious attacks from the network.

2.2.2 Security considerations

As an increasing number of physical objects get connected to the Internet, there is, at the same time, a growing potential for these systems to be hacked, if they are not properly designed from a security perspective. As devices get connected to the Internet, the Cyber World gets extended to the real world, with potentially serious consequences, if security is not properly handled.

Despite the immense potential of the Internet of Things, it also carries within the potential of malicious attacks originating in the cyber world and spreading to the real world
by having Internet-connected objects actuate real-world processes. Authors in paper [12] argue that failure of properly handling these new types of risks may result in missed opportunities and hindered realization of the benefits that IoT can facilitate.

Judging by the amount of IoT related incident disclosures reported in the media, it is safe to conclude that most IoT vendors do not consider the security of their IoT products during its design phase. This is most likely because these companies are usually always in a race against the clock, trying to be first to market so that they can reap the profits promised by forecasting companies.

The authors of paper [14] point out that one of the major requirements for IoT before widespread adoption can be realized is security. The paper groups security considerations of IoT into two main groups: (a) security functionalities or tasks such as confidentiality, integrity, authenticity, and authorization; and (b) design metrics that are necessary to enable IoT, such as cost, size, latency and most important of all is energy consumption.

Furthermore, the authors also point out that some requirements may be in direct contradiction with one another, such as the strength of authentication and privacy. This means that these parameters will have to be carefully considered during and after deployment.

In some scenarios, the physical security of devices can also be a critical issue, since it may be required to deploy IoT devices in certain areas where physical access to the device cannot be prohibited. Authors in [15] present some ideas that can be implemented in HW that can prevent unauthorized access and tinkering with deployed IoT devices.
Figure 2.2 summarizes different layers of security as it relates to different aspects of IoT architecture. The various security concerns are:

- Physical security of devices
- Secure communication of data to the IoT Gateway
- Secure transmission of data towards the Cloud
- Security of data at rest in the cloud

The authors in [14] claim that designing for the IoT has to go through a paradigm shift, in which manufacturers of micro-controllers, that enable these chips, should focus on security as a first-class requirement within their CAD process. As the microchip industry was progressing their focus was often shifting between increasing the number of transistors, delay, and energy consumption of their products. To help make IoT devices more secure, chip manufacturers should build security into their chips by design, possibly via some extra hardware components. Hardware-assisted security can enable trusted information flows, which would increase the adoption of IoT solutions in various domains.

Another key issue is in the nature of embedded IoT devices that are often operated from a battery. Due to the constraints put on the processing and storage capabilities of some IoT devices, complex security features and algorithms cannot be implemented in them, because it would cancel out their usefulness. As a consequence, either it becomes necessary to offload security related computation to a trusted entity, or new security algorithms will need to be invented, that are better suited for constrained devices.

As an example, certain digital signature algorithms are more expensive to compute (such as the RSA digital signature algorithm), while others are more resource efficient (ECDSA). Implementing the more efficient signature algorithm in hardware to increase the overall security of the chip may a good solution. One drawback of this approach is that if a bug is uncovered in the underlying algorithm that was implemented in hardware, then it becomes essentially impossible to fix it in the deployed IoT devices since the HW cannot be changed with a remote FW upgrade.

In some scenarios, for example under the jurisdiction of GDPR in the EU, it may be required to offer the possibility of IoT data revocation. In the paper [14], the authors point out that encryption can essentially be used for this purpose. Once data gets encrypted and
the key is deleted, it is essentially rendered useless and can be considered impossible to recover.

![Diagram of IoT Architecture](image)

**Figure 2.3**: Different layers of the Internet of Things architecture. Source: [12].

The authors of [12] argue that one of the biggest challenges in securing IoT besides the energy and cost constraints is the enormous device heterogeneity. The vast number of different devices produced by manufacturers who may employ proprietary protocols and create walled-garden ecosystems means that implementing standard security solutions may not be possible in certain IoT scenarios. Another consequence of device heterogeneity is that device constraints can take a very wide spectrum. Due to these constraints, IoT devices are a much easier target for malicious hackers.

The IoT architecture depicted in Figure 2.3 provides a good overview of different layers in general. Security is the biggest concern in the bottom ”Things Layer” as these are the actual IoT devices that are usually constrained with regards to computation, storage, and energy availability.

The Cloud Layer is on the other hand well-provisioned with resources and generally there are no constraints. This means that securing the systems in this layer is less of a problem. In the middle of Figure 2.3, the Edge layer can be seen with various devices such as IoT gateways. These devices are typically un-constrained and are usually positioned one-hop away from IoT end devices, therefore, according to the authors in [12], it
becomes viable to offload some resource-intensive tasks that aim to improve the security of IoT systems.

In two similar papers [16, 17] the authors propose the integration of Blockchain with IoT to handle secure upgrading of the firmware in a distributed and secure manner. They claim that the use of Smart Contracts can facilitate the automatic and integrity-preserving update of IoT which can eliminate the need for central repositories that need to be scaled to handle millions of deployed IoT devices.

The authors of paper [12] emphasize that security in traditional networked computer systems remains challenging to this day, however IoT devices, as a result of some of their characteristics, bring to the table a new set of challenges in the security domain. These challenges are:

- **Integration with the physical world**
  The coupling between cyber and physical worlds creates new kinds of threats that weren’t possible before. Control messages usually flow from Cloud or Edge towards the Things Layer, therefore it’s essential to secure those layers as well, not just the Things Layer. Compartmentalizing can help achieve this by preventing the propagation of a security breach from one layer to the others. Authentication and access control mechanisms play a key role in achieving this.

- **Heterogeneous devices and communication stacks**
  Different types of IoT devices may have very different profiles when it comes to processing and communication capabilities. Thus, traditional security solutions do not apply in most cases. As an example, IP-based security monitoring solutions such as IPSec, HTTPS or SSH will not work for IoT devices that are not equipped with the TCP/IP stack of traditional networks. The challenge of improving IoT security lies in adapting existing security solutions to the HW and SW requirements of the IoT devices, as well as coming up with new and novel solutions. A possible way to improve the security of non-IP based IoT devices is by relying on the gateways which connect them to the Internet, to provide security features.

- **Privacy**
  This is a tricky issue to solve because there is usually a tradeoff between security and privacy. Users would like to use smart IoT solutions, for example for home automation
or health tracking, but they also like to keep their data private. Breaching the security of health monitoring systems can potentially reveal sensitive medical data; while data from home automation systems can reveal a great deal about daily activities of home owners, which they may not realize was possible when their IoT solutions were deployed. How to strike the right balance between security and privacy is a question that is still open for more research.

ø **Immense scale**

Performing security-related tasks, such as key management, post-deployment system upgrade, and administration become immensely difficult at the scale of IoT deployments, that are envisioned for the future. Managing IoT the same way as traditional networked systems become impractical. As an example, the Mirai botnet from 2016 [18], that was targeting IoT devices such as IP-enabled cameras and home routers, resulted in massive targeted DDoS attack on DNS services. The malware exploited default factory credentials in many of the IoT devices that became infected, and the fact that these systems were vulnerable comes partially from the fact that it’s difficult to update them and remove default credentials. Authors of [12] argue that security solutions should be scalable, distributed and most importantly as automatic as possible, to be able to mitigate attacks such as the Mirai malware from 2016.

ø **Trust management problems**

There are a number of IoT scenarios that work in a Peer To Peer (P2P) or ad-hoc mode, which means that trust management in such scenarios remains a challenging topic. The lack of central and often trusted entity, high mobility, lack of globally unique identifier, and temporary relationships make it difficult to manage trust in IoT.

ø **Lack of security expertise**

Lack of awareness of cyber-security issues in IoT Manufacturers often aggravates the problems previously discussed. Factory default credentials widely known and available on the Internet are one such example of a lack of awareness of security best-practices. How can IoT designers, manufacturers, users and administrators be efficiently educated needs to be further examined?

ø **Resource constraints**

Several high-level goals have been defined by NIST to preserve confidentiality, integrity, and availability of data in traditional networked systems. The methods to
achieve those goals include encryption, authentication, access control, intrusion detection, and prevention, etc. The constrained nature of many IoT devices however often prevents the use of these mechanisms, because they were not designed with resource constraints in mind. That means that authentication through PKI that uses asymmetric cryptography may be unavailable for some devices. Other devices, such as RFID, are so constrained that even shared secret cryptography (symmetric) is prohibitively too expensive. Key distribution and key management are also key issues, because they may rely on asymmetric crypto that’s prohibitively expensive for IoT.

Authors in [12] conclude that security should be designed into the foundational architecture of IoT, rather than bolted on as an add-on post-deployment. To help facilitate this architectural shift in IoT security, the paper proposes 3 areas of focus:

1. **End-to-End security built into Things**: It can be achieved by tailoring existing protocols to the needs of IoT, such as IPv6 and 6LowPAN, or by increasing available resources on IoT devices, which enables the use of traditional protocols. A key advantage of End-to-End security is that there is no reliance on a trusted third-party, which reduces the risk of compromise by subverting this third-party.

