University of Sheffield

Securing IoT Applications through Decentralised and Distributed IoT-Blockchain Architectures

Subhi M. Alrubei

Main Supervisor: Edward Ball

Industrial Supervisor: Jonathan Rigelsford

A thesis submitted in partial fulfilment of the requirements
for the degree of PhD in Electronic and Electrical Engineering

in the

Department of Electronic and Electrical Engineering

February 6, 2023

Declaration

All sentences or passages quoted in this document from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure.

Name: _____

Signature: _____

Date: _____

Abstract

The integration of blockchain into IoT can provide reliable control of the IoT network's ability to distribute computation over a large number of devices. It also allows the AI system to use trusted data for analysis and forecasts while utilising the available IoT hardware to coordinate the execution of tasks in parallel, using a fully distributed approach.

This thesis's first contribution is a practical implementation of a real world IoT-blockchain application, flood detection use case, is demonstrated using Ethereum proof of authority (PoA). This includes performance measurements of the transaction confirmation time, the system end-to-end latency, and the average power consumption. The study showed that blockchain can be integrated into IoT applications, and that Ethereum PoA can be used within IoT for permissioned implementation. This can be achieved while the average energy consumption of running the flood detection system including the Ethereum Geth client is small (around 0.3J).

The second contribution is a novel IoT-centric consensus protocol called honesty-based distributed proof of authority (HDPoA) via scalable work. HDPoA was analysed and then deployed and tested. Performance measurements and evaluation along with the security analyses of HDPoA were conducted using a total of 30 different IoT devices comprising Raspberry Pis, ESP32, and ESP8266 devices. These measurements included energy consumption, the devices' hash power, and the transaction confirmation time. The measured values of hash per joule (h/J) for mining were 13.8Kh/J, 54Kh/J, and 22.4Kh/J when using the Raspberry Pi, the ESP32 devices, and the ESP8266 devices, respectively, this achieved while there is limited impact on each device's power. In HDPoA the transaction confirmation time was reduced to only one block compared to up to six blocks in bitcoin.

The third contribution is a novel, secure, distributed and decentralised architecture for supporting the implementation of distributed artificial intelligence (DAI) using hardware platforms provided by IoT. A trained DAI system was implemented over the IoT, where each IoT device hosts one or more neurons within the DAI layers. This is accomplished through the utilisation of blockchain technology that allows trusted interaction and information exchange between distributed neurons. Three different datasets were tested and the system achieved a similar accuracy as when testing on a standalone system; both achieved accuracies of 92%-98%. The system accomplished that while ensuring an overall latency of as low as two minutes. This showed the

secure architecture capabilities of facilitating the implementation of DAI within IoT while ensuring the accuracy of the system is preserved.

The fourth contribution is a novel and secure architecture that integrates the advantages offered by edge computing, artificial intelligence (AI), IoT end-devices, and blockchain. This new architecture has the ability to monitor the environment, collect data, analyse it, process it using an AI-expert engine, provide predictions and actionable outcomes, and finally share it on a public blockchain platform. The pandemic caused by the wide and rapid spread of the novel coronavirus COVID-19 was used as a use-case implementation to test and evaluate the proposed system. While providing the AI-engine trusted data, the system achieved an accuracy of 95%,. This is achieved while the AI-engine only requires a 7% increase in power consumption. This demonstrate the system's ability to protect the data and support the AI system, and improves the IoT overall security with limited impact on the IoT devices.

The fifth and final contribution is enhancing the security of the HDPoA through the integration of a hardware secure module (HSM) and a hardware wallet (HW). A performance evaluation regarding the energy consumption of nodes that are equipped with HSM and HW and a security analysis were conducted. In addition to enhancing the nodes' security, the HSM can be used to sign more than 120 bytes/joule and encrypt up to 100 bytes/joule, while the HW can be used to sign up to 90 bytes/joule and encrypt up to 80 bytes/joule. The result and analyses demonstrated that the HSM and HW enhance the security of HDPoA, and also can be utilised within IoT-blockchain applications while providing much needed security in terms of confidentiality, trust in devices, and attack deterrence.

The above contributions showed that blockchain can be integrated into IoT systems. It showed that blockchain can successfully support the integration of other technologies such as AI, IoT end devices, and edge computing into one system thus allowing organisations and users to benefit greatly from a resilient, distributed, decentralised, self-managed, robust, and secure systems.

Acknowledgments

All of my praise and thankfulness is to my Almighty god for giving me good fortune and showing me far more kindness than I deserve.

I would like to express my appreciation and deepest gratitude to my main supervisor Mr. Eddie Ball for his continuous support and guidance throughout my PhD journey. I consider myself one of the luckiest students at the university because of having Eddie as my supervisor. He has been instrumental in my development as a researcher and has created a wonderful research environment that was key element in achieving my research goals. It has been a great pleasure to spend the time of my journey under his supervision and I will always remember this period of time as one of the best time in my life because of Eddie.

I also would like to thank my industrial supervisor Dr. Jonathan Rigelsford for his support and guidance and his important role in helping me achieve my goals during my PhD journey.

A special thanks to Callum Willis who has helped me a lot during the early days of my study and guided me through an important period of this journey.

I would like to thank my mother for her love, kindness, and unending support, especially during the difficult period of the COVID-19 pandemic. Her encouragements have lifted me up during the difficult periods of my journey as a PhD student.

I would like to thank and express my deepest gratitude to my wife Rawa and to my children; Lamar, Mayar, Muath, and Firaas; for being there for me during this journey. A special thank you to my wife, in all honesty, for her patience in dealing with me during this time. I also would like to thank my mother in law for being there when our family needed her help.

Lastly, I would like to express my gratitude to all my friends and the staff at the Department of Electronic and Electrical Engineering.

Glossary

aBFT	Asynchronous byzantine fault tolerant
ACE	Authentication and authorization for constrained environments
ADEPT	Autonomous decentralised peer-to-peer telemetry system
AI	Artificial intelligence
AN	Authority node
BFT	Byzantine fault tolerance
BPIIoT	Blockchain platform for industrial IoT
CBPIV	Consortium blockchain-based public integrity verification system
CEDs	Consumer electronic devices
CoAP	Constrained application protocol
CSDS	Controller smart data system
DAI	Distributed artificial intelligence
DAG	Directed acyclic graph.
DAO	Decentralised autonomous organisation
dApps	Decentralised applications
DCAs	Data consumer applications
DHT	Decentralised hash tables
DL	Distributed ledger
DLT	Distributed ledger technology
DMLP	Distributed multilayer perceptrons
DOA	Data owners application
DoS	Distributed denial of service
DPoS	Delegated proof of stake
DSA	Digital signature algorithm
DSOs	Distribution system operators

ECDSA	Elliptic curve digital signature algorithm
EI	Edge intelligence
EOAs	Externally owned accounts
EVs	Electric vehicles
EVM	Ethereum virtual machine
FL	Federated learning
FN	Full node
G-PBFT	Geographic practical byzantine fault tolerant
GPS	Global positioning system
HDPoA	Honesty-based distributed proof of authority via scalable work
HN	Hybrid node
HSM	Hardware security module
HW	Hardware wallet
ID	Identification
IDS	Intrusion detection systems
IoT	Internet of things
IoV	Internet of vehicles
JSON-RPC	JavaScript object notation - remote procedure call
LoRa	Long range radio communication
NIST	National institute of standards and technology
OSCAR	Object security architecture
PBFT	Practical byzantine fault tolerance
PN	Participant node
PoA	Proof of authority
PoAh	Proof of authentication
PoB	Proof of burn
PoBT	Proof of block and trade
PoC	Proof of credit
PoEWAL	Proof of elapsed work and luck
PoS	proof of stake
PoW	Proof of work

P2P	Peer-to-peer
QBFT	Quorum byzantine fault tolerance
RAFT	Reliable, replicated, redundant, And fault-tolerant
RFID	Radio frequency identification
R-pis	Raspberry Pis
RSA	rivest-shamir-adelman
SBC	Single-board computer
SDN	Software defined network
SDS	Smart data system
SECG	Efficient cryptography group
SHA	Secure hash algorithm
UAVs	Unmanned aerial vehicles
UDP	User datagram protocol
UNL	Unique node list
VANET	Vehicular ad-hoc NETwork
VCF	Voting-based chain finality
V2G	Vehicle-to-grid
V2V	vehicle-to-vehicle
Wi-Fi	Wireless fidelity
WN	Worker node
3G	Third generation mobile system standard
4G	Fourth generation mobile system standard
5G	Fifth generation mobile system standard
6G	sixth generation mobile system standard

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The IoT systems typically rely on trusted entities for device management, ensuring security, and providing users with the required quality of services. The idea of a trusted centralised entity can be vulnerable to limited scalability issues, to single-point-of-failure issues, and to trust and security issues. IoT is a distributed, dynamic, and heterogeneous system, it will greatly benefit from a decentralised and self-managed technology that can eliminate these vulnerabilities. Blockchain technology is a self-managed, decentralised, and trust-less that can provide scalable, redundant, potentially autonomous, and secure solutions for IoT systems.

Blockchain can provide the IoT with a distributed means for managing the connected devices regardless of their adapted standards and communication's protocols, thus allowing for trusted interaction amongst the devices and trusted information exchange. It can provide the IoT systems with a secure platform that can be utilised for providing access control mechanisms including devices' authentication and authorisation. The immutable nature of the blockchain makes it a perfect solution for ensuring the integrity of data and provide a secure traceability and accountability within IoT systems. This will allow for secure use of the data generated from the IoT by a processing entity such as an AI engine to provide a trusted outcome as a result of using secured data.

While blockchain provides great benefits and advantages to the IoT, the integration of blockchain into IoT systems faces several challenges. For example, there is a lack of study to assess its usefulness and evaluate the impact on the overall system and the individual devices. A lack of a secure and public consensus protocol that is suitable for implementation within IoT systems and is able to harness the available power for better security. Another important integration consideration is an architecture that ensures the integrity of the data as low as the sensing layer while providing a mean for managing the end devices and provide a reliable control over them. Such an architecture should be able to support other technologies such as the AI to allow for better and secure processing of the IoT generated data.

1.1 IoT

The IoT in the age of permanent revolution holds a promise of something new and different that will affect our daily life. In 1999 Ashton was the first person to introduce the term The Internet of Things [1]. Over the years, researchers and institutes defined IoT in many ways according to their perspective. One of these definitions was introduced by the Cluster of European research projects on the Internet of Things [2] which stated that "Things are active participants in business, information and social processes where they are enabled to interact and communicate among themselves and with the environment by exchanging data and information sensed about the environment, while reacting autonomously to the real/physical world events and influencing it by running processes that trigger actions and create services with or without direct human intervention".

IoT is a network of connected smart things (i.e objects) with provisions of different services to various applications. The IoT can be built based on different architectures, however typically it consists of three layered subsystems [3]. The first layer is perception, in this layer a wide range of devices can be found, such as sensors, actuators, smart meters, and RFID tags. The presence of such devices at this layer allows the IoT system to sense and collect data and make changes within its physical environmental presence .

The second layer is the communication layer, where there are various devices that can be either wired or wireless. This layer allows the different devices from the lower layer to be connected through gateways, access points (i.e WiFi), or base stations to form an IoT network. This connectivity is enabled by different means of communication protocols such as near field communications, bluetooth, LoRa, cellular connectivity (e.g 3G, 4G, 5G, and 6G) and Ethernet. The final layer on the top of the IoT system is the applications, where IoT can be utilised to support a wide range of applications ranging from industrial to healthcare [3].

Over the years IoT systems have been growing rapidly and increasingly used by many different organisations and users within different sectors, such as health-care and industry. This presence of IoT in these sectors has offered organisations and governments a realistic opportunity to improve economical situation by enhancing its growth over the years and provides an easy way to improve people's lives in general, this is as a results of the vast amounts of useful information provided by IoT systems that can be used for better decision making.

1.2 Trust and security challenges within IoT

The IoT is different from the traditional Internet due to its unique characteristics [4][5]. IoT systems are mobile and in constant changing state where devices provide thing-related services in a constrained situation. This requires it to have the ability to be dynamic and adaptable to any change such as changes of device status, devices'

location, and network connectivity. In the IoT realm any object can be interconnected globally or with each other in large numbers. The lack of a global and open standards for the IoT hardware and software manufacturing purpose has resulted in manufactured devices based on different standards. They need to support different communication protocols to be able to communicate with different entities and among themselves to achieve the desirable functionalities of an IoT system. IoT system can scale up by encompassing a large number of devices that are higher than currently connected to the Internet. These devices generate vast amount of data that need an efficient management method to handle and process.

According to [6] by 2023 there will be about 43 billion devices connected to the IoT. These devices will be utilised in many applications providing different services to users; Fig.1.1 shows some of the IoT applications. To provide these services, IoT requires a platform for various smart objects to gather and exchange information amongst themselves and with human users. Many IoT devices may have limited power resource, are computationally constrained and have low storage capacity yet are generating large amounts of data which makes them difficult to secure, vulnerable and an easy target for intruders and attackers resulting in many security and privacy issues [7].

This poses significant challenges in addressing the security and the privacy of these

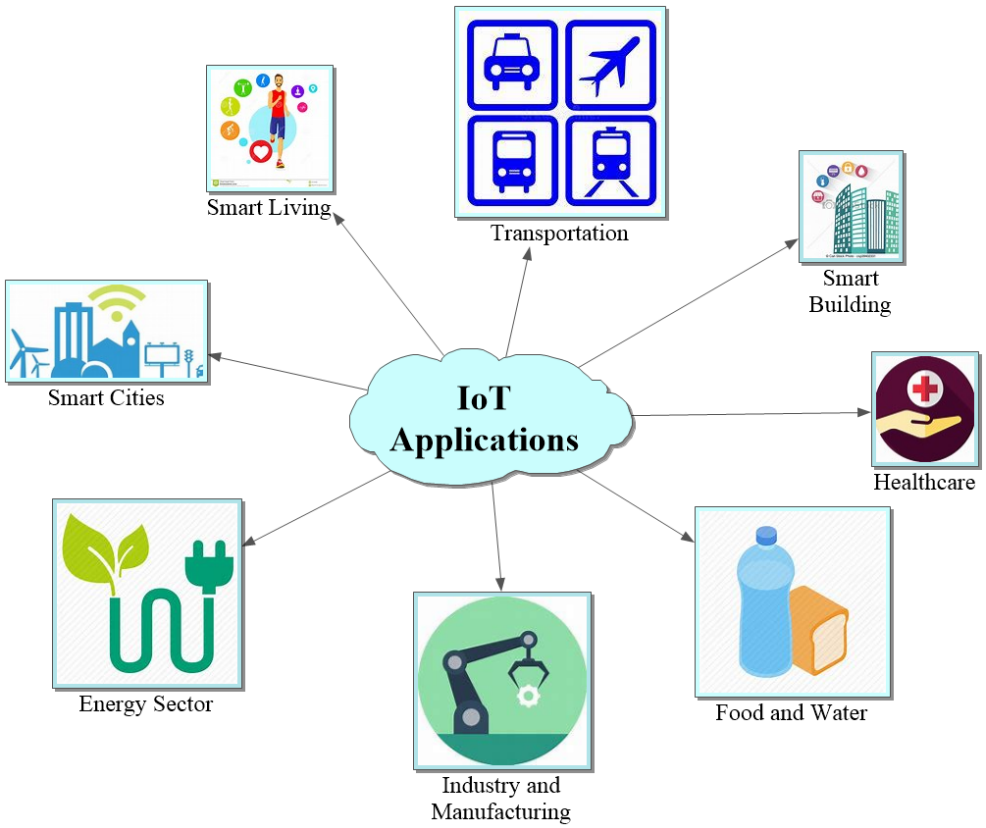


Figure 1.1: Some examples of IoT applications.

devices and data:

1. The first challenge is related to the distributed nature of IoT systems, which means that each device connected to the network is a possible entry point and its vulnerability can be exploited by an intruder to launch an attack on the network, such as denial of service (DoS) [8].

2. Typically, IoT system trusts a central entity such as cloud service provider for data processing, security and system management. This central entity introduces the risk of single point of failure that could affect the system performance and security.

3. IoT systems are utilised in many applications such as vehicular networks, manufacturing automation and smart grids where real time processing is an integral part and this requires system availability all the time [9]. This makes it very important to resolve the issues surrounding the use of a central entity for better system performance and security.

4. Authentication of devices and users is another significantly important challenge [10]. This challenge, in the presence of technology's advancements, which allows devices to exchange data for resources such as power and electricity, makes it important to find solutions to ensure devices safety.

5. One of the most important advantages of an IoT system is the ability to collect data from many sensors and use them for making timely decision [9]. This requires the preservation of the integrity of these data to protect against false data that could be used to make decisions.

The above mentioned challenges shows that for better security and privacy, new protocols, measures and solutions are required to overcome these challenges that may hinder the expansion and adaptation of IoT. Traditional security measures that are currently implemented within IoT are built around the concept of trusted centralised architectures [11]. This means such solutions will suffer from limited scalability, high cost, and single point of failure. Conversely, self-managed, decentralised, trust-less architectures provide scalable, redundant, potentially autonomous, and secure solutions for IoT systems.

One of the most notable trust-less and decentralised architecture is the distributed ledger technology (DLT), namely blockchain. DLT has been around for a long time; in 1991 the authors of [12] wanted to develop a system to timestamp documents without modifying them, so they proposed a solution based on cryptographically hashing a chain of items. Nevertheless, it was not until 2008 that blockchain was reintroduced in a popular form through Bitcoin and cryptocurrency [13]. Since then blockchain has attracted a lot of attention especially in the financial world because it was first built as a mean for executing financial transaction without the presence of a trusted third party. However, lately both in business and in academia the prospect of this technology

is being explored outside the financial sectors in many different areas and one of these areas is IoT.

Blockchain provides a robust, secure and decentralised platform for trustful interactions and information exchanging anonymously between devices and humans. Since IoT is a distributed, dynamic, and heterogeneous system; it will greatly benefit from the integration of decentralised self-managed and regulated blockchain networks [11] [14]. The following potential benefits can be provided by blockchain:

- The integration of blockchain into the IoT will result in a decentralised system that is fault tolerant and will eliminate the centralised entity and the singular point of failures [11].
- The IoT-blockchain will allow for the decentralised management of identity and will provide better authorisation and authentication features. Thus fostering the device's autonomy.
- Blockchain create a trust-less environment for devices to communicate with each other regardless of their built in communication protocol and manufacturer's standards and easily exchange data. It guarantees traceability and auditability of stored data and ensures its immutability. This means users and devices can easily verify the integrity of data or software received from the blockchain platform.
- Blockchain allows for the deployment of decentralised applications (dApps) such as smart contracts. This allows the IoT-blockchain applications to securely enforce conditions and terms over interactions amongst devices and users in a decentralised approach.

Blockchain and IoT are potentially a great fit, where blockchain can offer a solution to challenges within IoT such as: data integrity, devices authentication and authorisations, system availability and robustness. However, this fit of both IoT and blockchain requires an immense effort to integrate both technologies together. This is because IoT devices may be limited in power, computation resources and storage; it also produces vast amount of raw data that needs to be processed and analysed in a suitable environment. On the other hand, DLT such as blockchain and hashgraph are newly reinvented technology and still suffer from some issues such as scalability. This results in the following challenges:

1. Lack of IoT specific consensus algorithms that is customised to fit within IoT and to ensure efficient utilisations of a devices computation power.
2. Lack of a IoT-blockchain frameworks and architectures that are capable of providing applications with the following desired security and quality of services:
    - The ability to ensure the data collected by the IoT systems are secure in terms of integrity preservation and availability assurance. Allowing for processing entity such as the AI to securely and safely utilise such data to provide useful and actionable outcome.

- The ability to allow the processing of data close to the sources of the data while providing a means for devices and services management in a secure and redundant approach.
- Provide a mean for transactions and block propagations to the nodes within the network that ensures all nodes are synchronised and their local copy of the blockchain are mirrored with the global copy. This should be achieved in an acceptable level of end-to-end system latency.
- Transactions handling based on its importance, for example a change in the environmental status that needs immediate actions, or transactions that need special attention, such as transactions that are need to be process by an AI engine.

Solving these issues will allow for new secure and robust business models and will greatly push the boundaries for IoT expansion.

1.3 IoT and the artificial intelligence

The IoT is a major source of big data, which are generated from the huge number of smart devices connected to the internet. This data provides users with the ability to generate valuable information and knowledge. One promising technology in this context is artificial intelligence (AI) that can be utilised within the IoT to provide an intelligent means of processing data to produce valuable insights and predictions, and to enhance the process of decision-making automation.

Depending on the application and its requirements, the processing of these data by an AI system may be in the cloud layer, in an edge layer, and/or in the sensing layer (the smart devices). However, such intelligent implementation of AI into the IoT realm faces challenges, especially the implementation into the edge and sensing layers. In particular, these devices often lack adequate computational resources [15]. This leads to the idea of implementing AI in a distributed approach, distributed AI (DAI), to harness the available computation resources of the IoT devices in a way that result in little impact on these devices .

DAI is a subset of AI technology and according to [16] it is can be defined as "a computing paradigm that bypasses the need to move vast amounts of data and provides the ability to analyse data at the source". According to [17], three main characteristics define DAI: first, it is an approach to distribute tasks amongst nodes, secondly, it is an approach to distribute computation power, and finally, it is an approach for nodes' communications.

Due to the distributed nature of the IoT, where thousands of smart devices are available and can communicate with each other, it can offer a scalable hardware and software platform[18]. This platform can be utilised to provide an attractive ability in the form of processing data in a distributed approach and in near real-time and reduces the communication overhead needed to transfer data to a centralised entity such as the

cloud [19]. Thus, realising the benefits of true parallelism and distribution offered by AI in a fully distributed computing system.

The utilisation of IoT as a distribution platform for the implementation of AI can be complicated in terms of communication overhead, task management, and synchronisation. This is because nodes in distributed system requires to carry out a parallel computation every round and the number of rounds needed to complete the task and the number of messages exchanged between the nodes will result in a complex and undesirable situation [20]. To avoid this complexity and reduce latency by providing the AI system with historical data to facilitate future decisions, the IoT system requires implementation in an architecture that combines both decentralisation and distribution.

Blockchain technology is an ideal solution that enables distributed computing and achieves data storage in a large number of devices over a wide area network. The integration of blockchain into IoT can provide reliable control of the IoT network's ability to distribute computation over a large number of devices [21]. This would allow the AI system to use trusted data for analyses and forecasts while utilising the available IoT hardware to coordinate the execution of tasks in parallel, using a fully distributed approach.

1.4 Aim and objectives

1.4.1 Overall aim

The overall aim of this thesis is to investigate and study the impact of the integration of blockchain into IoT systems. It will evaluate the use of blockchain to enhance the security of IoT applications and to support the combination of AI engines with IoT and the AI engines with IoT and the edge. It aims at the evaluation of application security, the impact on the devices' power sources and the overall system latency and transaction's confirmation time.

1.4.2 Objectives

1. Many of the current blockchain and IoT integrations provided great solutions for many issues within loT, however most of them lack the complete performance analyses to evaluate the impact of this integration on the IoT devices and services. According to the authors of [22], who provide a comprehensive systematic literature review and analysis of blockchain solutions for IoT, most studies have not measured the complete transaction time from submission until the transaction is committed in the blockchain network. The authors of [22] also state that, for better performance analyses, 'the performance of the whole proof of concept should be analysed from end to end, from the transaction being submitted until the transaction being included and committed'. ***This leads to the first ob-***

jective of this thesis which to study and evaluate the integration of blockchain-IoT application using a real-world use case and to provide performance analyses of the system latency, network synchronisation and stability, and energy consumption.

2. The integration of IoT and blockchain needs a secure consensus mechanism that ensures IoT applications benefit from such integration without any substantial impact on the IoT devices. Currently there is a lack of consensus algorithm that is customised to fit within IoT and to ensure efficient utilisations of a devices' computation power. Thus, ***the second objective of this thesis is to design a consensus mechanism that is IoT centric, public, secure, and has limited impact on the device's individual power.***

3. The IoT offers a great distributed platform in terms of hardware that can be exploited to facilitate the true parallelism and distribution offered by AI. However, distributed computing requires each node in the system to carry out a parallel computation every round [20]. The number of rounds needed to complete the task and the number of messages exchanged between the nodes will result in a complex and undesirable situation. To avoid this complexity and reduce latency by providing the AI system with historical data to facilitate future decisions, the IoT system requires implementation in an architecture that combines both decentralisation and distribution. Blockchain technology is an ideal solution that enables distributed computing and achieves data storage in a large number of devices over a wide area network. This leads to ***the third objective of this thesis that is to provide a secure architecture based on blockchain to facilitate the implementation of the distributed AI over the IoT systems.***

4. The data collected by the IoT systems requires security especially in terms of integrity and availability and an integration of a distributed and secure system such blockchain can achieve such security features. With integration of a distributed, self-managed, and decentralised network, both the dynamic and distributed IoT system and the intelligence AI engine will benefit greatly from such integration [11][14]. With the presence of edge computing many benefits such as, providing IoT networks with a reliable ability to control the distribution of computation requirements over large number of distributed devices, improving the security posture of the overall IoT system by enhancing its ability to ensure data integrity and availability and ensures accountability [23], and enhancing the AI engine's ability to perform the required analyses and provide desire outcomes using these trusted data. Hence, ***the fourth objective of this thesis is to design, develop, deploy, and test a secure architecture that combines blockchain, AI, IoT end device, and edge layer into one system. This system will have the ability to monitor the environment, collect data, analyse it, process it using an AI-expert engine, provide predictions***

and actionable outcomes, and finally share it on a public blockchain platform.

5. While the integration of blockchain into IoT (IoT-blockchain) can enhance the IoT security, due to the nature of IoT devices some vulnerabilities, such as those related to the physical security, key generation, management, and storage, and the lack of trust in host systems can exist. Hardware security can provide protection to the IoT devices and enhance the overall security on the IoT-blockchain platform and the data held on it. This makes the evaluation of the viability of the integration of hardware-based security into IoT-blockchain and analyses its performance and resilience an important topic. ***The final objective of this thesis is to evaluate the viability of integrating a hardware secure module (HSM) and a hardware wallet (HW) into IoT blockchain applications and provide performance analyses in terms of power consumption and relevant security evaluation.***

1.5 Contribution

This thesis provides the following contributions:

- A study of real world IoT-blockchain application, namely flood detection, using the Ethereum platform. This included testing of physical deployed IoT devices that were part of the Ethereum platform. The performance analyses of the application included the measurement of the transaction arrival time and the system end-to-end latency, the network stability and node synchronisation, and the IoT device's energy consumption measurements. The security analyses and risk assessments of possible threats were provided as well (Chapter 3).

- A new consensus mechanism called Honesty-based Distributed Proof of Authority (HDPoA) via Scalable Work is proposed. A proof of concept permissionless blockchain system for testing and evaluating HDPoA using low cost and low power IoT devices was designed and implemented. A performance evaluation of HDPoA important parameters and the security analyses of possible attacks to HDPoA including risk assessments were provided (Chapter 4).

- A novel, and secure blockchain architecture for supporting DAI implementation on low-power and low-cost IoT devices was designed and developed. This includes the design and development of a blockchain protocol based on HDPoA that include transaction and block formats. A proof of concept were implemented to test and evaluate this architecture. Necessary performance measurements and evaluations were provided (Chapter 5).

- The design and development of an architecture that integrates four different technologies: IoT, AI, edge computing, and blockchain in one system that can

monitor and sense the environment, learn, analyse data based on the requirements of the executed task, and produce actionable outcomes. The architecture was validated experimentally and performance analyses in terms of system latency, system accuracy, and energy consumption of real-world applications in the form of an early warning system for the detection of COVID-19 in sewage water were carried out (Chapter 6).

- Evaluation of the usability of hardware security modules and hardware wallets within IoT-blockchain platforms. A performance evaluation regarding the impact of energy consumption following this integration and analysis of the nodes' security and resilience to attacks while using hardware security modules and hardware wallets was provided. An example application, secure community energy trading, is proposed and discussed based on secure IoT-blockchain technology (Chapter 7).

1.5.1 Papers dissemination

Table. 1.1 below provide a list of the published paper based on this thesis.

Table 1.1: List of the published paper.

No	Paper Title	Name of Journal or Conference	Doi	Related Chapter
		Journal papers		
1	Latency and Performance Analyses of Real-World Wireless IoT-Blockchain Application	IEEE Sensors Journal	10.1109/JSEN.2020.2979031	Chapter 3
2	HDPoA: Honesty-based distributed proof of authority via scalable work consensus protocol for IoT-blockchain applications	Elsever computer network	https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3999127	Chapter 4
3	The Use of Blockchain to Support Distributed AI Implementation in IoT Systems	IEEE Internet of Things Journal	10.1109/JIOT.2021.3064176	Chapter 5
4	A Secure Blockchain Platform for Supporting AI-Enabled IoT Applications at the Edge Layer	IEEE Access	10.1109/ACCESS.2022.3151370	Chapter 6
		Conference papers		
1	Ethereum Blockchain for Securing the Internet of Things: Practical Implementation and Performance Evaluation	2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)	10.1109/CyberSecPODS.2019.8885029	Chapter 3
2	Securing IoT-Blockchain Applications Through Honesty-Based Distributed Proof of Authority Consensus Algorithm	2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)	10.1109/CyberSA52016.2021.9478257	Chapter 4
3	A Secure Distributed Blockchain Platform for Use in AI-Enabled IoT Applications	2020 IEEE Cloud Summit	10.1109/IEEECloudSummit48914.2020.00019	Chapter 6
4	Adding Hardware Security into IoT-Blockchain Platforms	IEEE Latin-American Conference on Communications 2022	https://ieeexplore.ieee.org/document/10000585	Chapter 7

1.6 Organisation of the thesis

The rest of the thesis is organised as follows. Chapter 2 provides the literature survey, and chapter 3 provides latency and Performance Analyses of Real-World Wireless IoT-Blockchain Application. Chapter 4 presents the honesty-based distributed proof of authority via scalable work (HDPoA) consensus mechanism. Blockchain to support DAI implementation into IoT and to provide data integrity is presented in chapter 5. Chapter 6 presents blockchain for supporting AI implementation and securing IoT application at the edge. Adding hardware security into HDPoA protocol is provided in chapter 7, this followed by the conclusion and the future works in chapter 8.

Chapter 2

Literature Survey

Over the past few years both in industry and academic researchers have been trying to solve many IoT related issues using blockchain. In the following subsections, key state of the art in this regards will be discussed and evaluated.

2.1 Distributed ledger technology

Data storage and management is an important part for any application, making databases a requirement. Organising data in clearly defined databases allows for the applying of different operations such as Create and Read. These operations perform a certain work on the stored data and are called ***transactions***. Each transaction is carried out in isolation, allowing for each operation to be clearly differentiated. This makes it possible for any operations on the database to be reversed in case of an error.

There are three different approaches to organising and managing databases [24]. These approaches are:

1. Centralised databases: in this type, data are stored on a single storage unit, making it easy to manage and maintain the stored data. However, this type of organisation suffers from issues that can affect its performance and availability. For instance, bottleneck issues if many operational requests need to be handled at the same time, and the centre point of failure where the storage unit can be down and no redundant unit is available. How the nodes are connected in this approach is shown in Fig.2.1a.

2. Decentralised databases: no central storage unit is used, instead data are stored over multiple units. In this approach the storage units are connected to each other in hierarchical structure with multiple nodes from lower level connected to a particular node in the higher level, Fig.2.1b shows how nodes are connected in decentralised approach.

3. Distributed databases: in this approach the data are replicated and stored across different nodes that are physically independent of each other. The nodes in the

distributed databases are connected to each other to form a mesh network (see Fig.2.1c). The dataset is in a consistent state when all the stored data are exactly the same across all nodes. This means all nodes need to agree on a particular state and therefore reach a consensus over the state of the data. As these nodes are physically not connected they need to communicate among themselves to reach this consensus. This could result in Byzantine failures [25] which can complicate the consensus process.

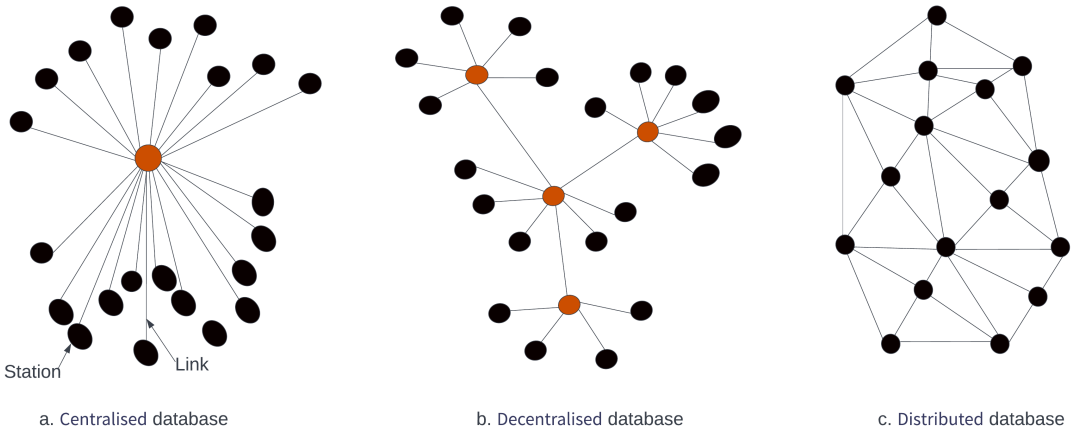


Figure 2.1: Approaches to how to organise and store data.

As described in [25] Byzantine failure refers to a situation where nodes avoid catastrophic failure by agreeing on a strategy, however, some nodes are unreliable. There are three types of Byzantine failure. First a node is not responding due to crash or is not reachable over the network, second a node's state cannot be conclusively determined, and third when a node acts maliciously. In order to overcome this failure, protocols and/or algorithms need to be in place to manage the synchronisation between nodes, these protocols are called ***consensus mechanisms***. Protocols such as Paxos and RAFT were introduced to overcome the first and the second byzantine failure [26].

Distributed ledger (DL) is a type of distributed database, where data are replicated and stored on different physically separated nodes. DL only allows for append, this means once the data are added to the ledger cannot be deleted or modified. From a security perspective DL assumes the presence of malicious nodes within the network, meaning any consensus in place needs to be resilient against the third byzantine failure (malicious node) [27]. In 2008 a solution for the third byzantine failure was provided by the introduction of Bitcoin [13] through the elimination of the central third party authority to append the data and the ability to distinguish between malicious and honest nodes on the network.

Distributed ledger technology(DLT) allows any nodes to append data to the distributed ledger and take part in the consensus mechanisms without revealing their

identities in a trust-less environment, this accomplished while the network security is preserved. The Bank of England [28] has defined DLT as "a database architecture which enables the keeping and sharing of records in a distributed and decentralised way, while ensuring its integrity through the use of consensus-based validation protocols and cryptographic signatures."

2.1.1 History of DLT

With the introduction of Bitcoin [13] cryptocurrency in 2008, one of the most notable trust-less and decentralised architectures in the form of DLT reemerged and became popular with the public and researchers. However, the idea of crypto has been around since the early 1980s. In 1983 David Chaum [29] introduced the concept of blind signatures that allows for pseudonymisation when exchanging data. This concept uses public encryption technology where each user has two keys, a public known key and a private secure key. The public key is used to prove that an entity or identity has originated and signed a certain data by decrypting its signature, allowing for data validation without revealing the real identity of the person who signed that data. By the blind signature concept Chaum managed to introduce the electronic payment system, and subsequently in 1989 founded the DigiCash company and developed eCash cryptocurrency that allows for anonymous and untraceable payment systems. However, to overcome the problem of double spending a third party in the form of banks was assigned to verify payments and ensure no double spending occurs. While at the beginning eCash was successful in 1998 the company DigiCash was dissolved. In 1991 the authors of [12] wanted to develop a system to timestamp documents without modifying them, so they proposed a solution based on cryptographically hashing a chain of items.This was another fascinating attempt to use hashing for signature chaining. Douglas and Barry in 1996 have introduced the e-gold cryptocurrency [30] which was backed by gold. While it has a high number of customers (1.4 million accounts) it has still failed short in terms of world wide and mass adaptation.

Proof of work (PoW) was invented by [31] in 1993 and later was formalised by Markus Jakobsson and Ari Juels [32]. Adam Back was the first to implemented the first form of PoW called the hashcash system [24]. PoW as described by [32] is a way in which a prover party can demonstrate that it has executed a certain amount of computational work in a certain time frame. According to [33] PoW is a preventive measure against services abuses (e.g., spamming and Denial of Service attacks) in the form of computational power. This computational power could be carried out by solving a mathematical problem before a user can access certain services. Timothy May introduced the idea of crypto anarchy [34], which involves the permanent elimination of governor entities and assumes anonymity and resists censorship, for example in regard to payments. Based on this idea in 1998 the author of [34] introduced b-money digital currency protocol. At the same time Nick Szabo introduced his initial idea of BitGold [35]. It is similar in architecture to Bitcoin in the way its structure aims at eliminating third parties, and being fully decentralised. BitGold utilised the PoW by asking the

user to solve a cryptographic task. However BitGold never was implemented and along with b-money failed to achieve mass adaptation similar to e-gold.

All of the previously discussed attempts to create a decentralised cryptocurrency have provided the main building blocks that in 2008 Nakamoto [13] have utilised and produced Bitcoin cryptocurrency, which can be considered the first generation of DLT (i.e., DLT 1.0). By the introduction of bitcoin the world has witnessed a re-emerging disruptive technology in the form of blockchain. This technology stimulated a mass adaptation of cryptocurrencies and great deal of research to integrate it into many applications, ranging from healthcare systems, internet of things , to financial systems. Bitcoin was the first distributed ledger that solved the three forms of byzantine failures. While bitcoin allows for some scripting to be executed, this capability is limited as a result of Bitcoin not being Turing complete. This leads to the introduction of the second generation of DLT (i.e., DLT 2.0). Ethereum blockchain introduced in 2015 by [36] and was the first Turing complete DLT that allows the deployment and execution of smart contracts over the distributed ledger. Since then DLT prospered and many applications and use cases were deployed and executed over DL, such as Non fungible token (NFT) and distributed gaming and lately the metaverse and decentralised autonomous organisation (DAO) was promoted as the next two big steps in this fascinating world of DLT. Figure.2.2 illustrate major events and important milestones in regard to the development of DLT.

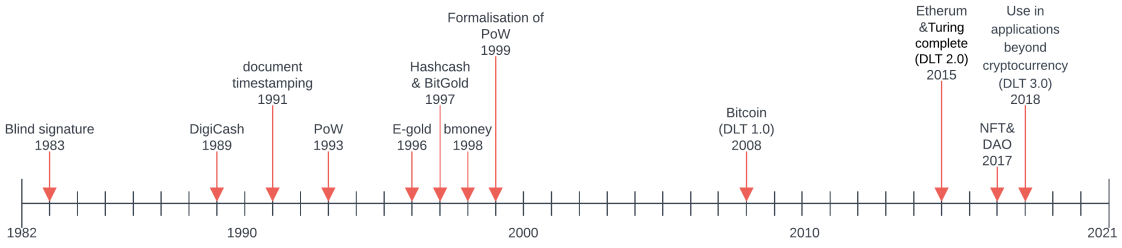


Figure 2.2: History of DLT - timeline of major events

2.1.2 Implemented concepts of DLT

DLT can be implemented in different formats based on how data are added to the ledger. ***Blockchain*** is the most well known and discussed form of DLT implementation. In blockchain, data are added to blocks and each block references the previously added block forming a chain of blocks (see Fig.2.3a). Each block consists of a header and a body. The header can contain different information about the block such as its height, the nonce (which is a random number, please see sub-subsection 2.2.2 for more detalis), the difficulty level, the size of the block, Merkle tree, the block's hash, and the hash of the previously block. The body contains all transactions. different examples of blockchain exist, more details will be provided in the next section 2.2.

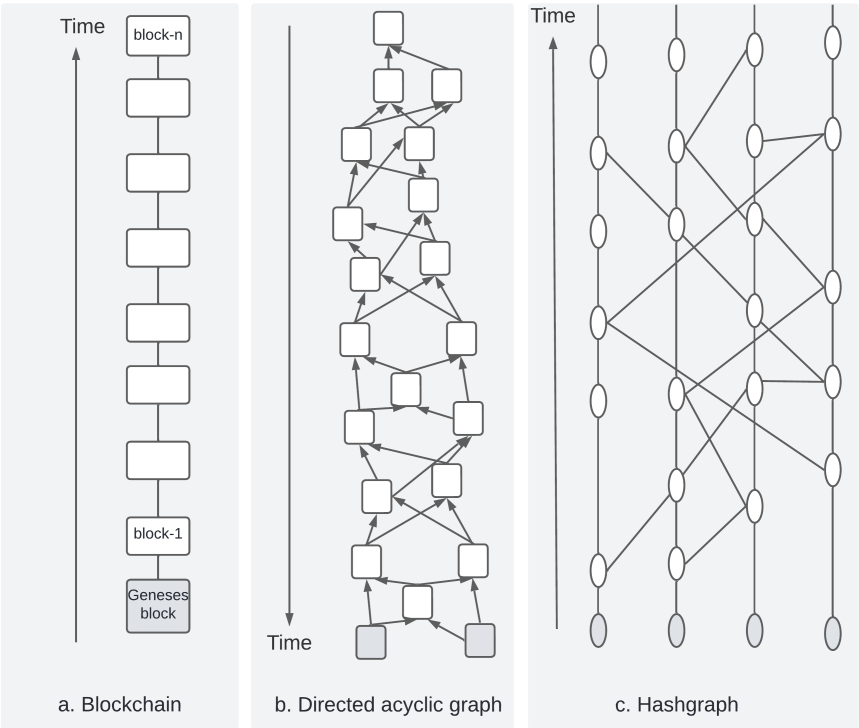


Figure 2.3: Concepts of how DLT is implemented

Another form of DLT implementation is the ***Directed Acyclic Graph (DAG)***. In DAG, the transactions issued by the nodes in the tangle constitute the site set of the graph (unlike blockchain where transactions are added to blocks), which is the ledger that store transactions. DAG is based on the idea that each new transactions added to the network should approve at least two previously added transactions on the DAG network. The network usually consists of multiple nodes that all can approve transactions, thus there is no node acting as a miner collecting and mining the next block. The more transactions added to the network the more doubly-approved transactions the network have [37]. An example of DAG based network is presented in Fig.2.3b.

One example of DAG implementation is IOTA. IOTA is a cryptocurrency intended for the IoT industry and uses the tangle protocol [38]. Tangle provides machine to machine communication features that can establish a micropayment system. First, transactions are issued by the nodes and published to the tangle. Then each transaction must approve two previous transactions. Direct approval in the tangle is called a direct edge and a transaction connected to another one by the same path is called indirect approval. The direct approval requires the node that issued the transaction to do some work in the form of a cryptographic puzzle in order to accomplish the approval; if a conflict occurs then the node will not approve the transaction. There are other example implementations of DAG such as ByteBall [39], DagCoin [40], and XDAG [41].

Hashgraph is another form of DLT. Unlike blockchain, in hashgraph data organised into events, where each event contains transactions associated with its timestamp. Events are propagated into the hashgraph network based on the gossip protocol (gossip about gossip), which allows it to create a directed graph as can be seen in Fig.2.3c. Hedra platform is implemented as a hashgraph [42]. Hedra uses asynchronous Byzantine Fault Tolerant (aBFT) consensus protocol which allows each node in the Hedra network to eventually know for sure that the network has reached a consensus on an event.

Blockchain will be the main concept that will be discussed in the reminder of this thesis.

2.2 Blockchain

Blockchain allows untrusted entities to transact and interact among themselves in a trusted and secured manner. This is accomplished without the presence of any intermediary or any centralised trusted institution. Blockchain achieves this through the use of cryptography and collaboration, which allows it to store transactions in a trusted ledger. It makes use of Public key cryptography, zero-knowledge proof, and hash functions.

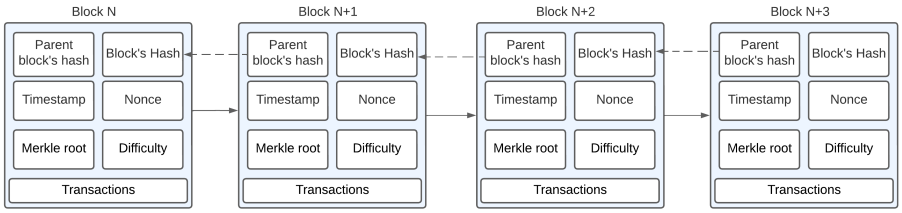


Figure 2.4: A chain of blocks

Blockchain can be defined as chain of blocks containing records of transactions with necessary data that makes it an immutable peer-to-peer decentralised technology in a trust-less environment, see Fig.2.4. A blockchain is a chain of blocks B; it can be called a digital ledger since each block contains a list of formatted transactions. Block B_n can be either confirmed (where all transactions in the block are final and cannot be deleted or edited), or unconfirmed. The ledger only contains a union ($\cup$) of confirmed blocks. The genesis block B_g is the first block in this union, and it is the only block without a previous block's hash in its header.

Beyond its usage in the financial sectors, blockchain offers great benefits to different application including the IoT due to its characteristics and the advantages it offers, which include [43][44][45]:

- Decentralisation: it will provide applications with peer to peer distributed ledger architecture. It offers the benefit of a robust and redundant environment where

the authority of one single entity is eliminated and removes the single point of failure.

- Autonomy: blockchain allows IoT devices to communicate among themselves without any intermediate servers allowing for exchange of services and data.
- Immutability: any data once added to the blockchain network will not be modified, unless verified by the majority of the nodes. This makes it difficult for an intruder to change or alter those data.
- Security: it provides a reliable identification and authentication mechanism in the form of public encryption keys. It can be leveraged for authorisation of devices accessing data based on sets of rules through the use of 'smart contracts'.
- Cost effective: the current IoT architecture typically relies on a central entity. This can cause increase of cost due to the use of infrastructures and maintenance. On the other hand, the distributed nature of blockchain would eliminate the costs associated with such architecture.

2.2.1 Type of blockchain

Based on access, the blockchain can be divided into two different types:

- Permission-less blockchain: it is also called public platforms. In this type any node can join freely without needing any authorisation from any one. Nodes within these platforms can publish blocks, issue and propagate transactions, and freely read and host the full chain of the blockchain (i.e., nodes can read and write without permissions). Usually the software of these blockchain platforms is an open source where any node can freely download. Due to the openness nature of these platforms, malicious nodes may be tempted to issue and propagate unlawful blocks in a way that subverts the network. In order to reduce or eliminate the risks from these malicious nodes, in public blockchain a consensus mechanisms is utilised where majority of the nodes on the network need to agree on the validity of any new block [46].
- Permissioned blockchain: where nodes must be pre-authorised by an authority to propagate new blocks and be able to access the data on the platform. Some nodes might be authorised to do both propagating new blocks and transactions and reading data and some nodes may be authorised to read data and propagate transactions, and others may be only allowed to read data. The permissioned blockchain platforms use consensus mechanisms, however these mechanisms often do not need the computation power or available expensive resources as in permission-less blockchain because private platforms usually can be controlled by a centralised authority.