2. **Security as a Service at the Edge Layer**: Security related computation and communications are offloaded from resource-constrained IoT devices to more powerful Edge Layer devices which act on their behalf. This inherently means that those Edge Layer devices need to be trusted. An example architecture for such a service is proposed in [19] called EdgeSec. It consists of seven major components that work together to systematically handle specific security challenges in IoT systems. Deploying such solutions can not only help with security but it can also improve real-time performance, as heavy computations can be offloaded to a device that is very near to the IoT Things Layer.

3. **Distributed security solutions**: Taking the previous bullet-point one step further, it may be necessary in certain cases to involve not only the Edge Layer but also the Cloud in the process of authentication and security in general. Some Edge Layer devices may lack the necessary resources to fully complement the IoT devices with their security needs. Particularly, Intrusion Detection and Prevention Systems (ID-S/IPS) may be better handled at the Cloud Layer, as resources are more abundantly available there. One issue that arises with offloading security functions to the Cloud
is the distance in terms of network hops; while the Edge Layer is usually 1 hop away from the IoT devices, the Cloud can be several hops and this could significantly increase the latency, to the point that it becomes unfeasible for certain use-cases.

In conclusion, the authors of [12] point out that End-to-End security should be preferred if the IoT devices are powerful enough to handle the resource requirements of the relevant algorithms. Otherwise offloading should be considered, but it remains an open research question as to how to distribute the offloading of functions between Edge and Cloud layers.
2.3 Blockchain technology

2.3.1 Overview

With the introduction of Bitcoin [20] in 2008, cryptocurrencies and their underlying technology, the so-called blockchain, was introduced to the general public. Since then this field has attracted a lot of attention from both industry and academia, especially since the value of Bitcoin soared the unforeseen heights in December 2017.

Nowadays there are several 10s of hundreds of standalone cryptocurrencies in existence. This is mostly thanks to the open-source nature of Bitcoin, which has allowed everyday people to clone and establish their own version, just by tuning and changing some configuration parameters of the original Bitcoin specification.

With this proliferation of alternatives of Bitcoin (a.k.a. alt-coins), there was a natural hype around blockchain in general, which also fueled a wave of academic and scientific projects exploring the use of blockchain technology for various purposes. A simple search query on Google Scholar reveals that as of May 2019 there are 55200 results for the search term "blockchain".

In general, there is a lot of misconceptions out there, about how exactly Bitcoin’s underlying technology, the blockchain functions. In essence, it makes use of some general cryptographic functions to form a linked list style data-structure that is resistant to tampering if used in a public distributed permissionless setting. There is, however, nothing special about blockchain which makes it exclusively suitable for crypto-currency applications.

According to [21], who presented at a recent blockchain conference called Blockchain North in Boden, Sweden, the majority of projects claiming to use blockchain for various use-cases are non-viable and their claims invalid. To simplify the understanding of different blockchain projects, he proposed the dissection of components into 3 Hard and 3 Soft categories, as seen below on Figure: 2.4

The basic idea of this conceptualization of different building blocks of the blockchain is that one can essentially create an almost endless variety of "blockchain solutions", by
selecting and combining various pieces from each category into the basket. Furthermore, even if one picks the same components over and over, it is still feasible to create different blockchain variants by tweaking various configuration parameters. Below an example list of components can be seen:

- **Consensus Model**: the decision model used to reach an agreement [*Schelling point, Randomization, Delegation, Round-Robin, Master/Slave*]
- **Consensus Algorithms**: A rule-set creating the process for reaching the defined agreement model [*Proof of Work, Proof of Stake, Proof of Elapsed Time, pBFT*].
- **Database Models**: Representation of the agreed state changes on persistent storage [*Hash-Linked list, Directed Acrylic Graph, Accounts vs Unspent Transaction Outputs, etc...*].
- **Administrative**: Deals with managing transactions and actors of the blockchain network [*Authentication, Role assignments, Access control, Network control*].
- **Governance Methods**: Provides authority over decision rights [*Transaction policy, Protocol updates*].
• **Incentives**: Defining rule-sets for rewarding honest and punishing dishonest behaviors \([\text{Mining reward, Mining fee}]\).

Once all the pieces are selected, a complete entity emerges that may or may not make sense. As an example, the Bitcoin project consists of the following components according to [21]:

1. Consensus model: Schelling point (Energy)
2. Consensus algorithm: Proof of Work
3. Database model: HashLinked List & UTXO
4. Administrative: PKI On-chain, open membership
5. Governance methods: Opt-in rule set
6. Incentives: Block reward and miner’s fee

The other major blockchain project besides Bitcoin is Ethereum [22], which differs, in that it replaces the UTXO model with an Accounts based model for keeping track of account balances.

Both Bitcoin and Ethereum have an estimated market capitalization measured in tens of billions of US dollars, so it is fairly safe to state that both projects are successful. However, Peter Altmann in his presentation [21] argues that over 99% of blockchain projects are non-viable due to some constraints that make the underlying combination invalid or useless. This could be for example implementing a closed or permission blockchain where all participants are known and pair it with Bitcoin-style Proof of Work consensus algorithm. This combination just does not make sense from a practical perspective.

It is also worth mentioning that Bitcoin and Ethereum are quite different in other regards as well, not just in the fact that they use different components from the categories of building blocks introduced in Figure 2.4. Bitcoin aims to be a decentralized cryptocurrency that helps to store value, while Ethereum aims to be a decentralized application platform, with integrated support for cryptocurrencies.
2.3.2 Brief introduction to Bitcoin

Bitcoin at the basic level works as a decentralized peer-to-peer (P2P) network of nodes running the bitcoin protocol processing transactions and organizing them into blocks that are sealed with cryptographic primitives. The below figure illustrates from at a high-level how the blocks store and organize transactions.

![Figure 2.5: High-level overview of Bitcoin’s hash-pointer data structure which provides the ability to detect changes in the blocks.](image)

Figure 2.5 represents a simplified high-level depiction of the choices made by Bitcoin’s creator for storing the transaction history of the crypto currency. Every block has a block header that stores information in a key-value structure.

First, there is a general index counter which increments block by block, starting from 0. For reference, the current height is around 577423 as of the writing of this thesis. Next, there is a HASH pointer field containing the hash of the previous block. In the case of the genesis block this is NULL (or rather it’s full 0-s). Following the hash pointer, there is a time-stamp which marks the creation of the block. Next, there is another HASH pointer which contains the root of the Transaction Merkle Tree, which is for ensuring that no transaction can be modified in the block without being able to detect it. Last but not least there is a Nonce value in the Block Header, which is used for the Proof of Work mining algorithm.

Following the block header, there are a bunch of transactions listed, which were verified and deemed to be valid by the miner who created the block. On average there are around 3000 transaction in every block. This can vary depending on the complexity of individual transactions. On the conceptual level, thanks to the UTXO mode, every trans-
action in every block has a certain amount of inputs and outputs. Inputs are references to previous transaction outputs (hence the name Unspent Transaction Output). This way there is no need to keep track of account balances, instead, it can always be calculated by summing up all the UTXO-s that belong to a given public key that belongs to an entity.

For this UTXO model to work, miners who validate transactions will need to keep track of all transaction outputs in their memory. In order to use a UTXO, the entity who originates the transaction that references that UTXO needs to be in possession of a private key that can be used to unlock it and sign a new transaction that uses the mentioned UTXO.

2.3.3 Nakamoto consensus mechanism

Reaching consensus in decentralized and distributed computing systems has been a fundamental issue that has been studied well before Bitcoin popularized blockchain. According to the author of [23], blockchain systems resemble a replicated state-machine and are aimed at solving the consensus problem of distributed systems. The classic model of consensus according to the Byzantine Generals problem relies on three properties that need to be satisfied:

- **Agreement**: this requires that there must not be any two processed or nodes which decide on different blocks.
- **Termination**: this requires that all correct processes or nodes must decide on a block eventually.
- **Validity**: this requires that the chosen block is proposed by some process or node that is valid.

Any application or algorithm that satisfies these three requirements is said to solve the Byzantine Consensus problem. However, public permissionless blockchain projects, such as Bitcoin, operate over the Internet, which at its core offers only asynchronous communication services. This means that there is no defined bound on the message delivery times (also known as *best-effort*). Unfortunately, however, thanks to a paper [24] from 1985, it is already known that consensus is unsolvable in asynchronous networks such as the Internet. As a way around this problem, several blockchain projects trade the guar-
accepts of Byzantine Consensus in favor of more probabilistic ones that take advantage of randomization.

According to [23], this randomization helps bypass impossibility by guaranteeing probabilistic results instead of deterministic ones. In case of Bitcoin, the probability of agreement about a proposed block increases exponentially as the length of the blockchain grows. This is why for example, a transaction is considered to be confirmed no when it gets included in a newly minted block, but when the blockchain grows at least 6 more blocks. These six new blocks increase the probability for agreement and thus provides confidence about the included transactions.

Bitcoin has popularized the Proof of Work consensus which has been adopted in many other blockchain projects, but there are plenty of others that are worth studying. From a high-level perspective, the Proof of Work puzzle has two consequences: (a) first it lets the network select the next miner, in a globally random fashion, who gets to propose the next block that will extend the blockchain; and second (b) it helps to protect against the Sybil attack, which happens when malicious actors create fake identities to try to act as legitimate participants in a consensus process, that they try to undermine.