While these platforms suffer from the issue of the trust in the centralised entity, nevertheless, these platforms provides the data and asset traceability, data storage redundancy, resilient, and distribution the same as with the public platforms [46].

2.2.2 Components of the blockchain

Public key cryptography

Public key cryptography (also known as asymmetric cryptography) is based on the idea that each entity has a pair of keys, the public key and the private key. Private key is kept a secret while the public key is published and shared with other entities. One of the most well known implementations of public key cryptography is based on the Rivest-Shamir-Adelman (RSA) algorithm [47] and Elliptic Curve Digital Signature Algorithm or ECDSA which is the elliptic curve analogue of the Digital Signature Algorithm (DSA) [48]. In blockchain the private key is always securely stored in a digital wallet either a hardware or a software. The use of the public key cryptography allows the entity to accomplish the following functionalities:

- The encryption and decryption of data which allows for secure communications between two entities. Sender uses a receiver public key to encrypt the data and then send it. The receiver once it receives the encrypted data will use its own private key to decrypt the data.
- Signing data which prevents nonrepudiation. Sender uses its own private key to sign the data while other entities can validate the signature using the sender's published public key.

ECDSA is the most used cryptography algorithm in blockchain, especially the secp256k1 [49]. Bitcoin uses a specific Koblitz curve secp256k1 defined by the Standards for Efficient Cryptography Group (SECG). The curve is defined over the finite field F_p [50]:

$$y^2 = x^3 + ax + b \tag{2.1}$$

With $a = 0$ and $b = 7$ [50], the y-coordinate of a point on the SECP256K1 curve can be calculated by $y^2 = x^3 + 7$.

Koblitz curves allows for fast computation and complex multiplication through the use of τ-adic expansion, and features many advantageous characteristics when used in elliptic curve crypto-systems [50] .

Public key cryptography within blockchain technology provides important functionalities for example, private keys used to perform the digital signing of transactions, public key is used for the addresses derivation and signature verification that was generated by the relevant private key.

Hash functions

The hash function is a mathematical operation that has the following security properties [51]:

- Hash function is a fixed size where it can take any input value and always produces a fixed size output, making it suitable to represent any size of data with fixed size output. Blockchain benefits greatly from these properties when performing digital signatures and representing blocks using hash functions.
- Preimage resistance. When hashing an input data it is easy to produce an output in the form of hash and when trying to reproduce the same input data using the output hash it is mathematically impossible to reproduce the input data. This makes it resistant to reverse engineering attacks.
- Hash functions are collision resistant. This means it is computationally infeasible to produce the same hash using any two distinct inputs.
- Another important property of hash functions is the fact that any changes to the input data even if it is a single bit the produced output will be completely different from the original one.

In blockchain hash functions are key technology, one of the widely used is the Secure Hash Algorithm (SHA), and more specifically the SHA with 256 bits as an output size (i.e., SHA-256). The SHA-256 is fast to compute because it is widely supported by hardware [46].

$$SHA256\ output = 32bytes = 256bits \tag{2.2}$$

This means there are 2^{256} (i.e., over $115 * 10^{75}$) possible digest values, making it highly unlikely that a collision can occur. The output usually represented in a hexadecimal string of 64-character.

Block

A block contains a header and body; the header has different fields such as, the hash of the previous block (sometimes called the parent block), the block own hash, a timestamp, a nonce, and the Markle tree root. The body of the block contains the transactions (see Fig.2.5) [52]. The first block in the chain is usually called the genesis block and has no parent block (i.e., no previous block's hash). Blocks are chained together by including the previous block's hash into the new block's header, hence the blockchain. This would ensure the security of the chain as if any block's hash changed, all of the subsequent blocks, hashs need to change as well, making it very difficult to change validated data as it would be very easy to detect any changes to the block header and nodes will reject it.

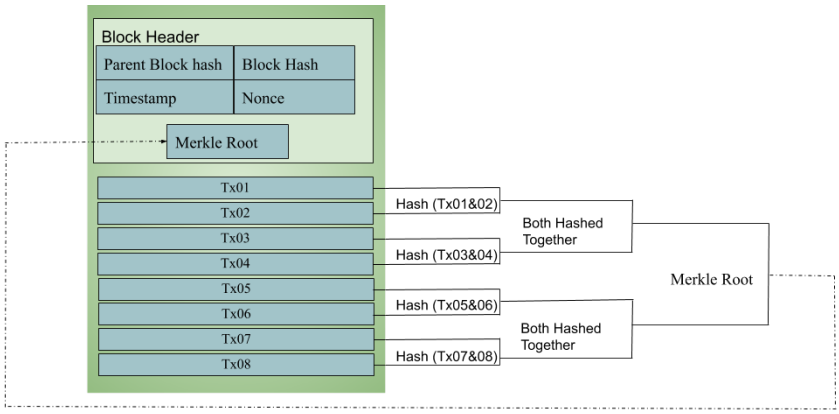


Figure 2.5: Block header, body and Merkle tree root.

Transaction

A transaction can be thought of as single bank statement that represented the transfer of an asset (money) from one person to another. Blockchain's transactions can carry a financial transfer, or data that are used to trigger the execution of a code (i.e., a 'smart contract') [52]. Based on the implemented blockchain platform and its requirements the transaction can be adapted to carry different data. However in general the mechanisms for transacting are similar. A user sends information to the platform that can include the sender's address and public key, its signature in digital format, and transaction's input and output [46].

Difficulty

The difficulty can be defined as how difficult it is for any node to find the hash that is below or equal to a given target hash value [53]. It represents how much computation power the node must put on in order to find the block's 'hash'. Usually the difficulty depends on how many leading zeros are at the beginning of the hash.

Merkle tree root

In blockchain, a Merkle tree root is constructed using the transactions' IDs. The transactions' IDs are placed in order, then a pair of IDs is concatenated together and hashed to form a second row of hashes. This process is applied to each row until it produces only one hash; which is the Merkle tree root (see Fig.2.5) [52].

Nonce

A nonce is a random number that is only used once. The nonce can be combined with other fields from the block such as the block height, Merkle tree root , and the previous block hash to create the new block's hash. In order to solve the cryptographic puzzle,

the miner, which is a node that collects and validates transactions, and mutates the nonce to change the header hash until it finds the right solution [52]. By only changing the nonce to find the correct hash, the other fields along with the data will be kept the same.

Forking

In blockchain, forking can be used to describe changes to the blockchain software. There are two type of forking; soft forking and hard forking. Soft forking is when the blockchain platform is updated where the update is backwards compatible, meaning un-updated nodes can participate and interact with updated nodes without any problem. Hard forking is an update or change to the blockchain platform that is not a backwards compatible, meaning that at a specific point, usually at a specific block height, all nodes must switch to the latest version of the software and make sure their version is updated to the latest software release. Unlike soft forking nodes that did not implemented the changes or the update will not be able to transact with the other nodes [46].

Forking also can be used to describe the case when two blocks are released on the network at the same time [54]. This will result in two chains on the network and this could create problems, especially from the security point of view and can be used to initiates double spending attack.

Addresses

In many blockchain platforms an address in the from of string of characters can be driven from the node's public key using a hash function. Another data such as checksum and version number can also be used along with the public key to generate the address. These addresses play a vital role within a transaction when populating the 'to' and 'from' fields. These addresses usually act as the public identifier for the participant node or user on the blockchain platform making it easy for other users to interact with the node or the user.

2.2.3 Blockchain platforms

In this section the important features of some of the well-known blockchain platform will be discussed.

Bitcoin

Bitcoin is a digital currency based on the blockchain technology introduced by [13] in 2008. Bitcoin implements PoW consensus algorithm to mine and validate blocks and ensure the security and data integrity of the network. Users are identified by their public key as their addresses and its operation follows the description in Fig.2.6.

In bitcoin all transactions happening on the network are recorded and made public in an immutable and decentralised distributed ledger. Nodes are free to join and leave

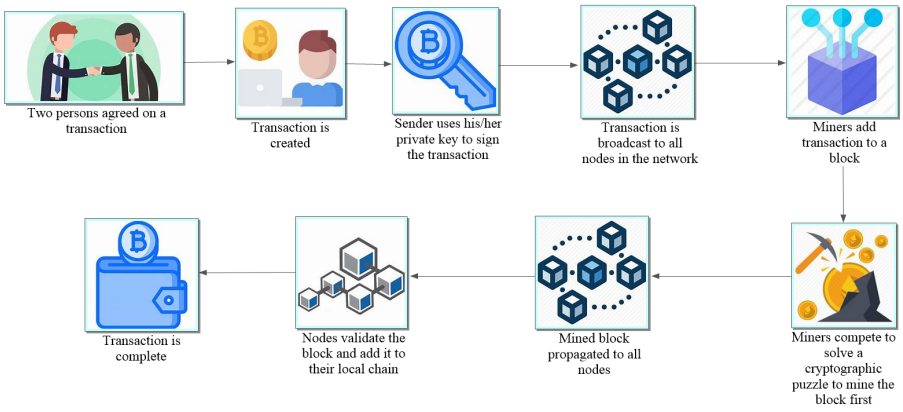


Figure 2.6: Typical process of any transact between two users within bitcoin blockchain.

the network as needed; once a node joins the network it accepts the current chain by participating in extending that chain by adding new blocks. By refusing to participate in adding new blocks the node announces its rejection of the current chain. Chains in bitcoin are computationally difficult for an attacker to change or manipulate unless they control the majority of the nodes' hash power. Bitcoin, with the use of PoW protocol, requires a lot of energy and computation power, and this makes bitcoin undesirable and difficult to implement in the IoT realm.

Ethereum

Ethereum is an open blockchain platform that allows users to program and execute their distributed applications (dApps) [55]. Ethereum is one of the first blockchain platforms that implement the virtual machine on the blockchain by having its own Ethereum Virtual Machine (EVM). EVM allows for the execution of arbitrary code and is the core of Ethereum which gives users the ability to decide how and for what their dApps can be used. Ethereum has two sets of accounts; accounts owned by private keys controlled by users called Externally Owned Accounts (EOAs), and Contract Accounts that users can activate using their EOAs by sending instructions in the form of transactions. Ethereum has its own currency called Ether, which is used to pay for miner rewards. Crypto fuel called 'Gas' is used to pay for the execution of operations such as transfer of digital currency or the execution of smart contract logic. Ethereum uses different consensus algorithms such as PoW, proof of stake (PoS), and proof of authority (PoA); the later one consumes the lowest resources. Ethereum is an easy platform to deploy on many architectures, including ARM-based Linux systems. Ethereum PoA is a suitable implementation within IoT in comparison with other platforms, the only drawback is that it is a permissioned protocol.

Hyperledger

Hyperledger is an open source platform under the umbrella of Linux Foundation [56]. Its architecture is a modular and configurable and adaptable for different applications such as, financial and supply chain. It supports smart contract that can be written in standard programming languages (for instance, Go, and node.js) making it easy to develop and deploy. Hyperledger supports pluggable protocols including consensus and identity management. Hyperledger has its own architecture for handling transactions called execute-order-validate.

Quorum

Quorum is a hard fork from the go-Ethereum platform and was built by JP Morgan as a platform for organisations to use. It supports both private and public transactions and private and public contracts through the separation of public and private states. It offers the implementation of different consensus mechanisms. It can be implemented using Quorum Byzantine-fault-tolerant (QBFT) consensus which is an improved BFT, using PBFT, clique PoA, or reliable, replicated, redundant, and fault-tolerant (RAFT) based consensus. This allows organisations to adapt the platform according to their needs [57].

2.3 Consensus mechanisms

Different consensus algorithms are being utilised by blockchain platforms and in this section, the features of some of the most common will be explored.

2.3.1 General-purpose consensus mechanisms

Proof of work

PoW was the first approach implemented within blockchain in bitcoin platforms [13]. It is permission-less and allows for building secure and public platforms. Nodes have the freedom of joining the network and leaving it as needed. The process of generating blocks is called mining and it requires nodes to compete among each other to solve a cryptographic puzzle based on varying a nonce to find the hash of the block. The first node to solve the puzzle is called the miner, which can mine the block and release it to the network. Other nodes can verify the hash of that block using the nonce that has just been discovered by the miner. The miner of the block receives a reward in the form of digital currency. This process is dictated by the difficulty which is adjustable based on block generation rate. PoW difficulty is adjusted to ensure that the average between two consecutively realised blocks is equal to the generation rate. PoW is a secure algorithm as long as honest nodes form the majority of the network, but the

computation power required for the PoW is increasing which results in higher energy consumption [58].

Proof of stake (PoS)

PoS was introduced as a possible replacement to the PoW due to its lower use of energy [59]. A mining process is conducted based on currency ownership, the higher stake of the currency the node has - the greater its chance to mine the next block and collect the reward. PoS in comparison with PoW is a reduced energy algorithm since there is no computation power needed to solve cryptographic puzzle. Nevertheless, this is a consensus disadvantage to other nodes that don't have higher stake of the currency, which will result in rich nodes becoming richer. It also vulnerable to 'Nothing at Stake' attack, where nodes could mine multiple blocks resulting in different forks, to generate more rewards [60].

Practical byzantine fault tolerance (PBFT)

PBFT is an algorithm design to tolerate Byzantine faults [14]. A primary is selected in each round and collect transactions. For the node to enter the process of mining, it should receive 2/3 of available nodes' votes. Nodes cannot join the network freely because of this voting process. PBFT is ideal for a permissioned network and also consumes less power and provides energy saving.

Proof of authority

PoA consensus protocol is one of the Byzantine fault tolerant algorithms family [25]. This protocol is mainly used in permissioned network; it is a simple protocol that does not involve any extensive computation works such as finding the nonce to mine blocks. The network relies on trusted nodes, called authorities, to mine and propagate blocks. The block header contains 65-byte field called 'extra-data' to store validators' addresses and another field for voting protocol which can be used to remove existing validators or to add a new one.

Within the Etherum blockchain there are two different protocols based on PoA; aura and clique. The process of confirming the transactions for both protocols is presented in Fig.2.7. Clique protocol does not requires any confirmation round as nodes once they receive the new block will validate it and if valid they will immediately add it to their local chain; otherwise it will be rejected and a round for voting will be required.

Proof of activity

Proof of Activity is another consensus protocol that is a combination of PoW and PoS [24]. This protocol works in two rounds. Firstly, nodes apply the PoW protocol to find the nonce and mine the block; then follow-the-satoshi function [24] is used to choose validators based on the PoS. The nodes with higher stake of digital currency have

higher chance of being chosen as validator. Validators then validate and sign the block and propagate it to the network. The reward and fees are shared among the validators and the miner. This protocol is complex, time consuming, and does not solve all the problems resulting from using PoW and PoS consensus.

Ripple

Ripple proceeds in rounds, initially, each server takes all valid transactions and makes them public in the form of a list known as the "candidate set". Then each server merges the candidate sets of all servers in the Unique Node List (UNL); UNL is a set of trusted nodes in the ripple network, and votes on the validity of all of these transactions. The transaction that passed this initial round will be carried over to the next round. In the final round transactions to be considered valid need to receive at least 80% of the UNL servers votes. Then all of these valid transactions will be appended to the ledger and that ledger is then closed and considered the new last closed ledger. Ripple is secure as long as more than 80% of the servers are not faulty or malicious [61].

Delegated proof of stake

In delegated proof of stake (DPoS), witness is an authority that is allowed to produce and broadcast blocks. At the end of each block a witness is assigned randomly. The assigned witness will collect transactions and add them to a new block and then sign it using its private key. Then a voting process applies where users can vote through

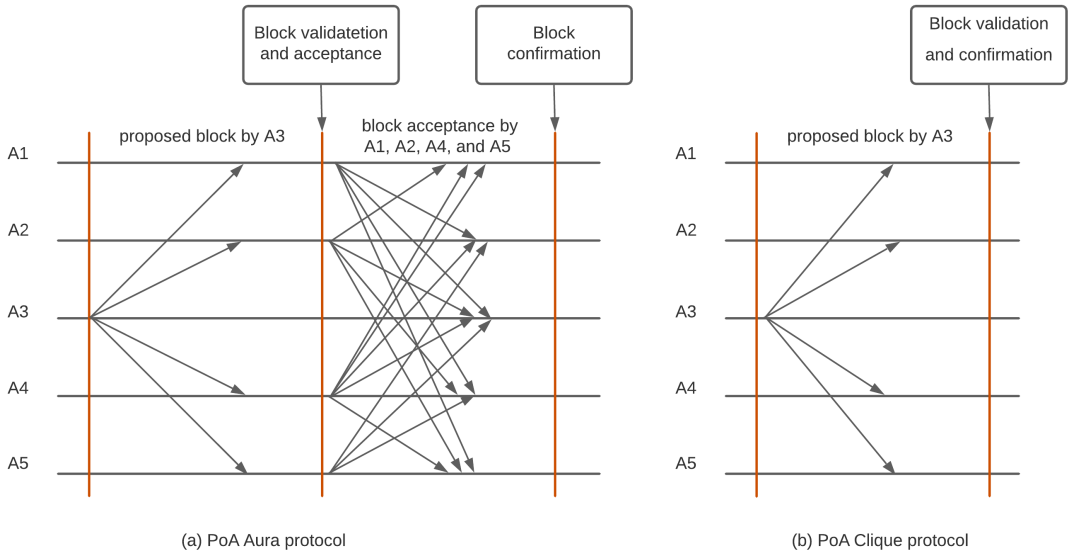


Figure 2.7: Proof of Authority Aura and Clique protocols — block acceptance and confirmation process

elected delegates. Users are incentivised and penalised based on their generated block [62, 63].

Proof of burn

Proof of burn (PoB) is another consensus protocol that was developed to replace the PoW [64]. It is based on the idea of burning coins in order to gain more mining power on the network. A miner to accumulate mining power instead of buying hardware devices it will send coins to a dead wallet or what is called an eater address. This wallet has a public key that is not associated with any private key making the recovery of coins from this wallet impossible. The more coins the miner burns the more mining power it will acquire giving it more chances to mine blocks and be rewarded [64]. While this protocol reduces the need for the hashing power that is abused on hardware and power consumption, it is very expensive as miners need to spend coins to make investment in the blockchain and show honesty and commitment.

Casper

Casper is a protocol that combines PoS and PoW into one consensus. It works by incorporating a set of validators along with a proposed mechanism based on PoW. It works by the concept of checkpoints where instead of dealing with the full block tree it only considers subtree called checkpoint tree. Validators need to deposit coins on the network in order for them to vote on new blocks and receive rewards [65]. Casper inherited some of the problems of both PoS and PoW such as the richer will be rich and the 51% attack.

PAXOS

PAXOS is a primitive consensus mechanism and one of the very first and is based on the idea of selecting a single value below the faulty circumstance of the network. It classifies the nodes into three different types; the proposers, the acceptors, and the learners. It works by first choosing a leader that is a distinguished proposer and a distinguished learner at the same time. This proposer will then send a message containing a proposal value to a set of acceptors. Acceptors compare the value with the current one on the network and once the majority (at least $N/2-1$ of N acceptors) accept this value, it will be the highest value on the network. Acceptors will then be required to respond by messages indicating that they accepted the value to the proposer and to all learners. This would allow the learners to know the choosing value [66].

RAFT

RAFT is a consensus mechanism for the management of a replicated log. It is supposed to be a simpler version of the PAXOS consensus that is easy to understand and implement. It works by first selecting a leader that is called the distinguished leader.

This elected leader will then have complete control over the acceptance of logs entries (i.e., transactions) from clients and then replicate them on the other servers within its cluster (i.e., in blocks) and finally inform these servers of the safe time to commit these changes in their local machine. Each cluster usually consists of five servers [67]. By having a distinguished leader that controls the consensus process, forking can be eliminated, however in the case of leader failure a delay can result in terms of commenting on new changes.

2.3.2 IoT-centric consensus mechanisms

Credit-based PoW

The credit-based consensus mechanism was introduced by [68]. The main idea is to run PoW based on credit, each node has a credit value and this credit increases as long as the node behaves honestly and any node that misbehaves has its credit decreased. The difficulty of the PoW is based on the credit the node has — as the credit increases the difficulty decreases. This approach might help to ensure nodes obey the rules and misbehaving nodes will have to pay penalties in terms of resources. In this approach a node with higher credit will be the one to mine the block first. However, as time progresses these nodes get richer in terms of credit and might collide to cause malicious activities (i.e., Collusion of rich stakeholders similar to the PoS). Also it is not clear how the network will deal with new honest nodes that joins the network at later stages and want to participate in the mining process and be rewarded.

Proof of trust

Similar to [68], the authors of [69] proposed proof of trust (PoT) consensus mechanism where the concept of trust graph is implemented. The node with the higher trust will be able to mine block at lower difficulty rate, the higher the trust level the node has the lower the difficulty for mining block. This can result on a more centralised node where only a few nodes control the networks.

PoW-based hybrid-IoT sub-blockchains

By combining BFT and PoW the authors in [70] showed through simulation experiments that IoT devices such Raspberry Pi can participate in PoW blockchain as full nodes, which store a full copy of the blockchain, initiate transactions, mine and validate blocks. They provided simulated performance analysis of blockchain implantation in IoT that included transactions throughput, average traffic on the network and stall blocks. They proposed hybrid-IoT sub-blockchains architecture based on a set of rules, see Fig.2.8. The sub-blockchains use PoW as a consensus algorithm and BFT protocols for inter-connectivity between sub-blockchains. However, this framework imposes some restrictions such as the number of nodes per sub-blockchains, the block size and

the location of nodes. Within IoT for such a framework there could be a need for hundreds of sub-blockchains to accommodate all devices increasing the interconnectivity overhead.

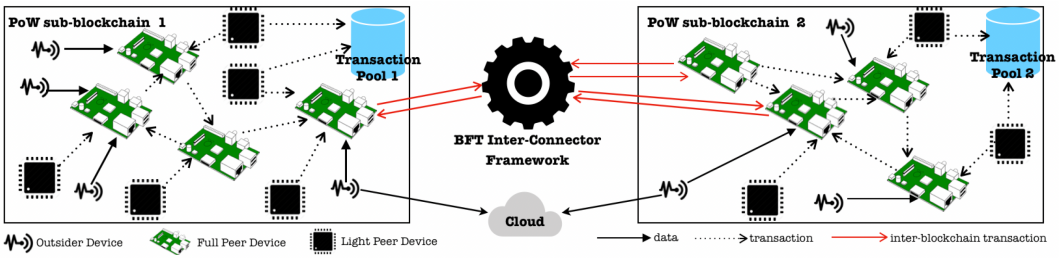


Figure 2.8: PoW-based hybrid-IoT sub-blockchains [70]

Microchain

Based on the concept of proof of credit (PoC) combined with voting-based chain finality (VCF), the authors of [71] have proposed another consensus mechanism for IoT systems called Microchain. It uses a subset of validators to run and manage the consensus process. However this protocol tends to rely on trusted nodes; it is easier for nodes with higher trust values to mine blocks. As a result such a blockchain network may become more central, where it may be possible that the network is controlled by just a single or a few nodes.

Proof of block and trade (PoBT)

The authors of [72] have proposed a new consensus mechanism for IoT applications called proof of block and trade (PoBT). PoBT was designed using the Hyperledger Fabric as the baseline platform. While it provides a good solution for trading validations it seems to introduce two different chains; local trades and global trades, this may introduce problems related to chain synchronisation.

Geographic practical byzantine fault tolerant (G-PBFT)

Work by [73] proposed the geographic practical Byzantine fault tolerant (G-PBFT) consensus mechanism, which is designed for IoT-blockchain applications. G-PBFT uses the geographic locations of IoT nodes to ensure that such nodes are not malicious. The locations of the IoT devices should be fixed for this consensus to be secure, making it unsuitable for dynamic and mobile IoT networks.

PoEWAL consensus mechanism

The lightweight probabilistic proof of elapsed work and luck (PoEWAL) consensus mechanism was proposed by [74] for blockchain implementation into IoT applications.

It is time synchronous mechanism that is similar to PoW in solving a cryptographic puzzle. It uses a located time slot for each miner to solve the puzzle. Once the time slot is ended each miner broadcast its finding (the hash) and the miner with the lowest hash value will be allowed to sign and propagate the next block. This consensus is time consuming with high energy consumption foot print and adds no value to IoT applications when comparing it with PoW.

Proof of X-repute

Proof of X-repute is another consensus mechanism proposed by [75] aims at IoT blockchain applications. It implements the concept of rewards and punishments to settle the values of the node's repute. The node that behaves according to the network rules receives repute rewards. The higher the node's repute value, the lower the difficulty at which it can mine blocks. In this consensus, as time progress a few nodes will get richer in terms of repute scores and might collide together to cause malicious activities (i.e. Collusion of rich stakeholders similar to the PoS), and also could result in a more centralised network where it can be controlled by only a few nodes.

Proof of authentication (PoAh)

The IoT-friendly blockchain [76] introduces the proof of authentication (PoAh) concept for the implementation of a light weight blockchain in the IoT. The concept is based on first miner mines and validates the block. Then an evaluation of the hash value is conducted by trusted nodes on the network. Every trusted node has a trusted value. If a node correctly evaluates the block and its hash, its trust value increases by one and if the evaluation was not correct its value decreases and becomes a normal node. PoAh connectivity and block processing illustrated by Fig.2.9 adapted from [76]. While the proposed mechanism introduced an extra layer of security via an added round using PoAh, it is similar to PoA in the way that the system depends on a few trusted nodes, which makes it vulnerable to DoS attacks.

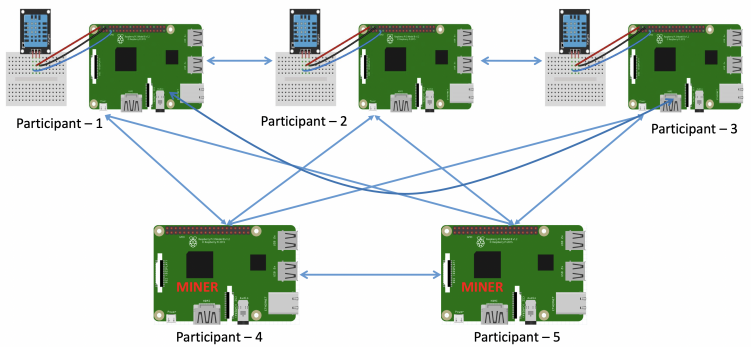


Figure 2.9: PoAh connectivity, figure taken from [76]

Table 2.1: Comparison between main general use consensus algorithms and IoT-centric consensus algorithms.

Consensus Algorithm	Numerical Complexity	Forking	Nodes' scalability	Vulnerability	Access	Transaction confirmation delay
General use consensus algorithms						
PoW	High	Yes	High	51% attack	permission-less	High
PoA	Low	Yes but dealt with efficiently	Low	Faulty nodes >(total node-1/3) DoS attacks Heavily depends on validators honesty	Permissioned	Low
PoS	Low	Yes	High	51% attack Collusion of rich stakeholders Nothing at stake attack.	Permission-less and Permissioned	High
Proof of Activity	High	No	High	51% attack Collusion of rich stakeholders	Permission-less and Permissioned	High
PBFT	High	Yes	Low	Faulty nodes >(total node-1/3) DoS attacks Heavily depends on validators honesty	Permissioned	Low
DPoS	Low	Yes	High	51% attack Collusion of rich stakeholders The rich may get richer	Permission-less and Permissioned	High
PoB	Low	Yes	High	51% attack Could be very expensive The rich may get richer	Permission-less	Moderate
Casper	Low	Yes	High	51% attack The rich may get richer	Permission-less and Permissioned	High
PAXOS	Low	Yes	Low	Faulty nodes (need at least $N/2-1$ of N acceptors) Very complicated to implement	Permissioned	High
IoT-centric consensus algorithms						
Credit-based PoW	High	Yes	Not clear	51% attack DoS attacks Centralisation issues Collusion of rich stakeholders	Not given	Not clear
Proof of Trust	High	Yes	High	51% attack DoS attacks Centralisation issues Collusion of rich stakeholders	Permission-less	High
PoAh	High	Yes	Low	DoS attacks Centralisation issues Heavily depends on few nodes	Permissioned	High
PoBT	Low	Yes	Low	51% attack Centralisation issues DoS attacks	Permissioned	Low
Microchain	Low	Yes	Low	DoS attacks Heavily depends on few nodes Collusion of rich stakeholders	Permission-less	High
PoEWAL	High	Yes	High	51% attack Very high power consumption DoS attacks	Permission-less	High

2.4 Smart contracts

In the mid-90s the computer scientist and cryptographer Szabo coined the term smart contract[77]. According to [77] the smart contract is defined as "a computerised transaction protocol that executes the terms of a contract". In blockchain a smart contract is a code stored on the network and can be triggered and executed by transactions on each node [78]. A smart contract is a form of decentralised application that runs on a virtual machine and has its own unique address on the network.

On blockchain the smart contract can be defined as executable protocol that is used to digitally facilitate, verify, and enforce the terms of a computerised contract between two or more entities or nodes on the blockchain platform. Since these contracts are executed on a platform that employ blockchain technology then they have the following characteristics [79]:

- The code of the contract or the program is tamper resistat because it will be recorded and verified on the blockchain platform and no party can change it or delete it from the network.
- The terms of the contract and their execution is achieved between anonymous and trust-less entities or nodes without any trusted centralised third party.
- The contract based on its terms and predefined conditions can have a transferable assets such as coins or tokens that can be transfer once these terms or conditions met.

2.5 Blockchain integration into IoT systems

The blockchain technology with its unique characteristics such as transparency, immutability, resilience and redundancy, and the possibility of deploying distributed applications such as smart contracts, offers great benefits to the Internet of things. These benefits can help in addressing some of the security issues on IoT, for example can help in providing secure access control, ensuring data integrity, and providing data and users privacy. It also can help in allowing for better management and connectivity for the IoT devices. In this section a survey of the related work on the integration of blockchain into IoT will be provided.

2.5.1 Integration for security purposes

The integration of blockchain and IoT into one system that is fault tolerant, equipped with important underlying security technologies such as hash function and public key cryptography, and with a consensus mechanisms that can ensure the validation and security of data. This solution can provide a resilient platform for improving IoT security in different aspects.

In terms of ***access control***, the authors in [80, 81, 82] introduce blockchain-IoT architecture for blockchain implementation within IoT application, namely smart homes for access control purposes. The architecture relies on a central entity which is a local home miner, to mine blocks, and implement the access control policy. It has a policy header that allows users to control the access to their data. This miner controls the issuing and distribution of encryption keys that allow devices to communicate with each other based on the predefined access policy. Data can be stored locally or on the cloud and could be shared with device manufacturers for performance enhancement and maintenance. This architecture ensures the confidentiality of data through predefined access control policy. However, the introduction of the centralised miner introduces the risk of single point of failure and makes the architecture more of a centralised one, which is against the concept of the decentralised blockchain.

The IoTChain architecture proposed by [83] combines Object Security Architecture for the Internet of Things (OSCAR) and the Authentication and Authorisation for Constrained Environments (ACE) framework to provide end-to-end solution for secure access control to IoT resources. For validation purposes they implemented this architecture on the Ethereum test net using smart contract. While this architecture provides owners with the ability to control access to their resources, it is using a key server that generate resource's decryption key. This does not help in decentralising IoT systems and does not achieve autonomy.

The work by [84] proposed the use of blockchain and smart control to facilitate access control within IoT systems. It is based on a tokenisation approach, where a user uses a cryptocurrency (i.e., tokens) to buy access rights to a digital or a physical asset. Using the same tokenisation approach Enigma [85] and [86] provided solutions for users to access encrypted data on the blockchain using policies provided in smart contracts.

Using blockchain and smart contract [87] and [88] proposed access control solutions for IoT applications. In these solutions the access policies are provided in smart contracts and can be used by users to grant or revoke access to their data. The authors of [89] proposed an access control solution for IoT based on blockchain and off-chain Decentralised Hash Tables (DHT). The access policies are stored on the blockchain while the data stored on the off-chain DHT. Once a request is made the DHT nodes will query the blockchain to ensure the user requesting to access the data has the right privilege to do so.

Blockchain is an ideal technology for providing a decentralised and robust authentication mechanisms. For providing attribute-based authentication within IoT systems the authors of [90] proposed a solution based on the hyperledger blockchain platform. In this solution the user's authentication method is implemented by an entity called access code, while the device code implements the query method for the data generated by devices. The administrator's access policy is provided by an entity called policy code.

The immutable nature of blockchain makes it an ideal technology to ensure ***the integrity of data*** within IoT systems, several authors have studied and used blockchain

for that purpose. The work by [91] proposed a framework for ensuring data integrity within the IoT system through the integration of blockchain. The framework consists of data owners application (DOA), data consumer applications (DCAs), cloud storage service and blockchain. Smart contract was used to implement the data integrity service over the blockchain. Users can interact with the deployed smart contract to access the pre-recorded data on the blockchain. The authors of [92] have proposed the consortium blockchain-based public integrity verification system (CBPIV) for integrity verification in cloud storage for IoT applications. The system is used to track and record the data auditor on the blockchain to ensure that all verifications are correct.

The work by [93] proposed a blockchain-based architecture for ensuring the data integrity collected by smart devices within a smart city. Data collected by devices is encrypted and validated by managers nodes on the blockchain network then it will be added to the blockchain. For end-to-end data integrity in smart cities the authors of [94] proposed a framework based on secret sharing, fog computing and blockchain. Secret sharing was used at the first layer to protect data collected by the IoT devices, fog computing was used to conceal the collected data and sending it to the blockchain, finally the immutable nature of blockchain was used to store the collected data.

To achieve traceability and anonymous and to prevent unauthorised data sharing for vehicle-to-vehicle (V2V) the authors of [95] used blockchain technology in their study for that purpose. For the purpose of threat intelligence integrity audit within the IIoT Zhang et al. [96] have proposed blockchain based solution that utilises double chain structure for this purpose. The solution implements the Paillier homomorphic encryption for achieving confidentiality during the sharing of the threat intelligence sensitive information.

The works by [97] proposed a framework based on blockchain technologies for securing the integrity of the stored data within smart city applications. For the same purpose the authors of [98] proposed an architecture for real-time context data integrity within IoT systems. It was based on blockchain, edge computing and multiple data storages, where blockchain was utilised to facilitate interconnectivity between blockchain networks and edge computing.

Blockchain technology adapt the approach of transparency, this means it does not provide data ***confidentiality*** by itself. However the implemented public key cryptography can be utilised for the purpose of data encryption. For example a confidentiality and anonymity solution based on blockchain technology was proposed by [99] for energy trading. It makes use of the public key encryption deployed on the blockchain for transactions encryption using the receiver's public key. For hiding its identity any user will generate a new pair of addresses based on both the public and private keys every time it is involved in an energy trading. Nevertheless, the proposed solution makes use of the hybrid topology of p-2-p network and hierarchical centralisation through the Distribution System Operators (DSOs) to provide authorisation of energy owners.

2.5.2 Integration for privacy purposes

The immutable characteristics provided by blockchain lay down the fundamentals for data transfer securely without the need for a trusted centralised authority. Thus providing a viable solution for IoT systems to share data and if needed execute electronic payments in a secure environment. Securing the data and ensuring its privacy is a challenge in the IoT and different researchers try to address this challenge using blockchain. The authors of [100] proposed a data management system called Enigma for peer to peer networks. Enigma is intended to ensure shared data are securely stored, always available and preserve its ownership. It uses a modified distributed hashtable for storing data. It relies on a permissionless blockchain network to enforce the access control policies and provide track and trace immutable logs. The authors claim that Enigma can be used within IoT to store, manage and use data collected. Nevertheless, the computational overhead of such a system need to be investigated against the ability of IoT devices.

The work by [101] proposed a blockchain architecture to preserve data privacy using Ethereum platform and smart contract. The idea of this work is to introduce blockchain gateways to interact with devices on behalf of users. They propose two type of smart contract to be stored on the Ethereum; one for a gateway and another for the devices. First, a device registers its ID, name, serial number and a set of policies on the Ethereum network. Then users (in order to access devices) query the gateway's smart contract, which in return provides the user with the list of register devices and their policies. Based on the policy obtained from the smart contract, users can accept and proceed or reject the policy, if they accept the policy the gateway smart contract will store that preference for future access.

The works by [102] showed that by using blockchain, pseudo anonymity can be achieved amongst IoT users and nodes when communicating and sharing data. In a similar fashion by using blockchain the FairAccess authorisation framework provided pseudonymity in IoT applications [103]. The work by [104] provided a privacy preserving blockchain platform. It utilises the attribute based encryption mechanism to ensure the privacy and the security of the data generated by the sensor layer. The platform deployed cluster heads who are responsible for sensors' data collection, processing, and encryption. The encrypted data are propagated to the network in transactions. These transactions only can be seen and validated by certain miners who have the right attributes to do that.

The Privacy Incorporated and SeCurity Enhanced Systems framework (PISCES) for providing privacy by design into IoT systems was developed by [105]. The framework provides the data provider with the ability to manage and define the privacy settings of their data. It employs Controller Smart Data System (CSDS) of the Smart Data System (SDS) to control the data based on the provider's defined settings. The work by [33] proposed a framework for integrating blockchain with IoT devices to provide a secure means to communicate in smart cities. The framework consists of four layers: application, distributed ledger, communication, and physical. This framework,

according to their authors' claims, has resilience against attacks that aim to compromise security and privacy. This is only a theoretical introduction to the framework, without a clear validation provided (such as practical or simulated implementation performance analyses).

The authors of [106] introduced a blockchain-based and decentralised personal data management system. It deploys a protocol that turns blockchain technology into an automated access control manager. The system eliminates the need for trusted third parties and ensures that users have total control over their own data. Another solution that ensures users have control over their personal data are proposed by [107]. The solution combines both blockchain and peer to peer storage networks for personal data privacy purposes. The works by [108] proposed a privacy preserving model for IoT healthcare applications. It is based on blockchain technology combined with a lightweight ring signature scheme for providing privacy and provides anatomy for the authenticated users on the network. Within energy trading and sharing both [109] and [110] have used smart contact and blockchain technology to enable users' privacy and to allow for better decisions regarding energy tariffs.

2.5.3 Integration for management

Another important benefits that blockchain can offer the IoT is the ability to provide a platform to mange and control IoT devices and manage the identities of these devices and the users on the IoT network as well.

A solution based on blockchain is proposed by Filament [111] for the purpose of controlling devices within IoT system. It is based on the TeleHash protocol by where any system, such as lights within a city, can be control over wireless networks. Smart contracts and a blockchain provide the base platform for Filament, aiming to enable smart objects such as sensors to communicate with each other and exchange data on distributed and autonomously fashion.

In terms of device software and firmware update within the IoT, the work by [112] proposed a blockchain based structure that ensure device's firmware is up-to date. The devices communicate with any node in the blockchain network and can request the latest update to its firmware, or if it is the latest version the node can check the validity of the firmware. The intended purpose of this work is to ensure device's firmware is up to date and to protect against possible firmware attacks such as zero-day attacks. This structure is an effective way of ensuring the security of device's firmware, on the other hand it depends on the honesty of the nodes on the blockchain networks. The exposure of the firmware on such transparent networks could potentially expose further zero-day vulnerabilities. Such firmware distribution processes should be encrypted to deter any nodes with bad intent.

Another solution based on blockchain for the IoT is the one provided by [113]. The authors in this work provided a proof of concept on how to use Ethereum blockchain to manage IoT devices through the implementation of smart contracts. The system allows the control of devices based on policy stored on the smart contract. As a concept this

is a good example of showing the benefits of using smart contract and blockchain to control IoT devices, however the implementation was limited to the use of raspberry pi and further inclusion of actual light bulb and/or air conditioner will provide a better outcome. The work by [114] leverages the blockchain technology to provide scalable access management in IoT. The proposed system consists of: wireless sensor networks to enforce encryption connections, managers for managing the access control policy for a set of devices, agent node responsible for deploying smart contract on the network, smart contract to govern the access management system, management hubs to act as an interface between the IoT devices and the blockchain network and translate messages from CoAP to JSON-RPC, and finally a private blockchain network. This system provides secure access control for IoT devices through smart contract and blockchain, nevertheless the managers and management hubs could be integrated into one entity for more simplicity.

The work in [115] proposes CitySense which leveraged blockchain technology to solve the problem surrounding the sensors' data storage and management within smart cities. Moreover, for software development the authors apply the adaptive and iterative SCRUM methodology. This is a proposal that relies on a central collection endpoint, which is against the decentralised concept of blockchain. The authors of [116] proposed a blockchain based solution using a multi agent for a decentralised quality of service (QoS) measurement for better IoT-based services. It is based on the idea of ensuring data reliability in real time by relying on blockchain immutability.

Based on blockchain and with mobility support a hierarchical trust management protocol was proposed by [117] for distributed IoT systems. In the protocols the mobile smart devices will share information regarding trust in the services providers on the blockchain, allowing for transparent and fast trust evaluation process of these providers.

2.5.4 General framework and architectures of blockchain and IoT

Blockchain Platform for Industrial IoT (BPIIoT) proposed by [118] is a platform based on Ethereum blockchain that consists of single board computer, a connectivity to cloud and blockchain network and an interface to control sensors and actuators and collect the reading data. Figure. 2.10 shows an overview of the proposed platform. The main aim of this platform is to facilitate the decentralised communication and dealings among machines themselves or a communication between machine and human. This provides the ability in the industrial setup to monitor the health status of machines, automate the diagnostics process and ensure the availability of a secure and shared distributed ledger for transactions records. This platform is based on permissioned blockchain to offer trust, to ensure the safety of machines and the security of transactions.

A similar work to the above is [119], which proposes a light-weight blockchain based platform for the IIoT. This work, unlike the previous one, instead of using cloud solution, propose the use of the concept of on-chain and off-chain design. Transaction

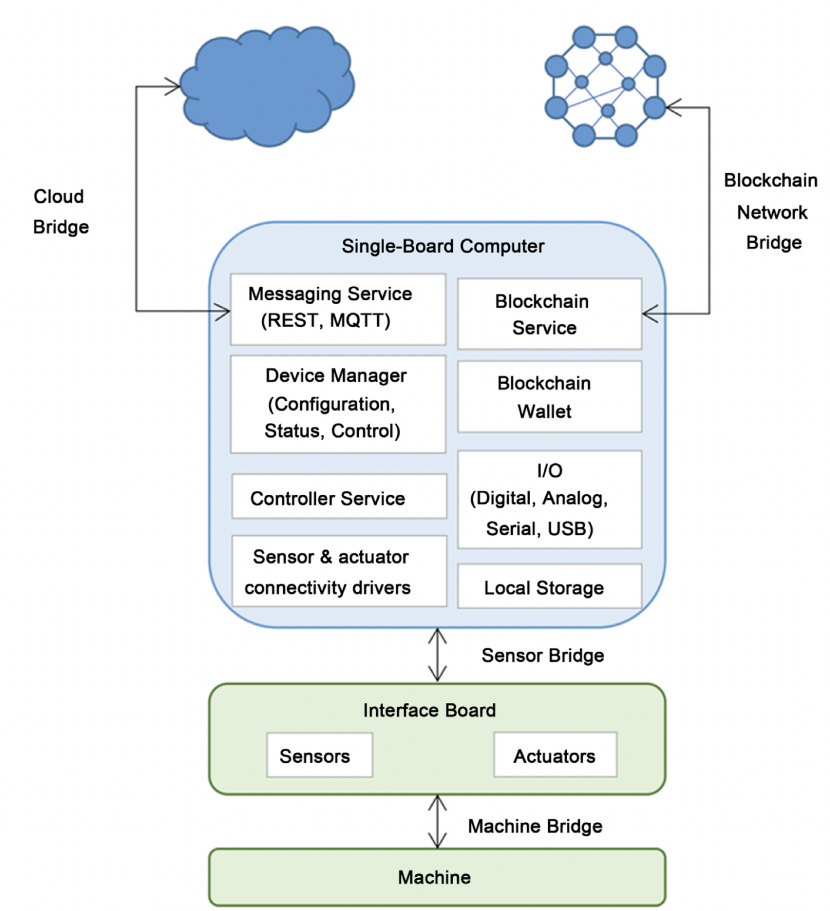


Figure 2.10: Blockchain platform for industrial internet of things (BPIIoT), figure taken from [118]

issuing and processing happens on the on-chain, which consists of normal nodes and verifications nodes. The normal nodes only issue and sign transactions while the verifications nodes are computationally powerful that are used for PoW calculation. The off-chain concept is used to store and process data through the use of distributed hash table. It is not clear how this implementation will mitigate traditional 51% attacks that could harm such network, especially with the calculation of PoW only done by few powerful nodes, making it easy to mount such attack.

IBM in 2015 [120] introduces autonomous decentralised Peer-to-Peer Telemetry system (ADEPT) based on blockchain and smart contracts. The idea of the system is to use the TeleHash protocol for messaging, Bit Torrent for distributed file sharing, and Ethereum platform and smart contract for autonomous devices coordination. The main aim is to use blockchain platforms and smart contract to overcome some of the traditional IoT networks issues, such as the trusts in centralised entity, the single point of failure, and the data and user's privacy. ADEPT is still a proof of concept and its ability to operate in an IoT contexts is still not fully tested.

2.5.5 Blockchain at the edge

The integration of blockchain and AI into the edge layer has attracted considerable attention from researchers in recent years. The work by [121] proposed a distributed cloud architecture model based on blockchain with a software defined networking controller at the edge. It consists of three layers: device layer for monitoring and data collection, the fog layer equipped with software defined network (SDN) controller receives raw data from devices associated to its own area, and is responsible for processing these data and provides services as required. Finally, the cloud layer receives all processed data from fog layer. Blockchain is integrated into both fog and cloud layers. This is a great example of how blockchain can be integrated with other technologies to provide high performance IoT system.

The framework proposed by [122] is an excellent example of how blockchain and edge layers can be used to secure IoT applications. It is introduced for vehicular communication systems by hosting security managers and blockchain, both of which are utilised to provide key transfer and management at the edge layer. Similarly, the authors of [123] proposed a new control system. It uses the hyperledger fabric blockchain, along with a smart contract, in a micro-service architecture at the edge layer to secure and validate data initiated at the lower layer. Another edge-based framework called EdgeChain was proposed by[124]. Similar to [123], it uses blockchain and smart contracts at the edge, so that devices in the lower layer can access resources at edge servers.

The authors of [125] proposed SURVIVOR a blockchain based framework in a software-defined networking (SDN) architecture to provide a secure platform for energy trading between vehicle-to-grid (V2G) for charging of electric vehicles (EVs). Another work by [126] proposed a framework called BEST based on blockchain and SDN technologies for energy trading and charging of electric vehicles (EVs) in secure and safe environment. While both frameworks provide good solutions to energy trading, they are still just proposals that need real-world implementation and validation. The work by [127] is based on using blockchain and SDN technologies to build an architecture of two parts that combined both features of centralisation and distribution for smart cities implementation. This is another good example of blockchain based solution for smart cities that need real world implementation and validation.