The above described Nakamoto-style PoW consensus is achieved by using a cryptographic puzzle that is resource intensive to solve. Once a solution is found, the node who first found it gets to propose the next block and will then broadcast it to other nodes who will accept it after validation.

This cryptographic puzzle that miners have to solve seems complex at first, but it’s nothing more than a simple one-way hash function that has to be called on the new block being proposed by the miner, and the end result of the hash function usually have to be smaller than a certain value (this is equivalent to the notion of having a certain amount of leading ’0’ characters in the hexadecimal format of the hash value). In order to solve the puzzle the miners have a Nonce value that they can increment until the resulting blockhash satisfies the criteria.

A crucial feature of this PoW-style crypto puzzle is that it has to be difficult to solve but easy to verify. Hash functions provide a convenient tool for this, as they are easy to compute, but the miner has to execute it over and over again with different Nonce values.
Figure 2.6: Details of Proof of Work puzzle. In each round of the algorithm an integer is incremented in the proposed block’s header, which is then sent through the hash function. If the output of the hash function satisfies the difficulty requirement the algorithm is stopped and the solution is announced.

to find the one that produces the desired Hash value.

Hash functions also have to satisfy a few requirements in order to be usable in this puzzle. It must offer Pre-Image Resistance, which is a fancy way of saying it has to ensure that it’s a proper trapdoor or one-way function. This means that given the hash value \( h \) of some unknown input \( m \), it should be very difficult to find any input such that the hash function \( H \) gives the same output as the original hash (2.1).

\[
H(m) = h \quad (2.1)
\]

\[
H(m_1) = H(m_2) \quad (2.2)
\]
Architectures of Blockchain Protocols

The other very important requirement for the Hash function that is used in cryptocurrencies is Collision resistance. This means that given the same Hash function \( H \), it should be very difficult to find two inputs \( m_1 \) and \( m_2 \), such that the Hash function results in the same output for both inputs (2.2).

Once these two requirements are satisfied, then it becomes possible to use a hash function for the purposes of this puzzle because it becomes near impossible to cheat. The only known way to solve it is brute-forcing the Nonce value until the desired hash output is found. This is how Proof of Work gets its name, as it requires miners to invest considerable computational power in order to solve the puzzle. Owing to the popularity of Bitcoin, its underlying Proof of Work mechanism has been adopted in many other blockchains and cryptocurrency projects, such as Ethereum, Bitcoin Cash (forked from Bitcoin), Litecoin, Monero, Dogecoin, just to name a few.

Since blockchain projects such as Bitcoin are usually used in a distributed peer-to-peer manner, it is entirely possible, and actually quite common, that two individual processes in the network create the next block at exactly the same time, creating a so-called fork in the blockchain.

Figure 2.7 details shows the details of this phenomena. As it can be seen on the figure until Block X everything is straightforward. Then for the next proposed block, two contenders arrive from two different miners. Now the blockchain has forked and it is the job of the consensus protocol to resolve it. In the case of Bitcoin’s Nakamoto Consensus mechanism, this is resolved by the nodes choosing to always work on the longest possible chain, that has the most amount of work put into it. This means that when the length of the two forks is equal, nodes can choose randomly. But as soon as the next block arrives and there is no contender this time, the tie will be broken by the node who proposes this new block (Step 3 on Figure 2.7.)

Essentially, the Bitcoin Consensus mechanism can be reduced to the combination of the Proof of Work puzzle with the algorithm that selects the main branch from the global blockchain (which is defined in [23] as the union of all forks). This main branch selection algorithm is where Bitcoin and Ethereum differ slightly. Ethereum uses GHOST (Greedy Heaviest Observed Subtree), which instead of selecting the longest sequence as the main branch in the case of Bitcoin, uses a weight that not only takes into consideration the
Figure 2.7: Example of a fork in the blockchain. This situation arises when two miners solve the puzzle simultaneously and extend the blockchain starting from the same block. The block that is proposed next will break this tie and decide which one of the contending blocks gets confirmed and which will be orphaned.

In case of Bitcoin, the fork resolution algorithm will choose the green line with Block 5 at the end that is longest, while in case of Ethereum the chosen one will be the orange line with Block 4 in the bottom right. This difference is mainly due to the design decision that results in substantially longer time intervals between blocks for Bitcoin than for Ethereum. On average Bitcoin, blocks are mined every 10 minutes, while in the case of Ethereum they are mined at 10-15 seconds. For Ethereum this raises the probability of forks appearing, so the GHOST algorithm is used to try to counter the negative effects of this and prevent wastage as much as possible [23].
Figure 2.8: Depicts the difference between the fork resolution procedure of Bitcoin and Ethereum. While Bitcoin miners always extend the longest chain, Ethereum uses the GHOST algorithm to choose the sub-tree which has the most amount of work put into it.

The GHOST algorithm of Ethereum is also helping to prevent another issue, which is that malicious miners can disrupt honest miners by not announcing a newly forged block as soon as it’s discovered. Instead the dishonest miner will keep it a secret for a while, immediately start mining on the following block, which extends the secret block, meanwhile, other nodes are wasting their mining power on extending the block that is not the latest anymore (but they don’t know it yet). GHOST can help with this by accounting blocks proposed by miners in multiple branches, and not just focusing on the fastest growing and longest branch.

2.3.4 Previous PERCCOM thesis

It is worth noting that within the domain of existing literature there is a previous master’s thesis written by Manish Lamichhane in 2017 [25]. In his work, Manish proposed to combine Blockchain with Smart Waste Management and evaluate its feasibility for the facilitation of payments.
He implemented Smart Contracts in Ethereum to handle automatic micro-payments from citizens who deposit waste to the company who will eventually take care of collection and recycling. While he concludes the thesis by stating that blockchain can help with creating a payment infrastructure for Smart Waste Management, he acknowledges that long confirmation times of blockchain applications can cause problems that need further solutions.

2.3.5 A case for private blockchains

While public permissionless blockchains, such as Bitcoin and Ethereum has enjoyed increasing popularity and adoption in the cryptocurrency domain, it is not yet clear whether they will live up to the claim, that it will completely replace and revolutionize our current money systems. There are probably many reasons for this, and one of the main ones is that, as mentioned previously, due to the nature of the permissionless PoW puzzle, consensus cannot be fully solved (as demonstrated in [24]).

Because these blockchains trade deterministic guarantees for probabilistic ones, there is a possibility of violating the safety of the protocol. Some solutions, however, have proposed to restrict the model and allow only authenticated and authorized nodes to join the consensus process. This can make blockchains safer and more lively, but only at the expense of changing its openness.

These so-called Consortium blockchains allow a pre-selected set of nodes to participate in the consensus, usually owned by different organizations and companies, hence the name consortium. This restriction is most commonly implemented by PKI certificates and common authentication methods. Properties of such blockchains include:

- **Permissioned**: Only a set of authenticated participants can affect the consensus (WRITE permission to the blockchain). However, it is still possible for anyone from the public to have READ access to the blockchain.
- **Global view**: it becomes reasonable to assume that each participant is aware of a complete list of all other participants. This helps with preventing the Sybil attack, by preventing an attacker forming multiple identities and flooding the network with malicious transactions.
• **Fault tolerant**: Thanks to the permissioned nature, one can assume the prevention of fake identities and a genera KYC process, which help to put a bound on the number of faulty nodes at $f << n$, where $n$ is the total number of nodes and $f$ is the number of faulty or malicious nodes.

According to [23], if it’s true that $f < \frac{n}{3}$, then it becomes possible to solve the Byzantine Consensus problem in consortium blockchains without the need for Proof of Work puzzles. However, there are still some limitations, such as the fact that this usually requires a leader election process. This, however, against the whole notion of decentralized blockchain, and this leader could decide to act honest only until it gets elected for a round, then go rogue. Examples of such consortium blockchains include Ripple, R3 and Hyperledger Fabric.

### 2.3.6 Challenges ahead

Blockchain systems have been undoubtedly on the rise in recent years, thanks to the promise of revolutionizing money and society. There are however several issues that need to be tackled before wide-spread adoption can be expected. In this sub-section literature is reviewed with regards to the challenges of the adoption of blockchain technology for various purposes.

Authors in [26] categorize Blockchain into 3 distinct versions. Blockchain 1.0 brings about digital or cryptocurrencies, such as Bitcoin and many other similar projects. Blockchain 2.0 elevates the stakes and seeks to enable a Digital Economy and the main vehicles for this are smart contracts that interact with cryptocurrencies, the main example being Ethereum. Blockchain 3.0, according to the authors will usher in a new Digital Society with applications in art, health-care, identity, education, cities and governance.

Despite these promising aspects, the authors in [26] also point out that several challenges remain, such as privacy, scalability and interoperability.

• **Privacy**: refers to the idea that while accounts are identified by pseudonymous Public/Private keypairs, it still remains a possibility to apply graph analysis tools to reveal information about individuals behind transactions [27, 28, 29]. As a coun-
termmeasure, it’s recommended practice to use a brand-new keypair for every trans-
action, but this still does not solve the issue completely.