A Lightweight privacy protection scheme based on blockchain proposed by [128] for the surveillance cameras at the edge layer. It allows the video surveillance systems performing surveillance while preserving the privacy of any individual captured in any video. The captured videos are transferred into the blockchain network that offers different capabilities such as ensuring integrity, feature sharing, and blurring keys management. At the edge layer a privacy policy is enforced in real time on the captured videos. The authors of [129] proposed a cloud-edge collaborative blockchain as a service paradigm. The aim of this proposal is to extend service provided by blockchain to the organisation's on-premises edge or private cloud, thus, realising the high availability of blockchain systems in a more edge autonomy environment.

Based on blockchain and mobile edge computing the authors of [130] introduced a security architecture for Vehicular Ad-hoc NETwork (VANET). The architecture consists of a perception layer that utilises the blockchain capabilities to ensure the security of the transmitted, an edge layer for processing, and a service layer where blockchain and cloud are utilised for securely storing data. However it is not clear how blockchain can be used to ensure the confidentiality of the data in transmission.

Acce-chain system that utilises different edge storage capacities for blockchain elastic storage proposed by [131]. The system is aimed towards latency sensitive vehicular for providing secure and efficient data access in order to accelerate the data exchange at the edge. While the platform does provide an immutable storage service for vehicles, its benefits in terms of latency sensitivity and improvement is questionable due to the nature of the consensus process on the blockchain platform which usually takes some time to validate the data.

2.6 Blockchain, AI and IoT integrations

The integration of blockchain and AI into IoT systems could unlock many advantages for users and organisations. In the following subsections the state of the art related to the integration of blockchain, AI, and IoT systems including the edge computing will be discussed.

2.6.1 Blockchain and AI at the edge

The integration of blockchain into the edge layer has attracted considerable attention from researchers in recent years. The authors of [132] proposed a platform named NeuRoNt based on the Ethereum blockchain and an edge layer hosted a smart contract. The platform consists of multiple agents powered by smart contracts that can solve complex problems. Ethereum and smart contract-based mobile edge sharing systems were proposed by [133]. AI used for data processing, and blockchain and cloud platforms were used to facilitate the sharing of services in IoT-enabled smart cities.

The authors of [134] also proposed an architecture for data analysis at the edge based on blockchain and AI. The aim is to enhance the security of privacy-critical systems, such as healthcare applications, by restricting raw data to producers only. The authors of [135] proposed a blockchain-based edge intelligence (EI) system for improved data security, privacy, and performance. It uses a public blockchain to ensure the communication security of consumer electronic devices (CEDs) and a private blockchain to ensure communication security among EI servers.

The authors of [136] introduced the edge and intelligence (Edgence) that utilises the edge layer to access the IoT devices, and then use the blockchain platform at the edge to realise the self governing services of the edge cloud to support the IoT dApps. The AI-chain system was proposed by [137] based on the edge intelligence and blockchain for learning results sharing intelligence among edge nodes in beyond fifth generation

(B5G) networks. The authors also claim that the system can be used to solve the resource allocation problem within these networks for better effectiveness. The system was only validated through simulation.

The works by [138] based on consortium blockchain, smart contact, and an AI at the edge layer proposed an efficient and secure platform for knowledge management and trading. The purpose is to allow the trading of knowledge between nodes in a peer-to-peer implementation. Another system based on blockchain, AI, and edge called (BEMA) proposed by [139]. It allows participants to locally host a heterogeneous learning models and share it with each other. At the same time it allows nodes to receive each other's models and utilise it and process it locally.

In [140] the BlockDeepNet framework was proposed, which combined the implementation of deep learning, blockchain, edge computing, and smart contracts for data analyses in IoT. Blockchain was used to securely exchange local and global updates of the deep learning model. BlockDeepNet general overview of the reconfigurable IoT network for deep learning is shown in Fig.2.11 along with the location of blockchain clients at each layer [140].

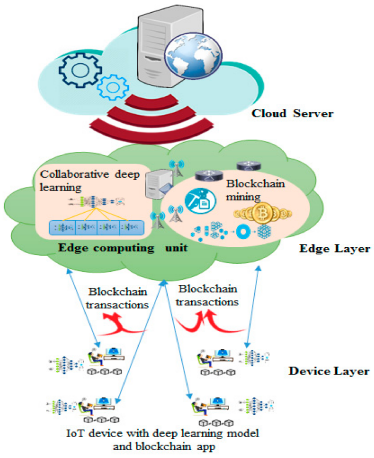


Figure 2.11: BlockDeepNet reconfigurable IoT network for deep learning, figure from [140]

2.6.2 General implementation of blockchain and AI within IoT

The work by [141] using blockchain and machine learning introduced a prediction framework called ModelChain. This allows multiple healthcare institutions to train the same framework for better results in terms of health prediction. The work by [142] proposed the DeepCoin framework for smart grids based on blockchain and deep learning. The deep learning used is an intrusion detection systems (IDS) scheme for detecting fraudulent transactions and attacks in the blockchain-based network. Another framework proposed by [143] is based on deep learning, SDN, and blockchain

for enabling high-performance and cost-effective computing resources for smart city applications. Nevertheless, both [142] and [143] frameworks suffer from centralisation issues.

The authors of [144] provided a practical integration that combines federated learning (FL) and blockchain with the aim of securing big data and preserving privacy within IoT systems. It achieves this by using fuzzy hashing to detect suspicious activities, such as poisoning attacks in FL-trained models.

The work by [19] proposed a FL system for helping manufacturers develop smart home systems. It uses consumer's data for training an ML model to assess home appliance manufacturers. Blockchain is used to ensure accountability within the system, especially when a model performs an update operation. The authors of [145] proposed BAFFLE, a blockchain-based FL environment that leverages smart contracts for coordinating the model aggregation.

The work by [146] introduced a scheme for batch authentication in the internet of vehicles (IoV) based on blockchain and AI. The aim is to address the security challenges that result from the communication between different entities within IoV-based smart cities. The scheme provides the IoVs with the secure ability to authenticate themselves when two vehicles are communicating and for a group of vehicles to be authenticated by the roadside unit. The authors of [147] proposed a pandemic situation supervision scheme based on blockchain and AI. It utilises an AI-equipped swarm of drones to monitor an outbreak in the case of a viral pandemic. This scheme was designed to help control the spread of viruses by ensuring that people follow the guidelines and performing surveillance checks (e.g., face coverings, temperature measurements, and social distancing). Similarly, the work by [148] proposed the use of blockchain technology along with unmanned aerial vehicles (UAVs) for patient data collection within healthcare. It uses UAVs to collect data and a blockchain to store the collected data. It uses tokens and shared keys to establish secure communication with users' body sensors.

2.7 Summary

In this chapter the working principle of blockchain were discussed in detailed along with different general purpose consensus mechanisms. The related work was studied and analysed in order to find the research gaps.

Many authors provided different blockchain-based solutions to issues within the IoT systems. However, none of these studied provided a comprehensive study of a real-world blockchain-based IoT applications. These studies failed to provide a practical implementation with field tests in order to evaluate the performance of any blockchain-based IoT applications. No study provided a complete measurement of transaction arrival time and end-to-end system latency, energy consumption, the impact of the block propagation delay on the network stability and nodes synchronisation, along with security and risk assessment.

Based on the study and analysis of the related work, there is a gap in term of designing a consensus mechanisms that is permission-less and non financial suitable for implementation within a public IoT-blockchain networks. Many authors proposed different consensus protocols suitable for the IoT systems, such as [69, 71, 149], however these protocols fall short of providing an IoT-centric permission-less protocol. This is because either they suffer from centralisation issues or difficulty adjustment issues, are based on financial applications, or they are for permissioned access only implementations.

Due to the distributed nature of the IoT, where thousands of smart devices are available and can communicate with each other, it can offer a scalable hardware and software platform[18]. This platform can offer a great advantages for the implementation of the DAI systems, realising the benefits of true parallelism and distribution offered by AI in a fully distributed computing system. Many authors explored the integration of AI and IoT into one system. However, none explored the blockchain as a means of supporting the implementation of the DAI over the distributed IoT hardware. This is a gap worth time and effort to explore and investigate.

Another advantage of the distributed nature of the IoT with the presence of the edge computing and AI is the possibility of utilising it to provide an attractive ability in the form of processing data in a distribution approach and in near real-time to reduce the communication overhead needed to transfer data to a centralised entity such as the cloud [19]. Many of the literature such as [132, 133, 134, 135, 136, 137, 150] integrated the edge, AI, and blockchain into one system for different purposes. However none of these have combined these systems along with the IoT in an architecture that ensures the integrity of the data from their submission by the end devices until an AI engine processes these data and provide insightful outcome while providing transparency and full traceability.

By taking advantage of the edge devices computation abilities, the AI engine intelligence, the vast amounts of useful information provided by IoT, the immutability and decentralisation capabilities of the blockchain, a system that is secure and able to monitor the environment and collect data, learn, analyse data based on the requirements of the executed task, and produce actionable outcome can be created.

Chapter 3

Latency and Performance Analyses of Real-World Wireless IoT-Blockchain Application

In this chapter, a practical incorporation of blockchain into the Internet of Things is demonstrated using Ethereum Proof of Authority (PoA). It provides performance analyses, which include measurement of the transaction arrival time, the system end-to-end latency for different network implementations over cellular and Wi-Fi, and the average power consumption. This includes the study of the effect of network bandwidth on the stability and synchronisation of all nodes on the blockchain network.

3.1 Introduction

The Internet of Things (IoT) is increasingly being utilised, by both businesses and individuals, for many applications. This utilisation means increases in the smart devices that are connected to the IoT, which will significantly increase the challenges related to devices' interconnectivity and management, data and user privacy, and network, data, and device security. At the same time, blockchain approaches provide a decentralised, immutable, and peer-to-peer ledger technology that could be the right answer to these challenges. Significant challenges, however, accompany the integration of blockchain into the IoT, since IoT smart devices may suffer from resource and power constraints and blockchain is associated with scalability and delay issues.

3.1.1 Problem statement

Blockchain and IoT are potentially an ideal fit, where blockchain can offer a solution to the challenges within IoT, such as data integrity, device authentication and authorisations, and system availability. Immense effort, however, is required to integrate the two technologies. This is because IoT devices may be limited in power and storage;

they also produce vast amounts of raw data that need to be processed in a suitable environment. At the same time, blockchain still suffers from some issues, such as scalability.

According to the authors of [22], who provide a comprehensive systematic literature review and analysis of blockchain solutions for IoT, most studies have not measured the complete transaction time from submission until the transaction is committed in the blockchain network. The authors of [22] also state that, for better performance analyses, 'the performance of the whole proof of concept should be analysed from end to end, from the transaction being submitted until the transaction being included and committed'. In terms of performance analyses and providing complete measurement of transaction arrival time and end-to-end system latency, this information is not provided by any of the authors of the current related work. Based on this, there is a need to study and evaluate the performance of blockchain-IoT application using a real-world use case. In this work, a performance analyses of the system latency, network synchronisation and stability, and energy consumption were provided. Table 3.1 provides a comparative analysis between this work and related works.

Table 3.1: Comparative analyses between this work and the related work.

Work	C1	C2	C3	C4	C5	C6
[80, 82]	No - Proposed an architecture and uses simulation for validation	Platform like Bitcoin but without PoW (more of PoA)	Qualitative evaluation and Simulation Results	No	No	Simulated results of energy consumption of the smart home miner - which is a PC (not the end IoT device)
[70]	No - only simulation using Bitcoin simulator in NS-3	Bitcoin PoW in a sub-blockchain architecture	Yes- simulated performance analyses that includes; block sizes and block generation intervals, and evaluating the effect of varying the number of IoT devices and their locations	No	No	No
[151]	Yes - prototype (3 nodes and a smartphone)	Ethereum	No	No	No	No
[152]	Yes- Proposed Blockchain Platform for Industrial Internet ofThings (BPIIoT), and validate it with practical implementation	Ethereum	No-only evaluation without measurements	No	No	No
[115]	No	Planning to use Ethereum	No	No	No	No
[125]	No	Not clear	Simulation Results	No	No	No
[126]	No	Proposed their own consensus algorithm	Evaluation and Simulation Results	No	No	No
[127]	No	Ethereum	Evaluation and Simulation Results	No	No	No
This work	Yes- 16 IoT nodes were deployed around the city of Sheffield, UK.	Ethereum PoA – deployed private network.	Yes-measurements were provided of system latency including block propagation and importing, energy consumption, and node synchronisations and network stability	Yes, a module that predicts the latency based on the number of nodes was provided	Yes	Yes – using pragmatic IoT devices.

Notes: C1: Practical Deployment, C2: Blockchain Platform, C3: Performance Analyses, C4: End-to-End System Latency Measurements, C5: Study of Network Stability and Nodes Synchronisation, C6: Energy Consumption Measurements

3.1.2 Contribution of the chapter

The contribution in this chapter can be summarised as follows:

- Practical implementation of an IoT-blockchain application for flood monitoring and detection using Ethereum Proof of Authority (PoA) [153].
- Utilisation of Smart Contract to coordinate and automate the execution of decisions within IoT realm.
- A performance analysis is provided, which includes the measurement of the transaction arrival time and the system end-to-end latency for different IoT-blockchain network implementations over 3G cellular and Wi-Fi.
- A comprehensive study of the network stability and node synchronisation for both network implementations for different transaction submission scenarios.
- IoT device's energy consumption measurements for both implementations (over Wi-Fi and over cellular networks).

Published papers

The following papers based on this chapter were published:

1. S. M. Alrubei, E. A. Ball, J. M. Rigelsford and C. A. Willis, "Latency and Performance Analyses of Real-World Wireless IoT-Blockchain Application," in IEEE Sensors Journal, vol. 20, no. 13, pp. 7372-7383, 1 July1, 2020, doi: 10.1109/JSEN.2020.2979031.
2. S. Alrubei, J. Rigelsford, C. Willis and E. Ball, "Ethereum Blockchain for Securing the Internet of Things: Practical Implementation and Performance Evaluation," 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Oxford, UK, 2019, pp. 1-5, doi: 10.1109/CyberSecPODS.2019.8885029.

3.1.3 Organisation of the chapter

The rest of the chapter is organised as follows. Section 3.2 provides an overview of the system design and node components. Section 3.3 presents the system analysis; this is followed by security analysis of the system, which is described in section 3.4. The practical implementation is presented in section 3.5, thus followed by the details of the results in section 3.6, and finally the chapter summary was provided in section 3.7.

3.2 System design

In order to study the impact of integrating blockchain into IoT applications and be able to provide a performance analyses and complete measurement of transaction arrival time, power consumptions, and end-to-end system latency, a system for flood detection were deployed and tested. This system is comprised of up to 16 nodes where some of them are assigned the responsibility of mining and propagating new blocks. An overview of the system main components are illustrated in Fig.3.1, the red nodes are the authorised signers.

The main advantage of the system is measuring water level on a body of water such as a lake or a reservoir. In case of water reaching a level where it is highly likely will cause a flood the system then will automatically act by turning on a water pump to allow some of the water to be drained out of the water body until the water is reduced to an acceptable level.

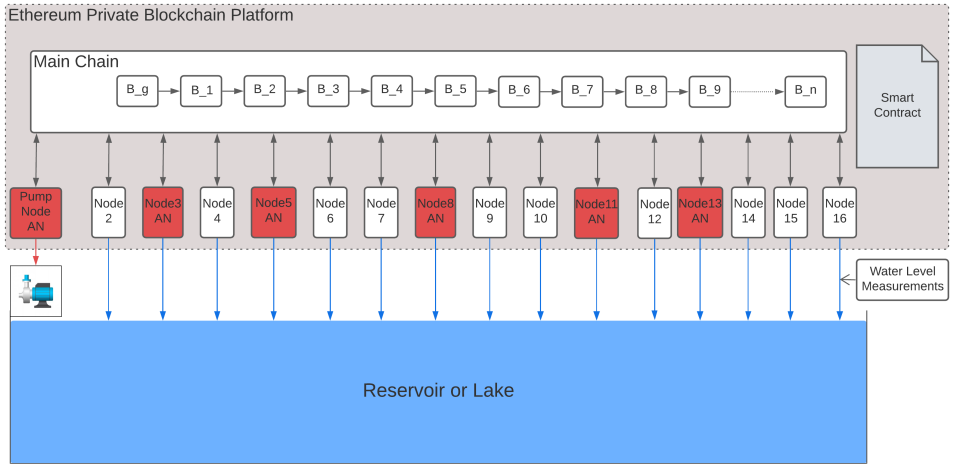


Figure 3.1: Overview of the flood detection system.

The system is based on Ethereum PoA [153] private blockchain platform and smart contract. The system consists of 16 different physical nodes. Each node comprise of the following hardware:

- Single-board computer (SBC) Raspberry Pi.
- An ultrasonic sensor.
- A cellular board in the form of Adafruit Fona 3G.
- An interface board to facilitate communication between the SBC and the sensors and the Adafruit Fona 3g. Figure.3.2 shows the physical hardware components of the node.

These nodes can be connected to each other by Wi-Fi or 3G cellular connectives. Ethereum clique client were installed and deployed on each of the nodes, and a smart contract was created and deployed on the blockchain network as well.

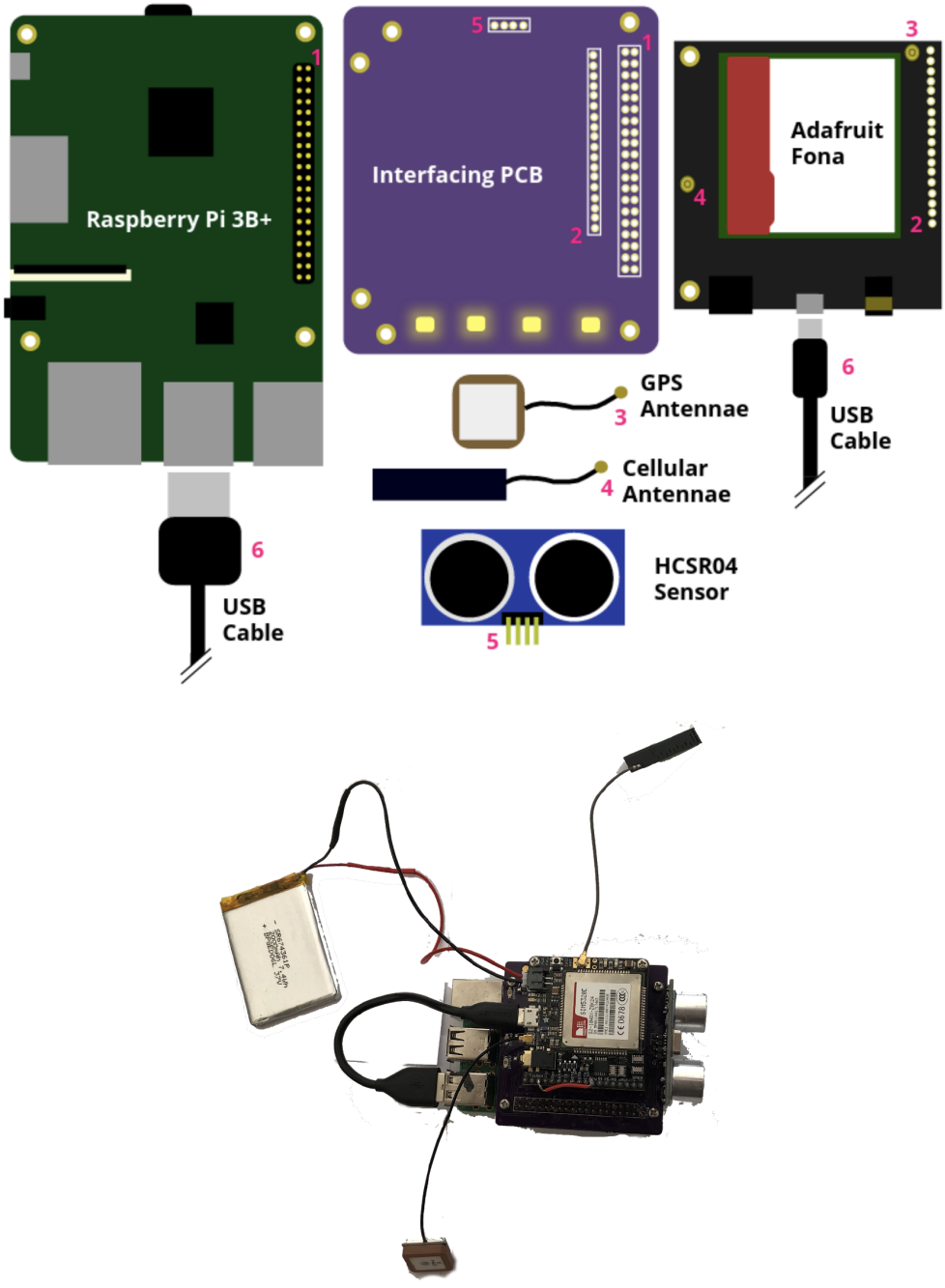


Figure 3.2: Hardware components of the node.

3.3 System analysis

The system under consideration is based on the Ethereum clique PoA [153]. This protocol allows predefined authorities (signers) to mine and propagate blocks to other nodes in the network. Once a block is received by other nodes, its transactions are immediately confirmed, resulting in a latency of 1 block, because the protocol has been built around the trust of the authorised nodes. This provides significant benefits, in terms of lowering the network latency and energy consumption, and is ideal for implementation as a client in an IoT realm. Ethereum has its own EVM, which allows for the deployment of dApps, such as smart contracts, stored on the blockchain and can be triggered and executed by transactions on each node [154], more details about Ethereum can be found in subsection 2.2.3. In the following subsections, the analysis of transaction arrival time on the Ethereum network is presented. Table 3.2 presents a list of the variables used and their definitions.

Table 3.2: Definitions of variable used.

Variables	Definition
$t_{0,1,2,3,\ldots,n}$	Times at which the miners release new blocks into the network.
$BP = t_1 - t_0, t_2 - t_1, \ldots, t_{n+1} - t_n$	Block period time (the minimum time between the release of new blocks).
transaction arrival time TAT	The time from transaction submission by a node until the transaction confirmed on the network.
Tx	Transaction
t	Time
T_m	The time during which a miner mines the block.
ΔT	The time towards the end of a block mining time; transactions that arrive during this time will not be included in the next block.
T_{pd}	The transaction propagation delay from transmission by a node until it arrives in a miner's transaction pool.
t_i	The ideal time for transaction submission during the system steady state.
S_{LP}	The period of time the sensor takes to measure the distance from the water level.
transaction gas T_g	The amount of gas the miners charge for the processing of each transaction.
block gas limit B_g	The maximum allowance of gas charges (the sum of all transactions' gas consumption).
N_v	Validators Nodes (store full copy of the blockchain and allowed to mine and propagate blocks)
N_p	Participant Nodes (store full copy of the blockchain but are not allowed to mine and propagate blocks)
ND	Number of Nodes
N_t	The total number of transactions in a miner's transaction pool.

3.3.1 System characteristics

Ethereum blockchain network with block generation based on the block period (BP) of a fixed value was considered. The system has the following characteristics:

- Multiple nodes are connected to one another in a peer-to-peer network via wireless links.

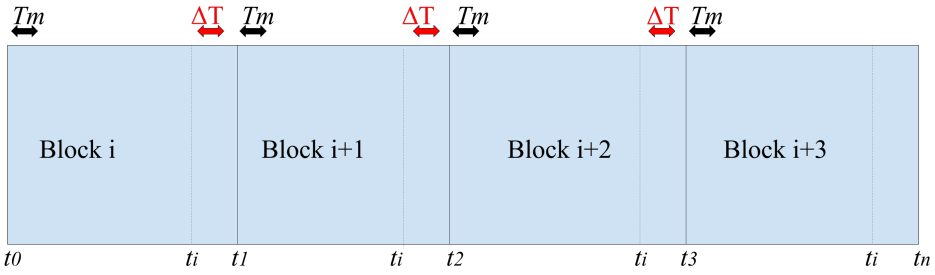


Figure 3.3: Timing of block mining and transactions submission ideal time.