- **Scalability**: refers to the notion that was mentioned earlier in this section, that in
  order to increase the TPS (Transaction Per Second) performance of blockchains, the
  probabilistic PoW puzzle has to be traded in favor of a deterministic algorithm, that
  often requires the architecture to be consortium (private and permissioned) [29, 30].

- **Interoperability**: refers to the fact that today’s blockchains operate as some kind of
  walled garden which prevents easy cooperation [26]. This way it becomes impos-
  sible to transfer directly value from one blockchain to another, which is especially
  problematic in certain health-care scenarios where this is crucial [31].

When it comes to Blockchain and IoT integration, one of the biggest issues, according
to [32], is that the resource-constrained nature of IoT seems to be in a direction opposite
of what Blockchain requires. The ledger of public blockchains such as Bitcoin grows
continuously, as of the writing of this study (May-2019) the total size is around 210 Gi-
gabytes. This alone is prohibitive to many classes of IoT devices. Additionally, there is
the constrained in processing power and battery capacity, which prohibits the use of PoW
puzzles entirely. Bitcoin, as an example, has been estimated to consume on a global level
more electricity than smaller individual countries, such as Denmark.

Most projects that combine IoT and Blockchain envision a scenario where lightweight
IoT devices may participate in Blockchain networks by connecting to a more powerful
IoT Gateway [33], which acts on their behalf and interacts with the blockchain network.
This Gateway can be a full-node with mining capabilities if the devices are not battery
powered, or it can be a more lightweight one as well. This method of integration helps to
alleviate the problem that comes from the fact that IoT devices, due to their constrained
nature usually act in a Client-Server model, whereas Blockchain devices work in a Peer-
to-Peer model. This proposed IoT Gateway can potentially help solve this issue.

Another constraint that results in incompatibility between IoT and Blockchain is a
discrepancy in communication frequency. IoT nodes usually require sleep cycles to man-
age to conserve battery, whereas P2P blockchain networks often need frequent packet
exchanges to keep their state consistent and up to date [33]. This discrepancy can perhaps
also be alleviated via the use of IoT Gateways mediating between the two domains.
Authors in [34] point out the need for better privacy and security in an IoT context with blockchain application. Their blockchain solution in Ethereum handles the authorization of network access based on the ACE framework. In any IoT enabled blockchain scenario it may be necessary to protect in transmission between peers of the network. If an adversary captures this information they may be able to store it for later analysis and gain valuable information out of it.

Finally, another issue exists that affects not just Bitcoin but all other blockchain projects that use public key cryptography (PKI) based on the RSA algorithm. This threat comes from the potential that, in the not too distant future, Quantum Computers may be developed and we already know that algorithms exist (one such example is Shor’s algorithm), which is able to solve the mathematical problem that underpins the RSA public key cryptosystem. If this indeed happens, then an attacker in possession of a Quantum Computer can potentially unlock any Bitcoin wallets and spend coins at will. To protect against such a scenario, there are already efforts ongoing that aim to develop post-quantum (meaning safe in a world where Quantum Computers exist) algorithms that can withstand such attacks, however, it is an entirely different question how to integrate such algorithms into Bitcoin and other such projects.
2.4 Quantum Computing

In this subsection a general overview is presented about Quantum Computing based on Quantum Mechanics, followed by a brief introduction of traditional Key Distribution / Agreement protocols. Finally Quantum Key Distribution is introduced and differences between QKD and Traditional Key Distribution are highlighted.

2.4.1 Overview

Quantum Computing emerged during the 1980s and is a sub-field of Quantum Information Science. It borrows ideas from Quantum Mechanics to establish a model for a computer different from traditional computers. The main difference being that traditional computers speak the binary language of \(0, 1\).

As the reader may know, classical computers store the binary information in various ways, such as electrical currents on a memory chip, or capacitive charge on an SSD device or magnetization used by older HDDs. Quantum computers, on the other hand, use quantum bits or *qubits* which are a bit more complex than classical bits, in that the state of a qubit is usually represented as a *vector* in two-dimensional vector space, also known as the *state space*.

Besides the differences in representing the base unit of information, quantum computers also differ from classical ones in the way they store this information. While classical computers store information usually by using electrons to create a difference in charge, quantum computers can use electrons, photons or even different atoms. Different methods exist for this purpose, such as using the spin of an electron or the polarization of a photon. These details are highly specific and various ongoing research efforts are trying to create the best solutions for various scenarios.

In order to build a quantum computer, physicists and computer scientists are trying to exploit the lessons learned from Quantum Mechanics, which deals with the fundamental laws of physics when it comes to the smallest scales of energy in atoms and subatomic particles.
2.4.2 Traditional Key Distribution

The most common way of providing communication security nowadays is encryption, which can hide the real message from adversaries even if they manage to record it and store it for later use. There are two basic types of encryption, one is called symmetric the other asymmetric. The difference lies in the use of encryption keys. Symmetric encryption algorithms use the same key to encrypt and decrypt the message to be transferred, while asymmetric algorithms use a key-pair, such that if the message is encrypted with the encryption key, it can only be decrypted with the decryption key.

Once encryption is enabled, the problem of secure communication reduces to key management. In case of asymmetric encryption, the situation is somewhat simpler, as the encryption key is usually kept secret but the decryption key needs to be revealed to the public. If someone wants to send a secure message to an entity using asymmetric
cryptography they need to look up their public key and encrypt the message with that key. This ensures that only the owner of the key pair can decrypt it using the secret key that must never be revealed.

In contrast, the encryption key of symmetric algorithms must be kept secret and rotated frequently. As such, it becomes a question of how to agree on a shared key between two parties who cannot trust the communication channel between them to send messages safely. One obvious solution could be to come up with some shared secret by one of the participants and send it to the other by mail in the post. This solution has some obvious drawbacks such as long delivery times and the possibility of interception.

A clever solution to this problem was found by Whitfield Diffie and Martin Hellmann in 1976, who demonstrated that it’s possible to use the same underlying math theories that establish public-key cryptography. In principle, their algorithms allow the establishment of a shared secret over a public channel that may be observed by a MITM adversary.

The algorithm is comprised of 5 different steps and all messages that exchange can be seen by and possible adversary on the public channel between them. In Step 1 Alice and Bob agree on a pair of public values (P is a prime and G is a generator number). In Steps 2 and 3 the two parties Alice and Bob pick a random value they keep secret and calculate an equation that involves modular exponentiation. In step 4 they exchange this calculated value. In Step 5 they do one further calculation using this recently exchanged value and as a result, they will both end up with the same secret value.

The strength of this algorithm lies in the difficulty of solving the discrete logarithm problem, given the public and secret parameters that have been calculated and exchanged. There are however two main issues with this algorithm.

1. **Does not provide authentication**
   Since the protocol does not provide any method to authenticate, it becomes feasible for an adversary to impersonate both parties and perform a MITM attack. This implies that the attacker is able to intercept and replay messages between two participants, such as Alice and Bob from Figure 2.9.

2. **Not safe against quantum computers**
   Peter Shor in 1995 published his algorithm [35] which was proven to be able to solve
the discrete logarithm math problem, which underpins many cryptographic algorithms, such as RSA, and the Diffie Hellman key exchange protocol. This means that such crypto functions become insecure as soon as quantum computers are developed that are capable of executing this algorithm.

2.4.3 Quantum Key Distribution

Quantum Key Distribution (or QKD in short) arose in the 1980s, with the BB84 algorithm from 1984 [36] as one of the earliest demonstrations of the use of quantum mechanics in the field. This field is often referred to as Quantum Cryptography. To be able to explain how QKD systems work, it is perhaps easier to think of an analogy that uses fiber optic channels as a medium of communications.

Essentially, in classical fiber channel communications, the information is often transmitted by the presence or lack of photons flowing from one end to another. The simplest method is using the presence of photons as bit 1 and lack of photons as bit 0. This way a very simple (and not very efficient) way of communication emerges, using plain fiber optic channels.

In contrast, QKD systems exploit the fact that thanks to Quantum Mechanics and recent advances in our ability to manipulate and measure even the slightest modification of physical particles, a new way of communicating information emerges. QKD systems, in essence, exploit the fact that using the same fiber optic channel, an entity on one end can send single polarized photons and the other entity on the opposite end of the channel can measure and understand the polarization of the sent photons.

Figure 2.10 depicts two different scenarios. The top half shows a traditional fiber optic channel that sends photons in a non-quantum way. In this scenario it becomes possible for an attacker to impersonate Bob for Alice and Alice for Bob, so that when Alice and Bob execute the classical DH key exchange algorithms they both think they communicate with the other person, but in reality the attacker is impersonating each entity towards the other and defeats the whole purpose of the protocol. In the bottom half of Figure 2.10 a quantum channel is depicted which makes it impossible for a MITM adversary to disturb the channel and avoid being detected.
Algorithms such as the BB84, which is one of the earliest attempts at creating QKD systems transmit information encoded in the polarization of photons. The advantage of QKD includes the fact that relying on the laws of quantum physics (specifically the no-cloning theorem), allows one to detect with high probability the presence of malicious third-party who is trying to listen in on the communication. In fact, when a malicious actor tries to copy the photons he will inevitably disturb the quantum state and let the two participants know by increasing the error rate of the fiber optic channel they share for communication.

By clever use of the underlying laws of quantum mechanics, the BB84 was the first protocol to show that it’s possible to use such algorithms for secure distribution of encryption keys. When it becomes possible to detect malicious man-in-the-middle (MITM) attacks, one can replace the traditional key distribution mechanisms, such as the Diffie Hellman Key Exchange protocol that is vulnerable to quantum computers, with new protocols that offer much better protection against adversaries.
Researchers from ITMO University have demonstrated in [37] the possibility of using the BB84 protocol and Sub-Carrier Wave method of Quantum Key Distribution in a metropolitan test network to securely distribute quantum keys. Their system is claimed to be robust against environmental fluctuations and offers optical synchronization with a 1,06 Mbit/s key rate. The system was deployed in two different test-bed environments with different photon detector types and reached a maximum distance of 265 km.

As pointed out by [38], Quantum Cryptography and QKD are often used interchangeable however one thing should be clarified. QKD systems are not used to encrypt/decrypt or transfer encrypted information. The only use for QKD is to securely distribute securely key material, which then can be incorporated into any symmetric crypto algorithms. Below a few pros and cons of QKD versus traditional key exchange protocols are discussed:

Table 2.1: Pros and Cons of QKD.

<table>
<thead>
<tr>
<th>PROS</th>
<th>CONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security provided by laws of physics</td>
<td>Needs dedicated hardware</td>
</tr>
<tr>
<td>Secure against quantum computers</td>
<td>Expensive and still in development</td>
</tr>
<tr>
<td>Future proof</td>
<td>Distance limitations</td>
</tr>
</tbody>
</table>

2.4.4 Quantum efforts and Blockchain

According to [39] the field of Quantum Information Science (or QIS) is poised to become the next frontier where several players will try to make a breakthrough in order to attain so-called quantum supremacy by being the first to develop a universal Quantum Computer. To this end, there are several ongoing efforts in the world, including private companies and state-owned research centers trying to solve the biggest problems of the QIS field.

Most notably in China a $11 billion project was announced to build a huge national quantum laboratory in Hefei, which is dedicated to making the country a leader of this race. Russia established a national research center back in 2010, called RQC (Russian Quantum Center) which aims to bring researchers together within the country and also internationally to cooperate and invent new technologies that may help advance the state of QIS. European and US researchers and companies are also very active in the field
and several products are available on the market already when it comes to Quantum Key Distribution equipment.

At the intersection of blockchain and the quantum world, there are several interesting projects that exist and some of them will be reviewed in this short section. First, the author of [40] in 2016 proposed a scheme in which the no-cloning theorem of quantum mechanics is exploited to create their so-called Quantum-Bitcoin. This solution is different from Bitcoin, does not use the costly Proof of Work mining mechanisms and claims to allow dishonest peers to mint Quantum Money without risking its integrity. They also claim transactions can be verified immediately locally, which is a major difference to Bitcoin.

Besides the known algorithms that run much faster on quantum computers, and quantum cryptography, the author of the said paper argues that quantum mechanics can also give us new and improved money systems, by exploiting the same no-cloning theorem that underpins quantum key distribution systems.

The authors of [41] propose a scheme in which they utilize quantum computing to empower Bitcoin by speeding up the mining procedure using a modified version of Grover’s algorithm [42], which was invented by Lov K. Grover in 1996. They propose to use this fast search algorithm to help speed up the PoW puzzle, but it’s not clear how effective the proposed solution is and what effects it has on the incentive mechanisms of Bitcoin.

Authors of [43] present an experimental setup of a quantum-secured channel between two branch offices of a bank in Moscow, Russia. Their solution uses an already deployed and lossy fiber optic channels to generate key material which is then adopted by VPN equipment that securely transmits information between the two branches.

The same infrastructure is used in another work [44], in which the authors combine blockchain and QKD to provide an information-theoretically secure version of a cryptocurrency. In their setup, a pairwise quantum channel is established between each pair of nodes (4 in total) they use the obtained keys to secure the blockchain communication layer. Their solution also changes from the traditional PoW mining consensus and replaces it with an information-theoretically secure broadcast protocol which helps the nodes reach an agreement about the state of the blockchain.
3 SYSTEM DESIGN AND IMPLEMENTATION

In this section details of the implemented scenarios are discussed. First we motivate the choice of methodology that was followed, then an overview of the envisioned Smart Waste Management Architecture is given, then the Block[Chain] Alchemy Lab framework and toolkit is introduced. Then Mininet is briefly introduced, after which the two scenarios, based on the BC variants, are described in detail.

3.1 Methodology

Figure 3.1: Overview of Design Science Methodology, emphasizing the six major steps and 4 possible entry points for the whole process. Source: [45].

To aid with this study, the Design Science Research Methodology was chosen, because it is a generally suitable methodology for Information Systems research. As pointed out by the authors of this methodology in [45], natural and social sciences try to understand reality, design science, on the other hand, attempts to develop things to serve a human purpose. Essentially, it can be thought of as the seeking of a solution for a problem formulated in desires.

At its core, DSRM creates and evaluates artifacts that are intended to solve the identified problems. To achieve this it offers three objectives: (a) be consistent with prior literature; (b) provide a nominal process model for research; (c) provide a mental model for evaluating and presenting the research; as well as a few guiding principles, practices and procedures.
The six main steps of the DSRM which can also be seen in Figure 3.1. When it comes to this thesis project, the methodology was following by answering these points individually during the various phases.

Table 3.1: DSRM process in the context of this study.

<table>
<thead>
<tr>
<th>Step</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The problem definition originates in the original topic proposal, based on the initial literature review regarding Internet of Things and Waste Management in a Smart-City context.</td>
</tr>
<tr>
<td>2.</td>
<td>The objective of the solution is establishing a model for providing secure communications in the context of an IoT-enabled Smart Waste Management scenario, and exploring how Blockchain and Quantum Key Distribution may help with this.</td>
</tr>
<tr>
<td>3.</td>
<td>The basic architecture of the secure Smart Waste Management model was designed and subsequently implemented in Mininet scenarios.</td>
</tr>
<tr>
<td>4.</td>
<td>Demonstration achieved through the implemented Mininet scenarios running the Blockchain and QKD enabled Smart Waste Management.</td>
</tr>
<tr>
<td>5.</td>
<td>Evaluation via a Python script executed within the Mininet scenarios that collects metrics about the Blockchain application, as well as energy measurements at the VM level via the ”powerstat” tool.</td>
</tr>
<tr>
<td>6.</td>
<td>Communication of the findings is carried out by this thesis report and a subsequent research paper that is to be published in a conference or journal.</td>
</tr>
</tbody>
</table>

3.2 Smart Waste Management Architecture

This section contains general information about the architectural elements of the proposed solution, which was eventually implemented in Mininet scenarios. Figure 3.2 shows different elements of the architecture. This described architecture is envisioned to serve as the underlying infrastructure for a blockchain assisted IoT-enabled Smart Waste Management system of a Smart City.

The different components of this system are described in greater detail:

- **Smart Garbage Bins** are considered to be the IoT layer in the architecture and they are envisioned to be battery-operated, solar-powered Raspberry Pi devices, equipped with LoRa antennas that they use to communicate with regional LoRaWAN gateways.
Figure 3.2: High level overview of the proposed architecture. Different components, such as garbage trucks, SGBs and Blockchain Nodes connected by quantum links provide a secure infrastructure that can facilitate payments for services of a waste management system in a Smart City.

Furthermore, these SGBs are equipped with various sensors to detect weight, and fill level, which are then collected and communicated to the central Waste Management Operator for processing.

- The garbage trucks are equipped with communication and processing units which are responsible for relaying location information to the Waste Management Operator, as well as local storage units for storing keys that are generated by the Quantum Key Distribution unit and are meant to be delivered to individual SGBs to be used for symmetric encryption to provide security.

- The Waste Management Hubs are designed to act as geographically distributed computation and communication units, for example, deployed in each district of a big city, which acts as the target for SGBs within its area. These hubs are also envisioned to be running a blockchain variant developed for this study.

- The QKD Backbone consists of communication infrastructure and hardware components deployed in a Smart City and used for various purposes, including to generate and distribute quantum keys to the regional Waste Management Hubs and to the trucks which eventually forward them to SGBs. The keys can be used for two different use-
cases: (a) Blockchain nodes implementing our Proof of Infrastructure consensus will take advantage of them for the purpose of consensus, while (b) SGB clients will use it to encrypt their transactions before sending them over the network.

In this scenario, citizens deposit waste into SGBs either in a sorted or un-sorted fashion. The deployed sensors of the smart garbage bin measure the fill level and communicate this information to the central WMO. The WMO then schedules an optimal path for a garbage truck to empty all the full SGBs.

When the Truck arrives the SGB initiates a transaction to pay for the service of waste collection. This transaction is submitted to the closest SWM Hub via the LoRa communication. The SWM Hub which acts as a LoRaWAN gateway also runs a local blockchain node. These SWM Hubs are deployed in the city in a distributed manner and form a peer-to-peer blockchain network in order to process transactions.