- Two different processes are the main traffic generators on this network: propagation of transactions and propagation of blocks through the network to all nodes; both are broadcast transmissions.
- The case where the delay of the propagated transactions depends on the condition of the wireless network was considered. This is called transaction propagation delay (T_{pd}).
- The size of transactions is assumed to be fixed, and only the gas charged by the miner for executing the transaction influences the block size.
- Nodes on the network are full nodes, where the full copy of the blockchain is stored locally and synched with the latest block in the network.
- The mining of blocks happens right at the start of the block period, at time T_m (as shown in Fig.3.3).
- Newly arrived transactions will not be mined until the next immediate block.
- Transactions are added to the block during the period ΔT. Any transactions arriving during this period will not be considered for that block.
- The total number of transactions waiting in the transactions pool at any given time t is N_t.
- Transactions are mined in batches; the maximum batch size is equal to the maximum block size, B_{size}.
- In Ethereum, the number of transactions that can be included in a block is based on the block gas limit B_g and the amount of gas consumed by each transaction (T_g).

- The interval between blocks is the block period (BP). After transactions are added to w block and mined, a miner will wait until the end of the BP to release the block to the network.
  - For every transaction, there is one BP service time.

3.3.2 Synchronisation process

In IoT constrained devices there are two possible scenarios in terms of deploying blockchain clients. The first one is implementing a full node where a device has a full copy of the blockchain. In this protocol, devices are fully part of the network where they mine blocks, propagate blocks, send transactions, verify transactions and blocks. The second scenario is where IoT devices will act as a light node and keep track of a blockchain network and synchronise only the block headers, for example, the Ethereum light client [155]. Nodes in this scenario depends on how well they trust each other to access and check blocks and transactions.

In this work only the first scenario where devices are full nodes but could act differently in the network in terms of mining blocks will be considered, and this will result in having two types of nodes. Nodes that keep full copy of the blockchain network locally and are able to mine blocks, validate them, and initiate and verify transactions and are called validators N_v. The second types are the nodes that keep a full copy of the blockchain network locally and are able to initiate and verify transactions but not allowed to mine blocks and they are called participant nodes N_p. The length of the global chain at time t can be describe as a $L(t)$. Since the validators N_v are allowed to sign and propagate blocks then the length of the local copy is defined as $LN_v(t)$ where $LN_v(t) \geq L(t)$. On the other hand, the length of the local copy in N_p should always be $LN_p(t) \leq L(t)$. The difference in the number of blocks between N_p and the global chain at any given time t can be defined as $D(t)$, and can be calculated by the process:

$$D(t) = L(t) - LN_p(t). \tag{3.1}$$

3.3.3 Transaction arrival time during steady state ($N_t \leq B_{size}$)

The probability of transaction arrival in the network is based on the Poisson process with arrival rate λ.

$$P(T \leq t) = 1 - e^{(-\lambda t)} \tag{3.2}$$

Let λ represent the rate at which blocks are added to the blockchain network; $\lambda = 1/BP$ blocks/sec, and assuming this rate for the remainder of this analysis. The time t depends on the block period, the number of blocks (n) for which a wait before transactions arrive in the network was needed, and the propagation delay (T_{pd}). Assuming that transaction submission at (t_0) (Fig.3.3), then the probability for the transactions to arrive in the network after n blocks is as follows:

$$P(n) = \begin{cases} 1 - e^{-1/BP \times (n \times BP - T_{bp})} & , T_{bp} < (n-1) \times BP \\ 0 & , T_{bp} \geq (n-1) \times BP \end{cases} \tag{3.3}$$

This is true provided the transactions arrive before processing time ΔT , that is to say $(Tpd < (BP - \Delta T))$; otherwise, $P(n) = 0$. Knowing the probability, the transaction arrival time (TAT) can be calculated by:

$$TAT = \frac{\ln(1 - P(n))}{-1/BP} + (T_{pd} - \sigma) \tag{3.4}$$

Where σ is a variable that represents the system and smart contract processing time.

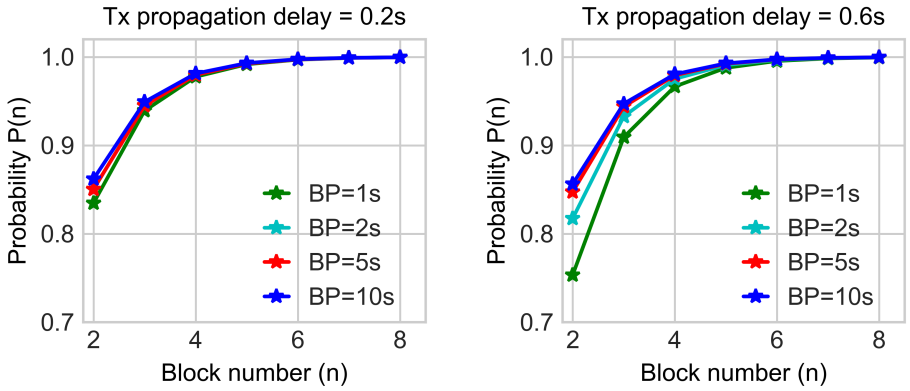


Figure 3.4: Probability of transaction arrival after n blocks.

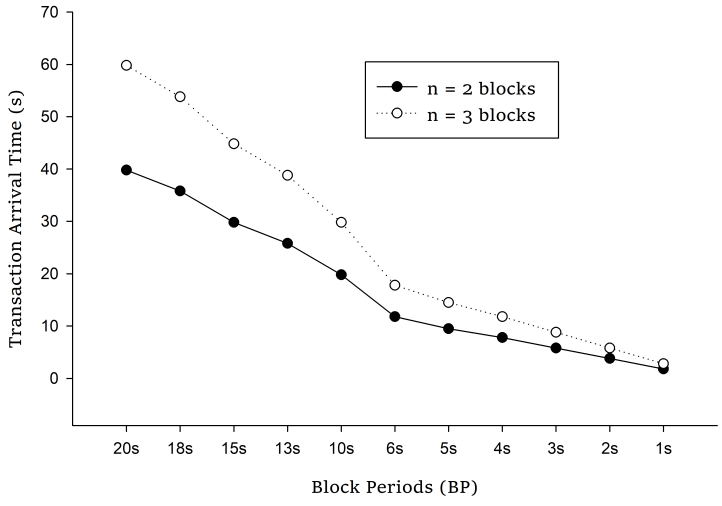


Figure 3.5: Transactions arrival time for different block periods.

Numerical analysis: assuming that $\Delta T = 0.2s$, the probability of transaction arrival in block number n for different values of T_{pd} for $BP = [1, 2, 5, 10]$ seconds can be calculated. In addition, using the values of $P(2)$ and $P(3)$ (i.e. arriving after two and three blocks) and assuming that $T_{pd} = 0.2s$, the transaction arrival time for $BP = [1, 2, 3, 4, 5, 6, 10, 13, 15, 18, 20]$ seconds can be calculated using 3.4. As can be seen in Fig.3.4 and Fig.3.5, it is clear that the blocks with shorter BPs (especially one and two seconds) are the most affected by T_{pd}; however, as the BP increases, the T_{pd} effect becomes negligible. This means that the longer block period (10s and above) should be implemented for better performance in networks with limited bandwidth.

3.3.4 Transaction arrival time during busy state ($N_t > B_{size}$)

During the busy period, where the system cannot accommodate all transactions waiting in the pool in one block, some transactions must wait in the pool for a number of block periods. The maximum waiting time in the pool can be defined as t_w. Assuming that transactions are served on a first-come-first-served basis, then T_{pd} can be neglected, and t_w can be calculated as follows:

$$t_w = \lceil (N_t \times T_g)/B_g \rceil \times BP \tag{3.5}$$

In such cases, the probability of transaction arrival after n blocks is as follows:

$$P(n) = \begin{cases} 1 - e^{\left(-1/BP \times \left(\left\lceil \frac{N_t \times T_g}{B_g} \right\rceil \times BP\right)\right)} & , \left\lceil \frac{N_t \times T_g}{B_g} \right\rceil < (n-1) \times BP \\ 0 & , \left\lceil \frac{N_t \times T_g}{B_g} \right\rceil \geq (n-1) \times BP \end{cases} \tag{3.6}$$

Knowing the probability, the transaction arrival time can again be calculated as follows:

$$TAT = \frac{\ln\,(1 - P(n))}{-1/BP} + \left(\left\lceil \frac{N_t \times T_g}{B_g} \right\rceil \times BP \right) + (T_{pd} - \sigma) \tag{3.7}$$

Numerical analysis: assuming that each node submits one transaction during the block period, $BP = 20s$ and $T_g = 21,000$. Using (3), $P(n)$ can be calculated , and the transaction arrival time also can be calculated using (3.7). Fig.3.6 and Fig.3.7 both illustrate the effect of the number of nodes ND on the probability of arrival and how increasing the block gas limit can reduce the waiting time before transaction arrival in the network. It is clear from both figures that as ND increases, the total transactions will increase, resulting in increased waiting time for transactions in the pool. This waiting time can be reduced, however, by increasing the block gas limit.

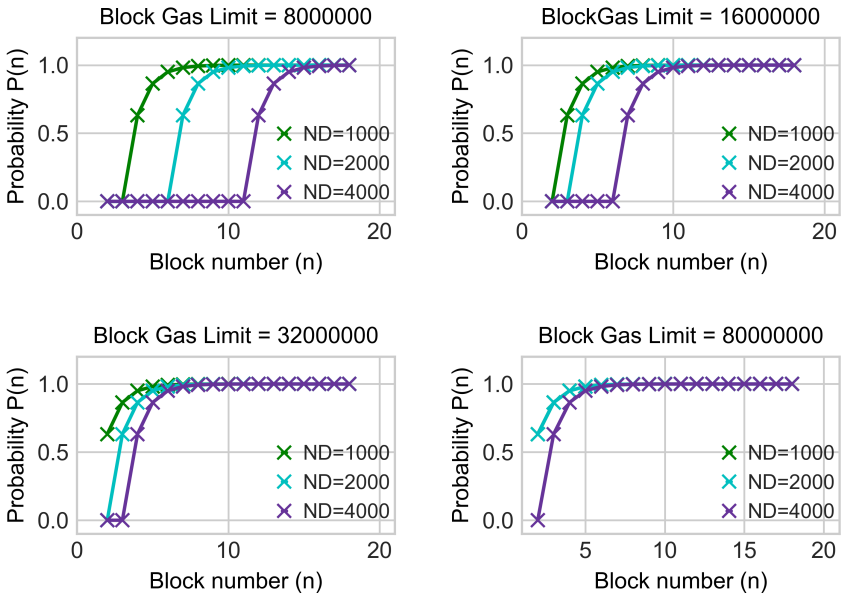


Figure 3.6: Probability of transaction arrival during busy period.

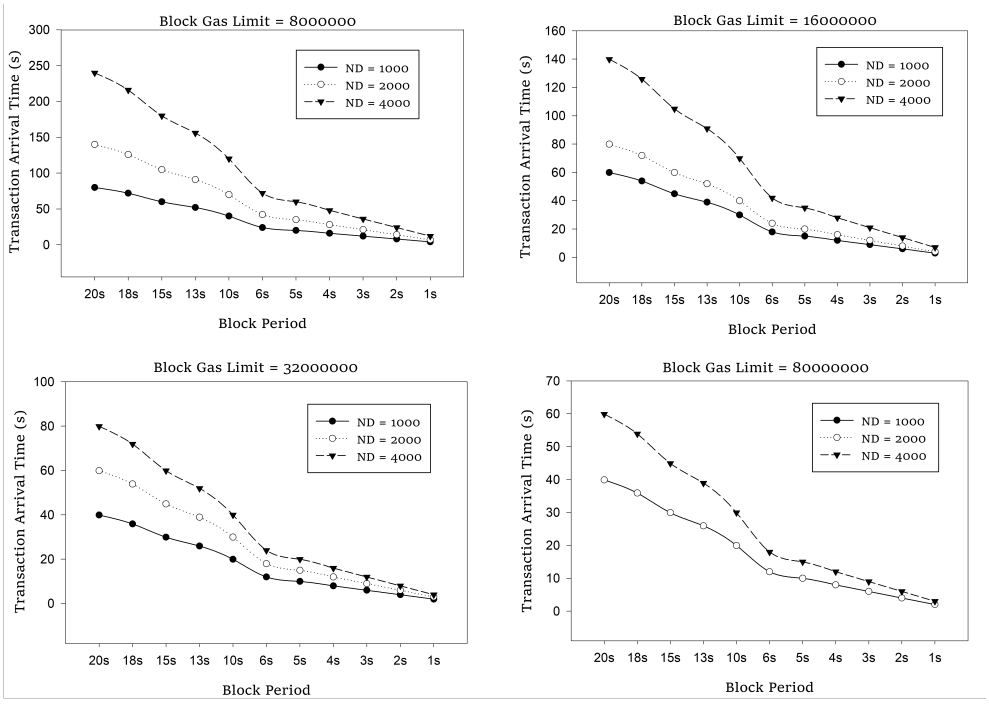


Figure 3.7: Transaction arrival - maximum time.

3.4 Security analysis

In this sections security analysis and qualitative risk assessments will be presented covering all the important attack that can cause problems to the system. The risk assessment was performed based on the NIST SP-800-30 [156] standard. NIST SP-800-30 was developed by the National Institute of Standards and Technology (NIST) for the purpose of providing guidance to organisations on conducting risk assessments.

NIST SP-800-30 provides organisations with the ability to identify: 1) threat to the information system and the organisation; 2) any internal or external vulnerabilities; 3) the possible impact if a threat exploited a vulnerability; 4) the likelihood of any impact that may occur.

The likelihood of threat occurrence is assigned a qualitative values from very high to very low, these values are explained in Table 3.3. Similarly the impact levels are assigned same values, see Table 3.4 for details. Based on the analysis of threat's likelihood and possible impacts the risk level can be determined and qualitative values can be assigned to each resulting risk from any threat. Table 3.5 provides the risk level determination based on the threat's likelihood and possible impact.

Table 3.3: NIST SP-800-30 assessment scale – Likelihood of threat event initiation by an adversary or occurrence as a result of non-adversary

Qualitative values	Description
Very High	Adversary is almost certain to initiate the threat event. Or Error, accident, or act of nature is almost certain to occur; or occurs more than 100 times a year.
High	Adversary is highly likely to initiate the threat event. Or Error, accident, or act of nature is highly likely to occur; or occurs between 10-100 times a year.
Moderate	Adversary is somewhat likely to initiate the threat event. Or Error, accident, or act of nature is somewhat likely to occur; or occurs between 1-10 times a year.
Low	Adversary is unlikely to initiate the threat event. Or Error, accident, or act of nature is unlikely to occur; or occurs less than once a year, but more than once every 10 years.
Very Low	Adversary is highly unlikely to initiate the threat event. Or Error, accident, or act of nature is highly unlikely to occur; or occurs less than once every 10 years.

In this thesis the risk level determination from NIST SP-800-30 as shown in Table. 3.5 was adapted in analysing the security aspects and performing risk related assessment to the proposed systems and protocols. The adaptation of the qualitative method over the quantitative method for assessing the risk to the proposed systems is because cost is not part of this research. According to [156] quantitative assessment is most effective in supporting cost-effective-based analyses.

Table 3.4: NIST SP-800-30 assessment scale – Impact of threat events

Qualitative values	Description
Very High	The threat event could be expected to have multiple severe or catastrophic adverse effects on the system operations, user's assets, or individuals.
High	The threat event could be expected to have severe or catastrophic adverse effects on the system operations, user's assets, or individuals.
Moderate	The threat event could be expected to have serious adverse effects on the system operations, user's assets, or individuals.
Low	The threat event could be expected to have limited adverse effects on the system operations, user's assets, or individuals.
Very Low	The threat event could be expected to have negligible adverse effects on the system operations, user's assets, or individuals.

Table 3.5: Risk level determination based on NIST SP-800-30

Likelihood	**Level of Impact**				
	Very Low	**Low**	**Moderate**	**High**	**Very High**
Very High	Very Low	Low	Moderate	High	Very High
High	Very Low	Low	Moderate	High	Very High
Moderate	Very Low	Low	Moderate	Moderate	High
Low	Very Low	Low	Low	Low	Moderate
Very Low	Very Low	Very Low	Very Low	Low	Low

3.4.1 DoS attack

PoA relies on pre-authorised set of signer nodes to validate transactions, mine new blocks, and validate new blocks. At the start of each block period (BP) one of the signers nodes (i.e authority nodes) will collect transactions, validate them, add them to new block, and sign and propagate the new block. Signers do this in a round robin process and within previously defined BP (e.g., 10 seconds). This can create a vulnerability as an attacker can target the signers in turn through DoS attack. This is because only pre-authorised signers are allowed to mine blocks, and also the attackers can easily know the BP and hence, know the time of when new blocks are released on the network. Another form of DoS is to target the node that is responsible for handling the water pump, which may prevent the operation of the pump during flood cases.

While the network operators can implement extra security measures to prevent such an attack, the likelihood of such an attack is *moderate*. If such an attacker succeeded in executing DoS attack the impact could cause sever damage to the network availability which may in turn prevent flood node from accessing the smart contract, and this requires quick and effective act on from the network operators. This makes the possible impact *very high*, thus the risk level from this attack is.

3.4.2 Malicious signer

It is possible that a malicious node can be added to the authorised signers' list, or may be a signer's private key or machine is compromised. In such a case the malicious

signer can harm the network, for example propagating an illegitimate block that may cause the pump node to activate the pump or prevent the activation of the pump in the case of flood. This may cause a sever damage to people life and/or assets. However, the protocol implements mechanisms to defend against such an attack. In PoA if the authorised signer's list contain N signers, then any signer may only mine one block out of J blocks. Also the protocol implements a mechanism that allows honest signers to vote out any malicious node from the signer list. These mechanisms are in place to limit the impact of the damage that can be caused by such an attack.

As PoA is a private protocol and relies on administrator works to add nodes to the signer list this makes the likelihood of such attack *low*, however the impact of such an attack can be damaging making the impact level *very high*, this result in risk level of *moderate*.

3.4.3 Censoring block

Another attack is when a signer or multiple signers attempt to reject adding a new block to the chain. This may be because in such a block trusted signers are voting out a malicious signer from the authorised's signer list. In order for such an attack to be successful the malicious signer needs to control at least 51% of the authorised signers, in which case the network is then totally controlled by the malicious signer.

As this is a private network controlled by a human administrator the likelihood of such an attack is *low* and its impact if successful is *high*, as administrated can step in and fix the situation, this result in risk level of *low*.

3.4.4 Vote injection

Spamming signer is an attack where a malicious signer can inject new vote proposal into each block it signs and propagates in order to manipulate the authorised signer's list. This attack is time consuming and also PoA implements a mechanism to prevent this attack by deleting votes after a window of W blocks.

The likelihood of this attack is *low* as a signer needs a longer period of time than the W window. The impact of this attack is *high* as if it succeeded, the signer's list may include malicious nodes. This makes the risk level *low*.

3.4.5 51% attack

Unlike the traditional 51% attack that is associated with the PoW based public blockchain platform such as bitcoin, the 51% attack that can target PoA is associated with controlling the majority of the authorised signers (at least 51% of them). This is a possibility but as explained above it can be a very time consuming, and there is also the private factor of the blockchain implementation making it difficult for any nodes apart from the owner of the network to control the majority of the signers.

Due to the fact that this network is private and the assumption that adequate security protection is placed to protect the network from the outsiders, although the impact of such an attack is *very high*, the likelihood of it is *very low*, making the risk level *low*.

3.5 Practical implementation

To perform the necessary measurements of the performance of Ethereum blockchain, a use case based on flood detection and control was designed and implemented. The aim was to monitor a reservoir, tanks, or a river such that, in the case of a flood, a controlling pump could be automatically activated to discharge the water and prevent the flood from occurring.

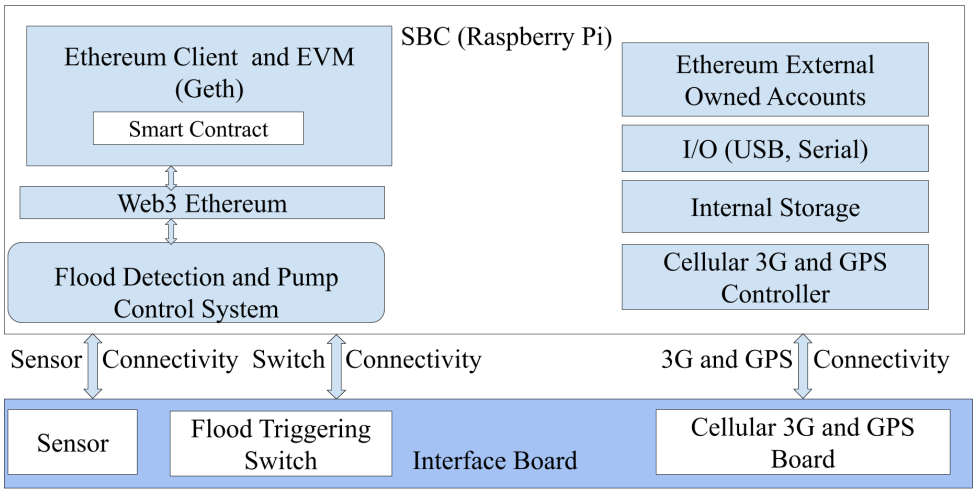


Figure 3.8: System components and connectivities.

3.5.1 Deployed system

The deployed IoT-blockchain system includes the following:

- A network containing 16 nodes was created, with one node controlling the water pump.
- Nodes can communicate among themselves using wireless communication (Wi-Fi or cellular)
- Each node has an Ethereum Geth client (specifically, clique PoA) and has its own EOAs.
- A smart contract that includes the following functions was created:

  - A function to establish the initial value of the global positioning system (GPS) designated area and the threshold of the water level.
  - A function for extracting GPS longitude and latitude data to ensure the node is within the designated area.
  - A function to allow nodes to submit water level readings.
  - A voting algorithm, based on the majority function, which is only invoked by the node that controls the pump to calculate the number of flood detection nodes and trigger activation of the pump if the majority of nodes indicate that a flood is occurring.

The diagram in Fig.3.8 presents the different components of the system and their connectivity. The system was tested for different block periods over Wi-Fi and over a 3G cellular network.

3.5.2 Test setup

The system was tested in a controlled environment for flood detection and control, and it was successful. Testing was continued, however, using a switch on the interface board to emulate flood detection, with nodes distributed around the city of Sheffield in the United Kingdom, see Fig.3.9 for nodes' locations. This was a compromised situation as it allowed for focusing the testing on aspects of the blockchain. The test scenario includes the following:

- A peer-to-peer connection is achieved through the implementation of User Datagram Protocol (UDP) hole punching using a rendezvous server [157].
- Nodes were distributed around the city of Sheffield.
- The tests were conducted for BPs of 1, 2, 3, 4, 5, 10, 15, and 20 seconds.
- The transaction arrival time and the system end-to-end latency were measured.
- Python programs were developed for the purpose of monitoring the status of the network and reporting the timestamps of transaction submissions and the time of the consensus on the network and the change of status.
- The transaction submission time could occur at any time during the BP. Delaying transactions until as late in the BP as possible, however, can ensure that all events are detected and that the latency is reduced. Considering that the aim of this system was to monitor any changes in the environment (water levels), this was important. The system was tested for three different scenarios related to the transaction submission time for all BPs under consideration:
  - Transaction submission at the start of the BP (t_0).

– Transaction submission randomly during the BP.