![High-level flow chart for waste management scenario.](image)

Once the transaction is confirmed the information is relayed back to the SGB and the Truck and service of waste collection are carried out.
A sequence diagram version his scenario can be seen in more detail on Figure 3.3 below. This figure is a simplified version of interaction between an SGB, the Waste Management Operator and the Truck waste collection service.

### 3.3 Block[Chain] Alchemy Lab

This section provides a brief overview of the Block[Chain] Alchemy Laboratory [46], which is created by a research team at ITMO University, lead by Prof. Oleg Sadov in Saint Petersburg, Russia. It is created in order to serve as a toolkit for educational and research activities related to blockchain. It consists of frameworks and programming tools for modeling different blockchain variants, with the aim to be flexible and enable the changing of different components, such as consensus, storage, transport, etc.

![Class diagram for BAL blockchain project](image)

*Figure 3.4: Class diagram for BAL blockchain project. Different blockchain consensus variants are implemented in different classes.*
In order to build the Mininet scenarios, two different variants of blockchain have been implemented by extending the Python classes provided by BAL. The basic functionality of blockchains has been implemented in a class called *BaseBlockChain* while the Proof of Work and the Proof of Infrastructure variants are implemented in separate classes extending this base class (*POWBlockchain* and *QuantBlockchain* respectively).

Some common functionalities are abstracted into a base python class, for example, saving to and initializing from the persistent database in the OS file system; while specific PoW and PoI functionalities are implemented in their own classes. One difference between the two variants is the type of neighbor node information that gets stored. For PoW it’s just a simple list of strings which store *IP:*PORT pairs, while for PoI there is additional information about quantum channels existing between every possible pair of nodes.

### 3.3.1 PoWBlockchain Class

This class implements a basic version of a Proof of Work blockchain by extending the functionalities of the *BaseBlockChain* class of BAL. The difficulty for the Proof of Work puzzle can be adjusted by passing a parameter upon the creation of the class, or by changing the class variable *difficulty* during run-time. This variable represents the number of leading zeros for a hash to be accepted as the correct solution.

```python
""" Proof of Work function ""

def proof_of(self, last_block):
    last_proof = last_block['proof']
    last_hash = self.hash(last_block)
    proof = 0
    while self.valid_block(last_proof, proof, last_hash) is False:
        proof += 1
    return proof

Listing 1: *Proof of Work hash puzzle implemented in Python (from BAL project).* The function takes the last confirmed block which is used as input to the hash puzzle to create the next block.

44
In the PoWBlockChain class, this is implemented in the valid_block function that gets called when the BaseBlockChain class’s mine method invokes the proof_of hash puzzle method to find the magic number that results in the correct hash.

3.3.2 QuantBlockChain class

In case of the Proof of Infrastructure (or Quantum) variant, this hash puzzle is removed and replaced with a function that simply calculates a new hash based on the hash of the whole previous block and the proof of that block.

```python
""" Proof of Infrastructure function """
def proof_of(self, last_block):
    last_proof = last_block['proof']
    last_hash = self.hash(last_block)
    raw_proof = '{proof}'.format(last_proof, last_hash).encode()
    proof = hashlib.sha256(raw_proof).hexdigest()
    return proof
```

Listing 2: Proof of Infrastructure consensus implementation in Python (from BAL project).

In this way, the PROOF in each block’s header contains a hash value that gets created by hashing the previous block’s hash and the PROOF value of that block.

For example, when two Proof of Infrastructure nodes cooperate in consensus, they will exchange their blockchain data and seal it by hashing the chain of proofs with a key that’s generated by the QKD channel between those two nodes who are running the consensus protocol.

3.3.3 QTopology class

Another important toolkit in the BAL framework is the QTopo class, provides a way to build resilient and secure infrastructure by utilizing parallel links and incorporating a mechanism to perform fail-over between a normal and a quantum protected channel in
case disturbances are detected in the network, which hints at a possible MITM attacker.

Figure 3.5: BAL QTopo.py example.

Figure 3.5 provides a brief overview of the idea and shows an example of executing the test scenario included in the BAL repository [46]. In this implemented mini scenario there are two hosts connected to two switches, which have two parallel links. One link is "normal" while the other implements transparent quantum encryption via keys provided by quantum key distribution emulators. Controlling the two switches is an SDN controller implemented in Ryu, which listens to incoming requests on the endpoint /channel/x/x and based on the request parameters it can instruct the switches to switch between using one or the other link.
3.4 Mininet scenario modeling

Mininet is an open source network emulator tool written in Python to enable rapid prototyping and experimentation with networks that may resemble a real-world infrastructure such as virtual hosts, networking devices, controllers and links. Spinning up and down the emulations takes seconds and the hosts are running a real Linux kernel and network devices support the OpenFlow SDN network paradigm [47]. Originally meant for helping research into Software Defined Networking, it is now used in a wide range of projects.

It is possible thanks to a feature of Linux kernel called network namespaces, which allow the isolated creation of individual processes with separate network interfaces, routing and ARP tables. Behind the scenes, the magic of Mininet is achieved by calling Linux commands which take care of creation and assignment of resources to namespaces accordingly.

Figure 3.6: Example scenario in Mininet, which can be created and executed by a simple command line in Linux in a few seconds.

Figure 3.6 shows a tree-like emulated topology of 2 layers of switches and 1 layer of hosts, and performing a ping test between each possible pair of hosts takes only 10 seconds. Output of the above emulation can be seen in Listing [5] in in Appendix 2.
3.4.1 Blockchain Node implementation

In order to facilitate the starting and stopping of different kinds of nodes inside the Mininet simulations, it was necessary to create a collection of classes that all extend from `mininet.node.CPULimitedHost` class of the Mininet Python API. The implementation details of these classes can be found under the folder of the projects GitLab repository: `secure-swm > bal_sw > bal_sw > swmnodes.py`

In the case of the Proof of Work Blockchain Node in Mininet simulations, first, there is a base class called `BCNode`, which is then extended by `POWNode`. The base class takes care of the implementation of starting and stopping the application on the instantiated node, while the `POWNode` class takes care of supplying the necessary parameters and arguments for the execution at simulation time.