– Transaction submission at the ideal time (t_i)

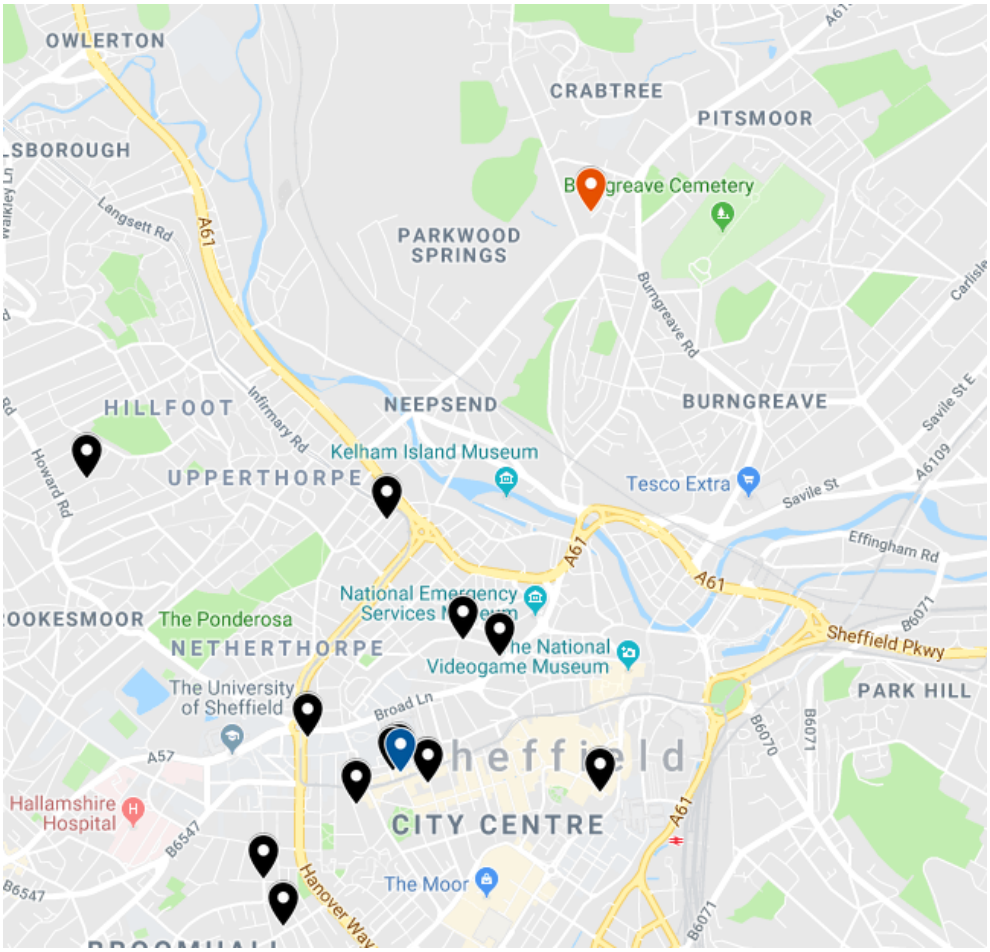


Figure 3.9: Nodes locations during testing over cellular 3G.

3.6 Results

For latency measurements, three different times to submit transactions to the smart contract: at t_0, randomly during the BP, and at t_i; was used. The following sections present the latency measurement results as well as discussion and comments regarding these results.

3.6.1 Ideal time for transaction submission during the system steady state

First, the ideal time for transaction submission was calculated. As can be seen in Fig.3.3, the mining of Block i happens right at the start of the block period, at time T_m. The ideal submission time was defined as t_i, which is the time towards the end of the BP and immediately before entering the critical period ΔT. To calculate this ideal time, the ΔT had to be identified and the sensor latching period (S_{LP}) and the transaction propagation delay (T_{pd}) were both calculated. *Sensor Latching Period (S_{LP}):* The S_{LP} for different distances was measured. As the distance from the water level increases, the latching period will increase, forming a linear relationship. In the implemented case, the water level threshold was 10 cm; the average sensor latching period to measure this distance was 0.614ms.

Transaction propagation delay (T_{pd}): Each node on the network submits transactions to the smart contract, and once they are accepted, they will be propagated to the other nodes on the network. The T_{pd} was measured for both transmissions over the Wi-Fi network and the cellular network, and the results are shown in Table 3.6. As can be seen from the table, propagation delay over the cellular network was higher than propagation delay over the Wi-Fi network. In this test, the Fona 3g board was used, which limits the connectivity to 3G.

Table 3.6: Transactions propagation delay (T_{pd}).

Over Wi-Fi				Over Cellular (3G)			
Avg	Max	Min	STD	Avg	Max	Min	STD
0.09 s	0.2s	0.064s	0.32s	1.8s	3.4s	0.6s	0.7s

Critical period ΔT: ΔT is the period during which miners fetch and add transactions to the new block. Based on the experiments and tests, it can be concluded that the final $\approx 400ms$ of the BP is the critical period, where any transaction arriving during this period has a very low probability of being included in the next block; instead, it will likely have to wait for the block after the next one. From the above measurements of ΔT, S_{LP}, and T_{pd}, the ideal time t_i for transaction submission can be calculated as follows:

$$t_i = BP - (\Delta T + AverageS_{LP} + MaxT_{pd})$$

For $BP = 20s$ and water level = 10cm and testing over the Wi-Fi network, the following was obtained: $t_i = 20–(0.4+0.000614+0.2) \approx 19.39s$ Using the measurement of t_i, it was possible to monitor the water level during the BP until t_i, at which point it was not possible to submit the transactions. By doing this, the following was achieved:

- Reduce the overall system latency.

- Ensure that all flood events can be detected on time and without extra delay by continuously monitoring the water level because submitting transactions at the start of the block could have resulted in a flood incident occurring after the submission, which would have resulted in extra latency of up to 1 BP.

3.6.2 Transaction arrival time (TAT)

The transaction arrival time in the network over both Wi-Fi and cellular networks was measured. The results were compared with the analysed values for all BPs under consideration. Fig.3.10 shows both the measured and analysed (using equation [3.4]) transaction arrival times for transaction submission at t_0. The transaction arrival time was only measured during the steady state because the system only has 16 nodes deployed. For the Wi-Fi results, all BPs were almost identical to the values obtained from the analysis. Conversely, the results of the test that was conducted over the 3G network demonstrates the effect of T_{pd} (on average, it was 1.8s (see Table.3.6)). This delay has a major effect on the arrival time, especially when shorter BPs are implemented (i.e. 1 second, 2 seconds, and occasionally 3 seconds). From Fig.3.10, this becomes clear when the measured values are compared with the analysed values. Based on this, BPs of 1 second, 2 seconds, and 3 seconds are very difficult to implement over a 3G cellular network. This is also clearly illustrated in Fig.3.11 and Fig.3.12; both present the average transaction arrival time for all BPs for transaction submission at t_0, at random time, and at t_i for both networks.

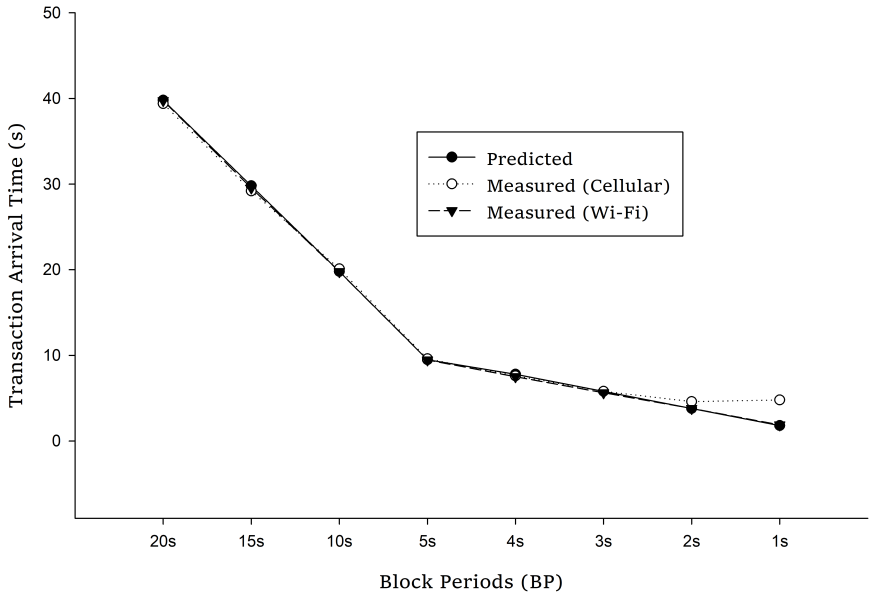


Figure 3.10: Measured and predicted transaction arrival time.

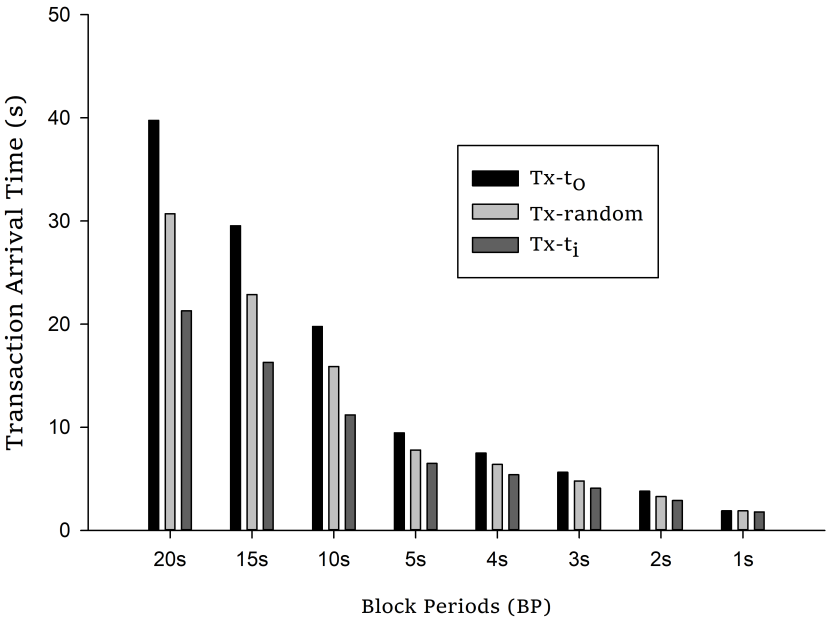


Figure 3.11: Transaction arrival time over Wi-Fi network.

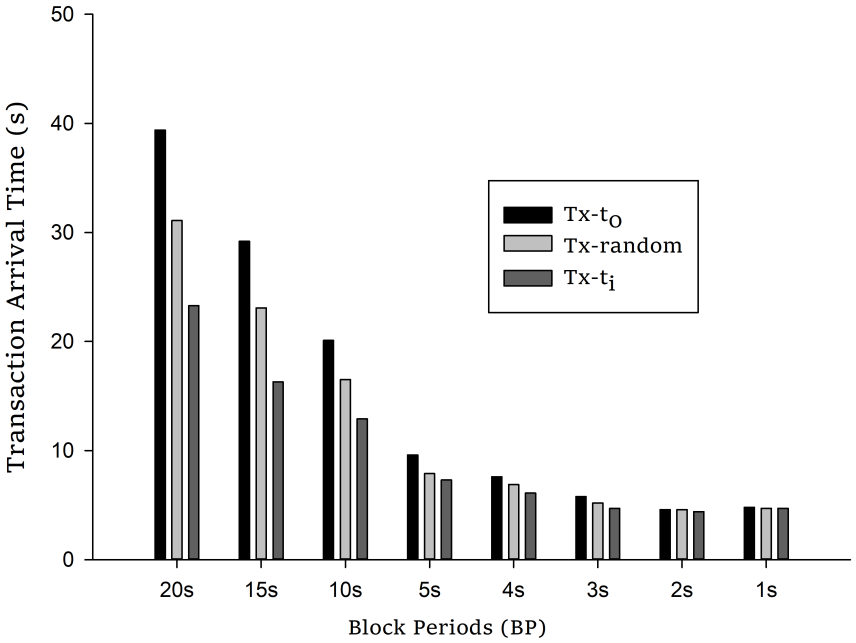
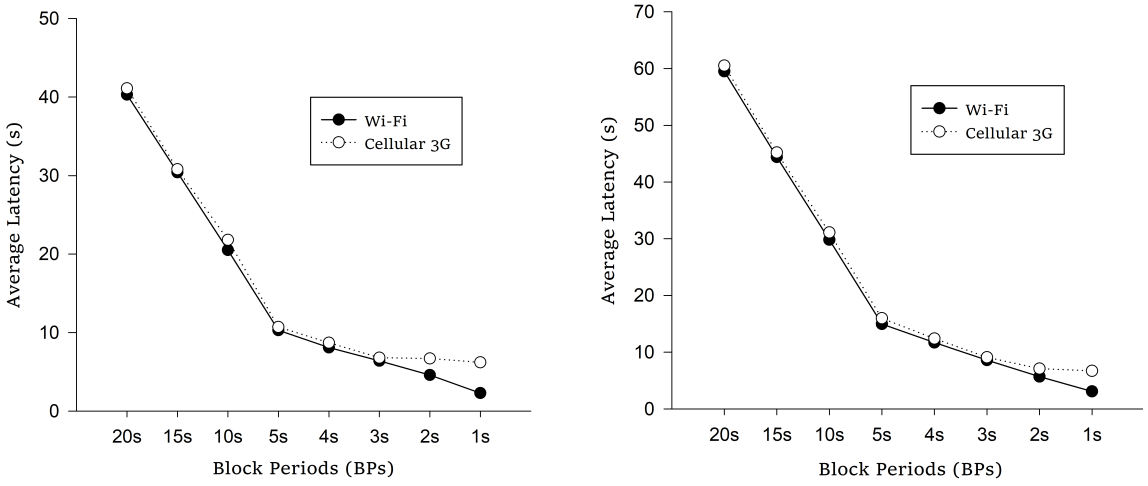


Figure 3.12: Transaction arrival time over cellular network.

3.6.3 End-to-end system latency

Ethereum miners add transactions to a block based on the amount of gas the transaction charges. Transactions that charge higher gas have priority to be added first to the block and mined before others. To prevent this from affecting the processing of the system voting algorithm before water level readings are processed, the measure of submitting water level transactions during the even blocks and invoking the voting algorithm during the odd blocks were introduced. This step introduces an extra latency equal to 1 BP. The test was conducted over both cellular and Wi-Fi networks for comparison purposes and to determine the effect of using a network with limited bandwidth on the overall latency and network synchronisation.

Figure.3.13 shows the average latency for all BPs implemented. As discussed previously, it is again clear that BPs of 1 second, 2 seconds, and 3 seconds cannot be implemented when a 3G cellular network is used. These three block periods will not help with efforts to achieve less latency; in fact, they will simply disrupt the synchronisation of the nodes, resulting in more nodes being out of sync with the network, and might cause the execution of the voting algorithm on obsolete water readings. Conversely, the implementation of all BPs over Wi-Fi was possible, except for the BP of 1 second, which occasionally could not be implemented. Unlike over the cellular network, BPs of 2 seconds and 3 seconds were possible to implement, and less latency was achieved. When implementing the BP of 1 second, however, there were occasions where the network stability was affected and implementation of a BP of 1 second caused the execution of the voting algorithm on water readings submitted by out-of-synch nodes.



(a) Tx submission at t_i

(b) Tx submission at t_0

Figure 3.13: End-to-End system latency over Wi-Fi and cellular networks.

3.6.4 Latency as a function of the block period

The network synchronisation is the ability of the network to ensure that all water level readings are processed and confirmed by the blockchain network within an acceptable amount of time. This will provide the desired accuracy for the system to monitor and control the water pump. During the steady state of the system, where all arrived transactions are included in the next mined block, the transactions propagation delay has a great effect on the implementation of short BPs. Figure.3.14 present the predicted and measured standard deviation of the end-to-end system latency as function of the BP over both Wi-Fi and cellular. As can be seen in the figure, synchronisation and stability of the network were not achieved for all BP implementations, especially during testing over the cellular network. This is due to the bandwidth limitation and the increased transaction propagation delay, which sometimes exceeded the BP. The 1-second BP implementation recorded the highest standard deviation, which rendered the accuracy and the certainty of the voting algorithm poor. The standard deviation decreased, however, as the BP increased, making the network more stable, with almost perfect execution of the voting algorithm. The network has only 16 nodes, each submitting three transactions during each even-numbered block. Within IoT, tens of thousands of nodes could participate in such a network, and this would increase the wait time and the latency. This is one of the limitations of this study: it was not possible to implement thousands of nodes to conduct more synchronisation testing. Nevertheless, the analysis of the implemented system provides a prediction module for the TAT during busy periods in the presence of thousands of nodes.

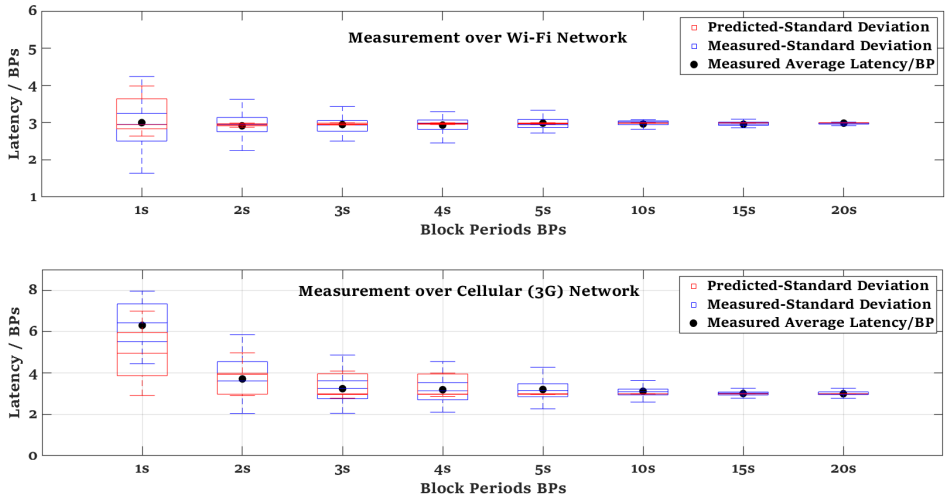
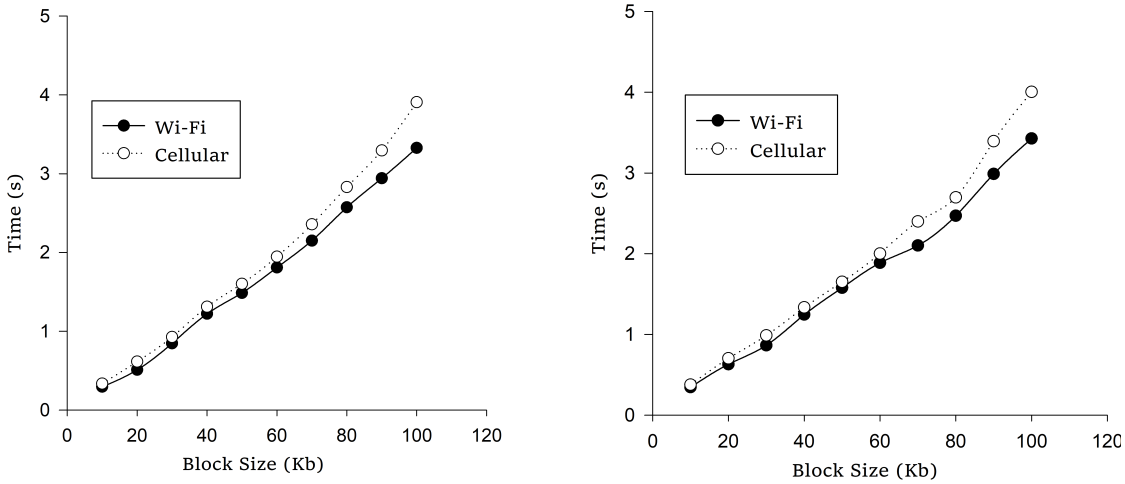


Figure 3.14: End-to-End system latency as function of the BP (Tx at t0).

3.6.5 Durations of block-related events

The durations of block importing, mining, and announcement are affected by the block size. As illustrated in Fig.3.15, these durations increased as the block size increased, resulting in the need for more processing time and power to accomplish them. This can be a problem for IoT devices, which have limited computation power, and could also be a problem for the implementation of shorter BPs. The latter issue could result in synchronisation problems because as the length of the global chain increases, the length of local copy in the IoT devices will become shorter. This is because nodes are not able to import another block before the release of the previous block; they are therefore not able to catch up with the global chain. This means that the freshness of the data and the current state of the blockchain will become uncertain.



(a) Block importing

(b) Block announcement

Figure 3.15: Delay when importing and announcing blocks by the nodes

3.6.6 Energy consumption

Ethereum PoA relies on trusted nodes to sign and propagate blocks, and this has a significant advantage in terms of power consumption because nodes do not have to perform any computational work. It is, however, important to characterise the system in terms of power consumption for deployment purposes, where the only source of power might be batteries. The Keysight 34450A 5 $\frac{1}{2}$ Digital Multimeter was used to measure the average current draw by the Raspberry Pi. First, the average when the Raspberry Pi was idle was measured and converted the average value to an average energy consumption. Subsequently, the average energy was measured for different cases, as shown in Fig.3.16. The energy consumption when running the full flood detection system over both Wi-Fi and cellular was measured. During both tests, the node being tested was a fully functioning node. A fully functioning node submits at

least two transactions each BP, signing and propagating blocks in turn and importing and adding blocks to its local copy. Each test was run for over 30 minutes with 10s as the BP, and over 190 blocks were generated and propagated in the network, with different sizes that ranged from 607 bytes to 100 Kbytes.

The results in Fig.3.16(c) indicate that there is a minimum increase in energy consumption of 0.36J (when testing the system over Wi-Fi) compared with Fig.3.16(a) (when the raspberry pi is in idle state). By contrast, the difference between the energy consumption of Fig.3.16(e) (when testing over 3G) and Fig.3.16(a) is more than double (2.95 J); this is due to the power drawn by the Fona 3G board. When all the measurements are analysed, the average energy consumption of running the flood detection system including the Ethereum Blockchain Geth client, regardless of the communication link, is a small amount of energy (around 0.3J). Knowing this result is crucial in selecting which method to use to power the nodes, especially in choosing the right batteries when deploying the system over cellular if no adequate power source is available.

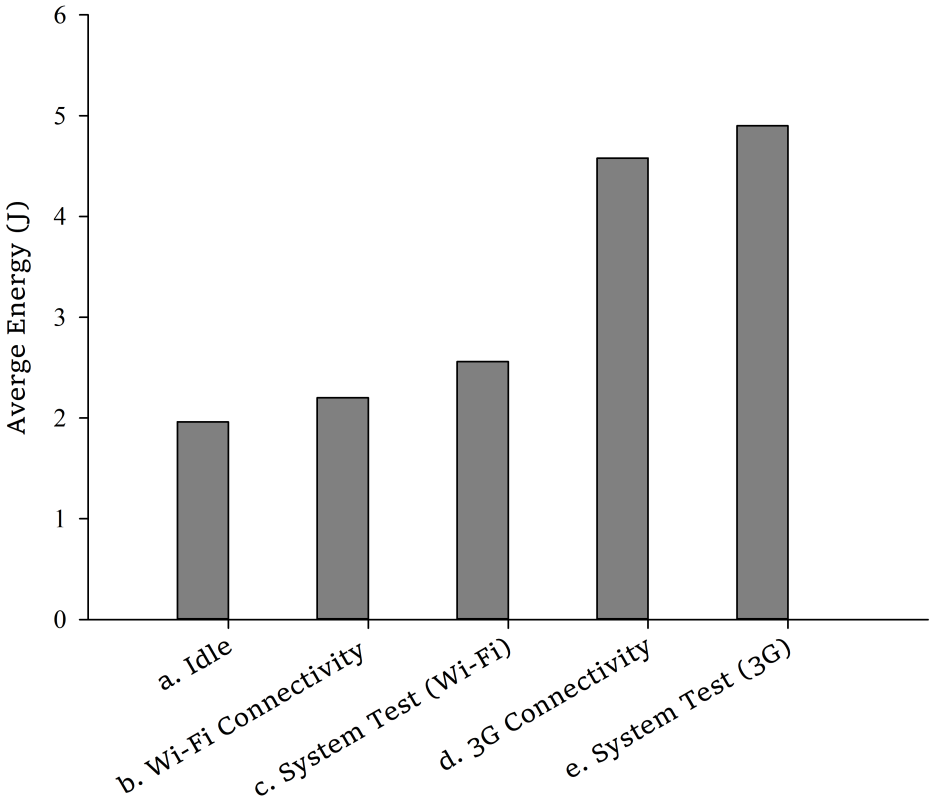


Figure 3.16: Average energy consumption during different system states.