```python
class POWNode(BCNode):
    def __init__(self, name, inNamespace=True,
                 server='bcn.py',
                 sargs='-p {port} -d {sdir}/pow-{IP}.db -v pow:5 -n {node_id}',  # ADDED -n (node_ID)
                 sdir='/var/tmp',
                 client='curl',
                 cargs='''-s -X GET -H 'Content-Type: application/json'
                 -d '{data}' http://{IP}:{port}/{command}''',
                 cdir=None,
                 ip='127.0.0.1', port='5000', **params):
        dir_path = os.path.dirname(os.path.realpath(__file__))
        srv_name = server
        BCNode.__init__(self, name, inNamespace=inNamespace,
                        server=dir_path+'/' + srv_name, sargs=sargs, sdir=sdir,
                        client=client, cargs=cargs, cdir=cdir,
                        ip=ip, port=port, **params)
```

Listing 3: Mininet blockchain node implementation for the Proof of Work `BCNode` simulation scenario, utilizing the `bcn.py` python script.

As it can be seen on Listing 3, the implementation details for `POWNode` are quite simple. They consist of defining a few default values for class variables that mostly store
strings. These stored string variables are used to construct a CLI call that is executed by Mininet upon startup of the scenario.

The `bcn.py` file, which is referenced in this Mininet Node Class, can be found in the project’s GitLab repository. It is a simple executable Python script that utilizes the POWBlockChain and QuantumBlockChain classes discussed earlier in Sections 3.3.1 and 3.3.2. To provide the blockchain node functionality it uses the Flask micro-framework, which helps to easily implement a set of REST API endpoints for various functions such as registering new peers, accepting transactions and doing consensus with peers.

The source code for these classes and the simulations can be found in the project’s public GitLab repository [48].

### 3.4.2 Proof of Work Scenario

This subsection describes the topology of the Mininet scenario in which a Proof of Work blockchain network was implemented and evaluated. Although Figure 3.3 gives a good overview of the envisioned waste management process, the implemented scenarios make some simplifications by hiding details of communication flows, such as communication between the Waste Operator and the trucks, and signaling of successful or failed transactions to the originator and the trucks. The below figure shows the general architecture of this scenario. Figure 3.7 contains the following types of nodes, which extend the functionalities of the `mininet.node` class from the Mininet Python library:

- **BCNode-[1-4]**: these nodes are running the PoW variant of the Blockchain Node application offering a REST-ful API that accepts transactions from SGBs in JSON format.
- **SGB-[1-12]**: these nodes are running a small python application script that generates and submits blockchain transactions to a target BCNode at a randomized time-interval (0-2s, 1-4s or 2-4s).
- **Switch-[1-5]**: these are running the OpenVSwitch acting as Layer2 learning switches controlled by an external Ryu SDN controller which is programmed to enable Layer2 connectivity (no routing, only packet switching within the same subnet).
Once the simulation is started the BCNodes boot up and start accepting incoming TX-es via their REST API. The SGBs will start generating these TX-es according to the random delay that was specified previously. During the simulations, this delay was varied between 0-2 seconds, 1-3 seconds and 2-4 seconds. This means that the SGB will send a new TX with a random delay between those two values.

While in the real world it does not make sense to have such frequent transactions originating from the same SGBs over and over, since it usually takes a few days for them to fill up, at which point the transaction would be initiated once. However, for the purposes of this simulation, it was decided not to instantiate thousands of SGBs, but rather use a smaller number of them and increase the intensity with which they generate the transactions towards BCNodes.

Furthermore, as it can be seen on Figure 3.7, there are 4 BCnodes and 4 groups of Garbage Bins. Each group of SGBs have been programmed to direct their transactions always to the same BCNode. This is meant to mimic how it would function in real life, when all the garbage bins of an area or district of a city would connect to the nearest
available SWM Hub in the same district. When a BCNode receives a transaction it will re-broadcast it to the other neighbors it is aware of, while also making sure not to forward the same TX over and over.

For BCNodes the main task is to listen for incoming transactions and accumulate them in a pool of pending TX-es, until the next block is created. In this case, before the next block can be appended to the blockchain, the PoW puzzle has to be solved which results in an integer that when hashed with the rest of the block gives a correct solution to the difficulty that’s specified at startup time.

This PoW puzzle computation is the main reason why PoW style blockchains are considered to be inefficient and hard to scale. If the system needs to handle a large number of transactions, but each block can only contain a limited amount of them, then in order to scale the system the computational resources need to be increased dramatically to speed up the transaction processing. However, in Bitcoin-style PoW schemes, there is a dynamic adjustment of the mining difficulty, which means if new resources are added to increase the system’s capacity the difficulty will eventually increase in order to ensure an average constant block generation time.

### 3.4.3 Proof of Infrastructure Scenario

Below is an overview of the Proof of Infrastructure scenario which uses the QKD-enabled Blockchain variant.

This scenario is largely the same as the previous Proof of Work scenario from Section 3.4.2, when it comes to the network topology and the functionality of the SGB Nodes. What is the main difference is the introduction of the QKD emulator nodes, as well as the interconnection of BCNodes that are now each equipped with their own Keyworker unit. These two nodes work together to achieve their purpose and below is a description of each:

- **Keyworker Unit:** this unit is written in Go and runs on every BCNode instance. It takes care of storing the latest key which was generated by the Quantum Key Distribution emulator, which is used for the consensus algorithm of PoI blockchain variant.
- **QKD Emulator Unit**: this unit is written in Go and takes care of the periodic installation of newly generated keys in the keyworkers that are deployed at each end of every quantum channel.

![Diagram](image-url)

Figure 3.8: *Proof of Infrastructure scenario using the QuantumBlockChain Python class to implement the quantum-key based consensus. The QKD emulator nodes make sure that quantum keys get distributed and stored in keyworker instances of the same channel.*

The BCNodes in this scenario implement mostly the same functionalities as the PoW variant BCNodes, with the main difference being in the way they reach consensus.

As mentioned previously in Section 3.3.2, this version of the blockchain removes the PoW puzzle which consumes a lot of resources and replaces it with a network of trust that is supported by the quantum channels which exist between every pair of BCNodes in the network.

The pseudo-algorithm in Listing 4 contains the logic that gets executed when the consensus is triggered on one of the BCNode instances of the simulation. When the algorithm gets triggered it is executed on all other BCNodes as if it was a chain reaction. This is to ensure that all blockchain nodes keep the same up to date blockchain in their memory.
""" Pseudo-Algorithm of: Proof of Infrastructure Consensus"""

```python
foreach (neighbor in neighborList):
    get neighbor blockchain
    get key from quantum channel of neighbor
    if (neighbor blockchain is valid with key):
        if (neighbor blockchain is longer than ours):
            replace our blockchain with neighbor bc
        return true
    else:
        return false
```

Listing 4: Proof of Work hash puzzle implementation in Python.

When the BCNode requests the blockchain from another peer, it will receive a response in JSON format, with an additional hash fingerprint, that can only be calculated if the peer was in possession of the same shared secret key that was installed by the QKD Emulator node to its keyworker unit. Upon receiving the JSON blockchain data, the BCNode will traverse the chain and validate it using its own shared key. If this step succeeds then the neighbor’s blockchain is considered valid and its length is checked. If the BCNode’s own blockchain is shorter, then it will adopt the longer one from the neighbor.
4 RESULTS AND DISCUSSION

In this section metrics and benchmarks are collected from the scenarios implemented in Mininet. Benchmarks were taken with various different parameters to be able to understand how they affect the performance of the implemented scenario. When collecting statistical data, the main focus was energy consumption and various blockchain transaction metrics, such as transaction confirmation delay and transaction throughput.

4.1 Simulation environment specifications

In order to gather the results presented in this section, my own laptop was used, which acted as a development and testing machine throughout the whole project.

- Model: Lenovo ThinkPad T460
- CPU: Intel Core i5-6300U CPU @ 2.40GHz * 4
- RAM: 16 GB
- Storage: Intel SSD 250GB
- OS: Ubuntu 18.04.2 LTS

Within this Linux machine, there was another virtual machine built up especially for the purpose of development and measurements of results of the scenarios. Parameters of this Virtualbox VM: 2 CPU cores, 8 GB of RAM, 20 GB of storage and the Ubuntu 18.04 operating system.

4.2 Energy consumption metrics

This section describes the various metrics that were gathered from the Proof of Work scenario implementation in Mininet. The tool used for this purpose is called powerstat, which enables continuous measurement of power consumption of a running Linux machine. The measurements that were taken were executed within the virtual Ubuntu host on my laptop.
4.2.1 Proof of Work energy profile

When it comes to the energy profile of Proof of Work blockchain applications, it is most well-known that the PoW hash puzzle consumes a considerable amount of resources, which is by design.

To characterize the energy consumption of our implementation of Proof of Work, two different parameters of the Mininet scenario have been changed between measurements to see how it affects power consumption. These two parameters are the PoW difficulty for the mining hash puzzle and the intensity of incoming transactions from SGBs.

The PoW difficulty corresponds to the necessary amount of leading zeros that are required for the hash to be accepted. During the simulations it was varied between (3-4-5). The incoming transaction intensity was defined on the SGBs as a random delay between (0s-2s, 1s-3s, 2s-4s).

Figure 4.1 shows how energy consumption varies when the difficulty and transaction intensity parameters are changed in the simulation. Each group of colors represents the
same difficulty (green: 3, yellow: 4, brown: 5, red: 6), and the different values of the same color represent different incoming transaction intensities.

As the figure above clearly shows, running the simulation at Difficulties 3 and 4 does not cause a considerable spike in the energy profile of the scenario. When the difficulty is increased to 5 and 6 however, the consumption jumps to around 32 Watts which is close to 100%. This is expected as the PoW hash puzzle becomes progressively difficult to solve as more and more zeros are required to be at the leading character spaces. The figure also illustrates well the effects of incoming TX intensity on energy consumption. When the intensity is highest (0-2 seconds) the consumption is also highest, but when the intensity decreases (2-4 seconds) the consumption also shows a general decline within each color group.

### 4.2.2 Proof of Infrastructure energy profile

When it comes to the energy profile of the PoI blockchain variant, the measurements are simpler, as there is no POW-style difficulty that can be tuned. Therefore only the incoming TX intensity was changed between the different measurements.

![PoI energy profile](image)

**Figure 4.2: The effects of incoming transaction intensity on Proof of Infrastructure energy consumption.**
Figure 4.2 shows how the energy profile varies for different TX generation intensities. When the intensity was lowest (2-4 seconds apart) the power consumption was around 14 watts; around the same for the next intensity (1-3 seconds). There is a slight increase in power consumption for the 0-2 seconds intensity measurements (average around: 15, with some noticeable spikes) and it clearly shows that the highest intensity results in the highest consumption of around 22 watts on average.

4.2.3 PoW vs PoI comparison

When it comes to comparing the Proof of Work and Proof of Infrastructure scenarios in terms of energy consumed it is rather straightforward. The PoW includes the hash puzzle which at difficulty 5 already consumes almost 100% of the available CPU power, which means it will drive up the power consumption as well. But when it comes to the PoI variant, it is clear that this version is much more efficient and consumes less energy.

Figure 4.3 shows the measured values for PoW (difficulty 5 and intensity 0-2) and PoI (intensity 0-2). Also included in the figure are the minimum and maximum possible values to be able to relate to the overall capabilities of the system on which the measurements were executed. The figure clearly shows that PoI tends to draw closer to the idle consumption level, while the PoW variant is near the maximum limit of energy consumption.

4.3 Blockchain application metrics

This section provides details on different metrics that were measured and collected targeting the functioning of the blockchain application, including transaction throughput and transaction confirmation delay.

- Transaction throughput (or TX throughput) refers to the amount of transactions confirmed within a given amount of time, and it is expressed in \([\text{TX/sec}]\).

- Transaction delay (or TX delay) refers to the time elapsed between the creation of a transaction and its appearance in the blockchain, and it is expressed in \([\text{s}]\).
Figure 4.3: Comparing the energy consumption levels of the two different scenarios with the same fixed incoming transaction intensity. The orange and pink lines represent minimum and maximum power consumption, while the others in between are measurements taken during the execution of Mininet scenarios.

4.3.1 Proof of Work blockchain metrics

Figure 4.4 shows the two different metrics of the PoW blockchain variant. On this figure it can be clearly observed that having a lower PoW puzzle difficulty coupled with high TX intensity (0s-2s) means that the throughput is highest (around 12 TX/sec) and the delay is lowest (less than 1s) of all other measurement scenarios.

On the other hand, when the difficulty increases to 5 (represented as the 3 rightmost column pairs), it becomes obvious from the chart that TX delay spikes from around 1s to almost 10s, while the TX throughput drops from about 12 TX/s to around 7 TX/s. This can be easily attributed to the PoW puzzle which requires a considerable amount of computational power to solve at the difficulty of 5.
4.3.2 Proof of Infrastructure blockchain metrics

When it comes to the Proof of Infrastructure variant the measurements are simpler as there is no hash puzzle difficulty to play with. The measurements differ only in the intensity of incoming transactions generated by SGBs.

Figure 4.5 displays the correlation between TX intensity and blockchain throughput and confirmation times. When the intensity is highest (0-1 seconds) the throughput is also the highest (around 22 TX/seconds) with minimal confirmation delay. When the intensity is the lowest (2-4 seconds) then the throughput decreases (around 4.5 TX/seconds) and the confirmation delay increases as well.

This observed behavior easily explained by the lack of PoW-style hash puzzle which in this case does not require and computational power to be expended. Thus when the intensity goes up the transactions spend less time waiting and are confirmed more quickly and so the delay decreases and the throughput increases.
4.3.3 PoW vs PoI comparison

Last but not least the blockchain metrics for PoW and PoI variants are compared in this subsection. To simplify the explanation, only the high (0-2s) and low (2-4s) intensity scenarios are compared here.

![PoI Blockchain Metrics](image)

**Figure 4.5:** *Delay versus throughput for PoI Transactions. Column labels represent TX intensity. Example: (0-2) means TX intensity of 0-2 seconds.*

![PoI Blockchain Metrics](image)

**Figure 4.6:** *Blockchain metrics comparison between PoI and PoW variants for two different intensities (a: 0-2seconds, b: 2-4seconds).*

As it can be seen below on the two charts of Figure 4.6 there are some similarities and some differences between Proof of Infrastructure and different variants of Proof of Work.
On chart (a) of Figure 4.6 the metrics can be seen for PoI, PoW(3,4,5) with TX intensity of 0-2 seconds, while on chart (b) of the same figure the same metrics can be seen for TX intensity of 2-4 seconds.

The data in the above charts clearly shows that there are similarities between PoI and PoW(3, 4) variants when it comes to TX Delay and Throughput, however when we also compare the energy consumption of the same variants from Figure 4.3 we can conclude that the Proof of Infrastructure variant is most energy efficient and has the best blockchain performance.
4.4 Sustainability analysis

As this project was carried out within the PERCCOM program, it is important to analyze the model and the obtained results from a sustainability perspective. In order to do this we adopt the sustainability pentagon from [49] to offer a coherent and comprehensive overview of this issue. This model and its use is further explained in [50].

![Sustainability analysis via the Pentagon method.](image)

In this pentagon sustainability model, there are 5 dimensions (Individual, Environmental, Social, Technological and Economic) and each can have 3 different levels of impact: Immediate, Enabling and Structural. When it comes to this thesis project the below observations have been made regarding this method of analysis:

- When it comes to the **IMMEDIATE** effects of this proposed solution, at the individual level there is most certainly a learning period for citizens who need to adapt to the new
method of waste management that integrates technology such as IoT and Blockchain. From an economic perspective, this can boost profits by requiring an upfront investment, which then over the years of operation pays off by reduced CAPEX. This also helps with advancing technological adoption. From an environmental point of view, the most important thing is that waste management efforts improve their efficiency.

- Within the **ENABLING** effects the new kind of business models available thanks to the integration of IoT and BC can bring new economic and social benefits. On the other hand, it can also cause loss of certain types of jobs due to increased automation of the waste management process.

- Possible **STRUCTURAL** effects are those that are observable at the macro level and manifest in persistent changes in behaviors and processes around waste management. While these are hard to predict, it is reasonable to expect changes in the behavior of people as it relates to their ways of life and generation of waste. A citywide campaign in parallel with the deployment of such an upgrade to the waste management system can go a long way to change people’s waste handling practices.
In this thesis project, the intersection between the Internet of Things, Quantum Key Distribution and Blockchain technology has been studied in the context of a Waste Management scenario of a Smart City. Waste management becomes increasingly challenging as cities grow larger and larger, so there is great interest in coming up with technological solutions that can make cities more livable and sustainable from this perspective.

The IoT paradigm is poised to be the next big thing for enabling novel solutions to problems old and new. Integrated with a blockchain layer, it can offer great potential to facilitate the secure delivery of services, such as waste collection, transportation, processing, and recycling. Studying this problem from a security perspective becomes really important because waste management is a critical utility that must not be disrupted by hackers and malicious attackers.

This study developed a model of secure waste management utilizing Quantum Key Distribution and Blockchain in a smart city scenario. In order to evaluate the possibility of such scenario two different variants of blockchain consensus were implemented, one based on the Bitcoin-style Nakamoto consensus using the Proof of Work hash puzzle, the other utilizing Quantum Key Distribution mechanisms for reaching consensus via Proof of Infrastructure.

As suggested in the literature, integrating blockchain with the Internet of Things has its own problems such as energy consumption and scalability. In the implemented simulations of this study, it is shown that using the Proof of Infrastructure consensus backed by Quantum Key distribution can provide improved performance with regards to energy consumption and scalability, as well as increased security. Despite the challenges of deploying such Quantum Backbone infrastructure in a smart city, we think it is going to be beneficial for a wide variety of projects, such as this waste management scenario, which can take advantage of the quantum channels and the quantum keys for novel consensus mechanisms and also for providing communication security via encryption.
5.1 Challenges

While working on this project, some of the biggest challenges included learning the basics of blockchain from scratch, and then adjusting this knowledge to work with the BAL framework from ITMO to carry out the implementation of the scenarios and Proof of Concept artifacts.

While working on the Proof of Concept scenario which involves integration with Ethereum, it was quite a challenge to start out because setting up the environment and synchronizing with the Ethereum test network is not a quick and easy process. Eventually it has been decided to use a private isolated Ethereum network to speed up this process, as there were no real consequences of this decision that affected the PoC.

5.2 Future work

Future work in this topic may investigate how the so-called next generation of blockchain technology, such as Iota’s tangle and the Hedera’s hash graph compares with ”traditional” blockchain solutions, and investigate how they can be integrated into such an IoT waste management scenarios in a Smart City context.

Future extensions of this projects may also further develop the BAL project’s library, to include more realistic support for handling crypto-currency transactions in a verified and secure manner.

Finally, in future efforts it may be investigated what mechanisms are necessary to prevent so-called double-spend attacks in case an already active participant of the Quantum Network gets compromised.
REFERENCES


distributed blockchain: Analysis, requirements and open issues,” *Future Generation
Computer Systems*, 2019.

blockchain for Internet of Things,” *Computer Communications*, vol. 136, no. January,

[34] O. Alphand, M. Amoretti, T. Claeyts, S. Dall’Asta, A. Duda, G. Ferrari, F. Rousseau,
B. Tourancheau, L. Veltri, and F. Zanichelli, “IoTChain: A blockchain security ar-
chitecture for the Internet of Things,” in *2018 IEEE Wireless Communications and
Networking Conference (WCNC)*, pp. 1–6, 2018.

[35] P. Shor, “Polynomial-Time Algorithms for Prime Factorization and Discrete Log-
arithms on a Quantum Computer,” *SIAM Journal on Computing*, vol. 26, no. 5,

[36] C. Bennett and G. Brassard, “Quantum cryptography: Public key distribution and
coin tossing,” in *International Conference on Computers, Systems and Signal Pro-

[37] A. V. Gleim, V. I. Egorov, Y. V. Nazarov, S. V. Smirnov, V. V. Chistyakov, O. I.
Bannik, A. A. Anisimov, S. M. Kynev, A. E. Ivanova, R. J. Collins, S. A. Kozlov, and
G. S. Buller, “Secure polarization-independent subcarrier quantum key distribution
in optical fiber channel using BB84 protocol with a strong reference,” *Opt. Express*,

ucci.it/docs/QC-Pros+Cons-0.4.pdf [Online; accessed 27-May-2019].

[39] Arthur Herman, “Winning the Race in Quantum Comput-

by the No-Cloning Theorem of Quantum Mechanics,” arXiv:1604.01383 [quant-
ph], 2016.


awareness of potential sustainability effects of software systems in requirements engineering,” in *International Requirements Engineering Conference (RE19), South Korea, Sept. 23-27 2019.*
APPENDIX 1. Emulation outputs in Mininet

Listing 5: Mininet emulation of a tree topology with depth of 3.
APPENDIX 2. Proof of Infrastructure sequence diagram

Figure A.1: Proof of Infrastructure sequence diagram for the consensus process with 4 blockchain nodes.