3.7 Summary

In this chapter, a real-world IoT-blockchain use case in the form of a flood monitoring and detection system was implemented. A performance analysis was provided, which included measuring transaction arrival time from submission by the node until the transaction's arrival on the network and measuring the system end-to-end latency for different block periods over a cellular network and Wi-Fi. The network stability and node synchronisation for various BPs in different transaction submission scenarios was studied. A study with a measurement of the average energy consumption was also provided, and that the average energy consumption of running the flood detection system including the Ethereum blockchain Geth client, regardless of the communication link, is a small amount of energy (around 0.3J) was demonstrated.

The study and analyses in this chapter showed that blockchain can be integrated into IoT applications, and that Ethereum PoA can be used within IoT for permissioned implementation. To conclude, it is important to consider the application requirements, especially in terms of criticality. Also, it is important to consider the type of communication protocol in use and the number of nodes and their locations when deciding which block period and block gas limit to implement.

Chapter 4

HDPoA: Honesty-based Distributed Proof of Authority via Scalable Work

This chapter introduces a novel consensus protocol called honesty-based distributed proof of authority (HDPoA) via scalable work that is an IoT centric. HDPoA is based on proof of authority (PoA) and proof of work (PoW). With the integration of PoW, HDPoA is able to realise the security advantages provided by PoW. This is achieved by utilising the IoT devices' collective computation power to mine and generate a new block. In this chapter, HDPoA was analysed and then deployed and tested utilising a purpose built testbed using devices that are low-cost. A performance measurements and evaluation along with the security analysis of HDPoA is provided in this chapter as well.

4.1 Introduction

Blockchain provides the IoT systems with a solution that can enhance their security due to blockchain's unique characteristics such as its distributed nature, the decentralisation approach to its design, and its ability to be a self-managed technology. This enhancement can be in the form of ensuring the integrity of the IoT collected data, ensuring users are accountable for their actions, eliminating single points of failure, and providing a better way of implementing access controls [158, 159]. The self-managed features of the blockchain can provide the IoT systems with a reliable and secure ability for controlling the distribution of computation works over a large number of connected devices,[23].

4.1.1 Problem description

Integrating blockchain into IoT systems presents some challenges. One of the main challenges is the ability to design a consensus algorithm suitable for the deployment and utilisation within the IoT realm, where some devices may lack adequate resources, such as computation power.

Many authors have proposed different consensus protocols for IoT implementation. The work by [69] introduced a new protocol based on PoW called the proof of trust, suitable for implementation in IoT systems. The work in [149] introduced the credit-based consensus for IoT-blcokchain applications. Similarly, the authors of [71] have proposed another consensus mechanism for IoT systems. It is based on the concept of proof of credit (PoC) combined with voting-based chain finality (VCF), where a subset of validators runs and manages the consensus process. All of [69, 71, 149] tend to rely on trusted nodes; it is easier for nodes with higher trust values to mine blocks. As a result such a blockchain network may become more central, where it may be possible that the network is controlled by just a single or a few nodes.

The work by [70] has shown by the introduction of the sub-blockchains concept that PoW can be integrated into IoT-blockchain applications. However, their implementation focused on a permission blockchain system. The work by [73] proposed the geographic practical Byzantine fault tolerant (G-PBFT) consensus mechanism, which is designed for IoT-blockchain applications. G-PBFT uses the geographic locations of IoT nodes to ensure that such nodes are not malicious. The locations of the IoT devices should be fixed for this consensus to be secure, making it unsuitable for dynamic and mobile IoT networks.

While the above works proposed different consensus protocols suitable for IoT-blockchain applications, nevertheless they all fail to provide a fully decentralised and secure protocol for public IoT-blockchain platforms. This means there is a requirements when integrating the blockchain technology into IoT systems for the design and development of a suitable consensus protocol. Unlike current general purpose and cryptocurrency-driven algorithms, the design of an IoT-centric consensus mechanism should address the main security and performance requirements of blockchain-based IoT applications.

Consensus mechanism characteristics

The main characteristics of any consensus mechanism that is designed for IoT-blockchain applications can be summarised as follows:

1. It must have the capability to deal with and handle as many dishonest and faulty nodes as possible and be able to limit these nodes ability in participating in the consensus or in harming the network. This means it must be resilient against different attacks such as DoS and double spending.

2. *Transaction finality*, wherein transactions cannot be deleted or modified, is an important element of the consensus security. Any IoT-centric consensus protocol

need to be able in ensuring that all transactions reach final and confirmed state on the network securely and as with lower latency. Reaching consensus finality would enhance the network abilities in avoiding forks from happening and minimising the transaction confirmation time. In the event that a fork occurs, the protocol should ensure that all nodes sync with the main and valid chain.

3. The design of IoT-centric consensus mechanisms should consider the possible drawbacks of constrained devices. It is an essential requirement that consensus algorithms utilise the available computation power of the devices without major consequences on the devices' individual energy consumption or their overall performance.

4. As the number of devices on the IoT network increases, the algorithm must be capable of adapting to high throughput since the number of transactions increases to ensure acceptable transaction confirmation times according to the requirements of the application.

One of the most secure consensus protocol and is the de-facto consensus protocol choice for decentralisation implementation of public blockchain is PoW [13]. However, its implementation within IoT systems is very difficult due to the high power requirements. In addition to PoW, there are many alternative consensus protocols that can be used by developers when designing and implementing their blockchain platforms, but most of these protocols either tend to be more centralised through the implementation of permissioned networks or they lack the security elements provided by PoW. One of these protocols is the PoA, namely the Ethereum Clique protocol [153].

PoA offers great advantages in-terms of lower transaction confirmation time and that it does not requires much computation power to produce blocks. However, one disadvantages of this protocol is that it is only suitable for permissioned blockchain implementation and suffer from the centralisation issues as a result of it being reliant on small number of authority nodes. In this chapter the PoA will be extended into a public and decentralised consensus protocol called HDPoA. This is done through the integration of another security layer based on PoW through honesty via scalable work. Table. 4.1 provides summary of related works in comparison to the proposed HDPoA

Table 4.1: Summary of some important related protocols compare to HDPoA

Consensus algorithm	Computation Complexity	Possibility of forks	Nodes scalability	Vulnerability	Type of access	Transaction Finality	Suitability for public IoT-Blockchain
PoW	High	Yes	High	51% attack	Permissionless	High	No
PoA	Low	Yes, but dealt with efficiently	Low	- Faulty nodes (total node-1/3) - Heavily depends on validators honesty	Permissioned	Low	No
PoS	Low	Yes	High	-51% attack -Collusion of rich stakeholders -Nothing at stake attack.	Permissionless and Permissioned	High	No
Proof of Activity	High	No	High	-51% attack -Collusion of rich stakeholders	Permissionless and Permissioned	High	No
PBFT	High	Yes	High	- Faulty nodes (total node-1/3) -DoS attacks	Permissioned	Low	No
Credit-based PoW	High	Yes	High	-51% attack -Collusion of credit-rich stakeholders -Centralisation issues	Not given	High	Yes
PoAh	High	Not given	Low	- Heavily depends on trusted nodes -Centralisation issues -DoS attacks	Permissioned	Not given	No
HDPoA	Low in term of impact on individual device	Rarely, but dealt with efficiently	High	51% attack	Permissionless	Low (almost immediate)	Yes

4.1.2 Contribution of the chapter

The contribution in this chapter can be summarised as follows:

- A new consensus mechanism called Honesty-based Distributed Proof of Authority (HDPoA) via Scalable Work is proposed. HDPoA is Based on PoA and PoW suitable for implementation in IoT applications.
- HDPoA realises the security provided by PoW while, reducing the confirmation time to just one block. This is achieved by integrating PoA and PoW into an HDPoA protocol.
- The design and implementation of a proof of concept permissionless blockchain system for testing and evaluating HDPoA using a total of 30 different IoT devices comprised of Raspberry Pis, ESP32, and ESP8266.
- Performance analyses and evaluation, using off the shelf IoT devices, of a permissionless blockchain system based on HDPoA protocol.

Published papers

The following papers based on this chapter were published:

1. S. Alrubei, E. Ball, and J. Rigelsford, "Hdpoa: Honesty-based distributed proof of authority via scalable work consensus protocol for IoT-blockchain applications," Computer Networks, vol. 217, p. 109337, 2022, doi: https://doi.org/10.1016/j.comnet.2022.109337.
2. S. Alrubei, E. Ball and J. Rigelsford, "Securing IoT-Blockchain Applications Through Honesty-Based Distributed Proof of Authority Consensus Algorithm," 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), 2021, pp. 1-7, doi: 10.1109/CyberSA52016.2021.9478257.

4.1.3 Organisation of the chapter

The rest of the chapter is organised as follows. Section 4.2 presents the architecture design followed by the the system analysis in section 4.3. Section 4.4 presents the system security analysis and risk assessments followed by the system implementation and experiment in 4.5 . This then follow by the results and evaluation in section 4.6, and finally, the chapter's summary is presented in section 4.7.

4.2 Architecture design

The previously discussed characteristics of a consensus's mechanism are all satisfied by the introduction of honesty-based distributed proof of authority via scalable work

(HDPoA) consensus mechanism. The following subsections will provide more detail of the design architecture of HDPoA.

4.2.1 Role of IoT-Devices

IoT devices may be incapable of providing the required computational power needed for some of the consensus protocols if they lack adequate resources. Therefore, designing a suitable consensus mechanism is an important factor when applying blockchain into IoT applications. There are many connected devices within IoT systems, each with different capabilities and different roles to play. Based on their computation power and storage capabilities these devices, as shown by Fig.4.1, can be divided into the following:

- Full nodes (FN). These are low-cost devices that have sufficient computational and storage capabilities to allow them to store the full copy of the chain and act as miners and validators in the network while also participating in the mining process.
- Hybrid nodes (HN). They are devices that are low in cost and power and are incapable of handling the storage of a full copy of the chain. However, their computation power can be used to perform small tasks, such as carrying the search for the right hash within a small set of nonces. They can also create and submit transactions and store the headers of the blocks.
- Participant nodes (PN). These are sensor devices that cannot store either the chain or the headers of the blocks. Yet, they have a major role to play in the network, as they are data feeders. They can be connected to any node (both hybrid nodes and full nodes), and they can submit transactions to the network through these nodes.

In HDPoA, as shown by Fig.4.1, the IoT devices based on their roles in the consensus mechanism were divided into two categories as follows:

- Worker nodes (WN). Any node that joins the blockchain platform will join as a WN; this includes both hybrid and full nodes. By correctly carrying out assigned workload, a WN can improve its honesty and trust level. Once a node has a high enough level of honesty and sufficient resources, it will be promoted to the authority nodes category. All nodes are assigned mining work on the network; these include trusted authority nodes and normal nodes.
- Authority nodes (AN). These are typically low-cost devices and are full nodes that store a copy of the global chain on their local storage. They sync with the most recent chain in the network. They are responsible for coordinating and managing the block generation process, validate any workers' work, reward worker nodes by increasing their honesty level, sign and propagate blocks, and validate newly

propagated blocks. Only the full nodes that have adequate computational power and storage capabilities can be authority nodes. This is because ANs need to store the chain of blocks locally in order to validate transactions and new blocks and when the AN is not in charge of managing the block generation process it will act as WN and participate in the block generation process by carrying out small amounts of the block mining process.

4.2.2 Honesty via scalable work

The proposed HDPoA is a new consensus mechanism that is intended to facilitate a decentralised and permissionless blockchain platform. In classical PoA, nodes are authorised to join the network by the network owner or administrator. Only a few nodes (the authority nodes) are allowed to generate blocks at fixed times and validate blocks arriving from other nodes. This will result in a more centralised network that is in opposition to the foundational concept of blockchain technology, which is

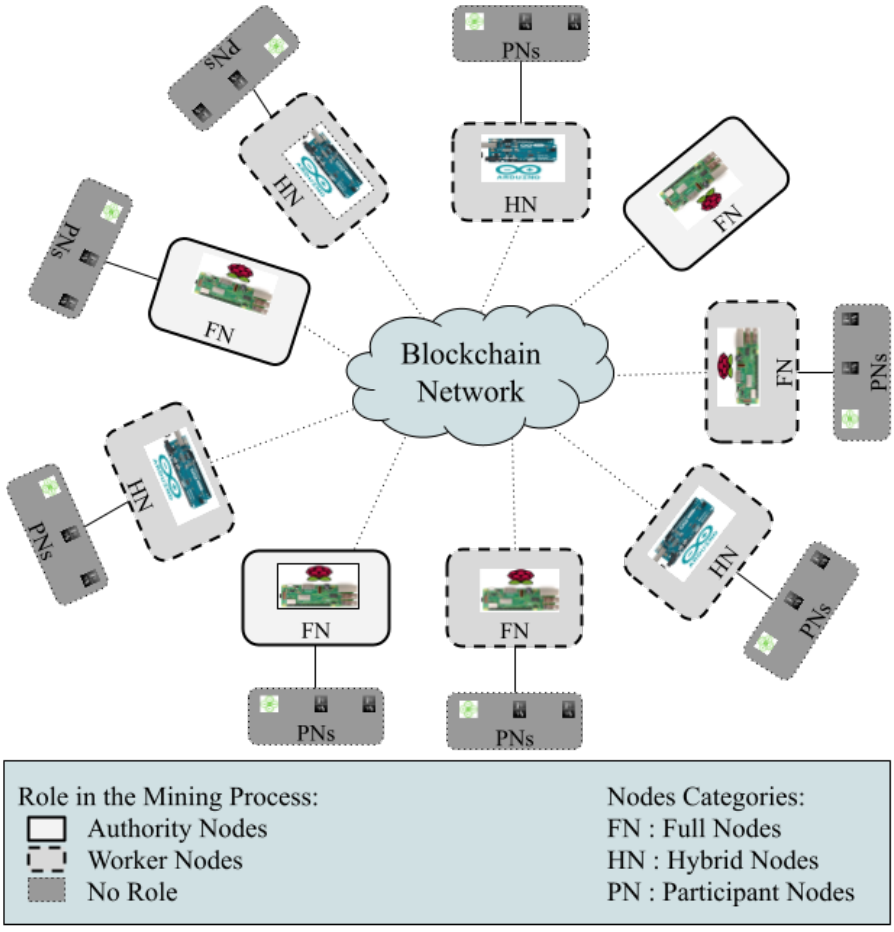


Figure 4.1: IoT devices' categories and their role in the mining process.

decentralisation. PoW on the other hand is the de-facto consensus algorithm choice for decentralisation implementation of blockchain and one of the most secure as well. However, it suffers from the long confirmation time (up to six blocks needed) and the ever increasing energy requirements as a result of the need for more power, which some IoT devices can not provide individually. In order to achieve the secure and the more decentralised and distributed consensus mechanism that is suitable for implementation within the IoT realm, a protocol that integrates the advantages provided by PoA and PoW into a new consensus mechanism was designed and developed. This is achieved through the introduction of two new concepts called *honesty level and scalable work*.

In the concept of scalable work, nodes have to perform work on the network to increase their honesty level. In turn, this would allow the nodes with high enough honesty score to mine and generate new blocks and validate other nodes newly generated blocks. The node may be assigned to perform any form of work, such as mining block or any other useful work.

The integration of the scalable work concept into the HDPoA consensus is an essential part of the HDPoA design process. It allows for the facilitating of a permissionless blockchain by integrating mining process based on PoW, hence, allowing for the incorporation of PoW security advantages. Through scalable work nodes are allowed to increase their honesty levels, allowing them to be promoted to the authority node category. Thus, the more authority nodes the more decentralise the network.

4.2.3 HDPoA algorithm

Authority nodes (ANs) are responsible for the block generation process. Every time before the generating of a new block a new AN is elected as primary miner, based on the round-robin process, to mange the generation of this block. Another secondary miner will also be elected, to ensure a new block is generated every time. WNs on the network will be allocated a small mining job to perform and must report their results to the elected node. The process of generating a new block begins with the elected primary authority node and spans over four phases.

In the first phase, elected AN will start the process by carrying the initiation steps of creating a pending block, which includes three steps. First, $AN1$ will collect and validate all transactions (Tx) from the transactions' pool and then add them into this pending block. Then, if needed, it will adjust the difficulty (D), set the hash target value H_{tv}, and create mining tasks based on the hash of the last block in the chain (LH), the difficulty (D), and transactions merkle root (MR). Finally, it will adjust the workload for each WN based on WN honesty level, and distribute these workloads (see Algorithm 1).

In the second phase, worker nodes will operate according to Algorithm 2. Any worker node that finds a solution (i.e target hash TH based on target nonce TN), will submit it to all ANs. In the third phase, when the elected primary authority node receives this solution will calculate the new hash NH validate it and sign and propagate a new block (see Algorithm 3).

In the final phase, other authority nodes will validate the newly propagated block and if valid they will add it to their local chain (see Algorithm 4).

Algorithm 1 HDPoA First Phase

AN1 elected as primary miner
AN1 Collects Transactions
if Tx is Valid **then**
    add Tx *to* $newBlock$
else
    Remove Tx from pool
end if
Calculate MR, D and H_{tv}
Calculate $WL(i)$ for each WN
Prepare WNs' tasks
return LH & D & H_{tv} & MR & $WL(i)$

Algorithm 2 HDPoA second phase

Input: LH, H_{tv}, MR, D, and (minNonce and maxNonce).
Output: TN

initial: $nonce \leftarrow minNonce$
while $nonce \leq maxNonce$ **do**
    $H = Hash(D \parallel LH \parallel nonce \parallel MR)$
    if H $value \leq H_{tv}$ **then**
        $TN \leftarrow nonce$
        return TN
    else
        nonce++
    end if
end while

Algorithm 3 HDPoA third phase

Input: LH,H_{tv}, D MR, and TN.
Output: $newBlock$

initial: $blockGenration \leftarrow false$
$NH = Hash(D \parallel LH \parallel TN \parallel MR)$
if NH $value \leq H_{tv}$ **then**
  if $AN \leftarrow primaryMiner$ **then**
    $B_w \leftarrow 3$
    if $AN \leftarrow secondaryMiner$ **then**
      $B_w \leftarrow 2$
    end if
  end if
  $newBlock.Height = lastBlock.height + 1$
  $newBlock.timestamp = Date.now()$
  $blockGenration \leftarrow true$
  $newBlock.sign()$
  Propagate $newBlock$
else
  $TN \leftarrow invalid$
end if
Wait for another WN

Algorithm 4 HDPoA final phase - Other ANs validate the new block

Input: LH, NB, H_{tv}, $lastBlock.height$,and TN.
Output: Valid newBlock

if $newBlock.Height > lastBlock.height$ **then**
  if All Tx in $newBlock$ Valid **then**
    Calculate MR
    if $MR = newBlock.MR$ **then**
      $H = Hash(D \parallel LH \parallel TN \parallel MR)$
      if H $value \leq H_{tv}$ **then**
        $newBlock \leftarrow valid$
        Add $newBlock$ to local chain
        return true
      end if
    end if
  end if
else
  $newBlock \leftarrow invalid$
  return false
end if

4.3 System analysis

HDPoA is a permissionless consensus mechanism that is suitable for the implementation into IoT system. In HDPoA the main source of data is the block propagation process and the transaction propagation process. There can be different type of transactions on the HDPoA-based blockchain, in this chapter regular and special transactions are considered.

Regular transactions Tx are the records of any new data on the digital ledger. This includes any update to previously added data, or the exchange of information or value between two nodes on the network. Reward transactions are special transactions issued by ANs to reward any WN that produces a valid work by increasing its honesty level. Deduction transactions are another special transaction issued by ANs to reduce the honesty level of any WN that misbehaves (i.e submit the wrong solution to a task). Table. 4.2 provide a lis of the used parameters and their definitions.

4.3.1 Mining time and hash power

HDPoA consensus mechanism utilises SHA-256 for producing the hash value. When the hash target value H_{tv} is at its maximum value the difficulty D is at its lowest level which is one. In other words, the hash has 24 leading zeros (this is because one R-pi can mine a block at this difficulty in approximately 10-13 minutes, this is in-line with the average bitcoin mining time), this means the $H_{tv} = 2^{256-24} = 2^{232}$. Based on this, any new difficulty D can be calculated by:

$$D = \frac{2^{232}}{H_{tv}} \tag{4.1}$$

If node i has a processing instructions clock rate of C_r, then for a message of size M bytes, the number of instructions clock ticks at C_r required to hash one byte can be defined as C_b. Based on this the hash power h_p of node i can be calculated by the following equation:

$$h_p(i) = \frac{C_r}{M * C_b} \tag{4.2}$$

If the total available WNs on the network at any given time t to participate in the block generation process is N, then the total hashing power on the blockchain network can be calculated by:

$$h_p = \sum_{i=1}^{i=N} h_p(i) \tag{4.3}$$

Base on this, the mining time t_m (i.e the time taken to find the right nonce that produces the hash of the next block that satisfies the required difficulty) can by calculated:

$$t_m = \frac{D \times 2^{24}}{h_p} \tag{4.4}$$

Table 4.2: Parameters used and their definitions

Parameter	Definition
H_{tv}	Hash target value
D	Difficulty
C_r	Clock rate
h_p	Hash power
C_b	Number of instructions clock ticks at Cr required to hash one byte
M	Message size in bytes
t	Time
N	Total WNs participating in mining the current block.
t_m	The mining time
Tx_{ct}	Transaction confirmation time
λ	Blocks' arrival rate
B_{pd}	Block propagation delay
T_{pd}	Transactions propagation delay
t_v	Time needed by one AN to validate the new block
B_{size}	Block size
Tx_{size}	Transaction size
NB	Number of blocks the network produces per day
P_s	Power consumed when the node is in sleep mode
P_{tx}	Power consumed during sending any data
P_{rx}	Power consumed during data receptions
P_m	Power consumed during node participation in the mining process
N_w	Number of mining works per hour
E_n	Energy consumption
t_s	Time in sleep mode
t_{tx}	Time when the node transmitting data
t_{rx}	Time when the node receiving data
B_c	Battery capacity
V	Voltage
B_L	Battery Life
H_i	Node i honesty level
H_T	The network's honesty threshold
S_p	Positive score
S_n	Negative score
w	Total number of tasks
H_p	Positive honesty level
H_n	Negative honesty level
HP_F	Hash power factor
H_F	Node's honesty factor
WL	Workload
W_{tot}	The total work that represents the maximum hash iteration
NB	Number of blocks the network produces per day
H_w	Honest AN voting weight
M_w	Malicious AN voting weight
γ	The voting weight factor
A_s	Attacker's maximum attack surface (total weight the attacker can control)

4.3.2 Transactions confirmation time and throughput

Transaction confirmation time Tx_{ct} is an important aspect of HDPoA, HDPoA was designed for transaction to be confirmed on the network as fast as possible. As the protocol added a second security layer on top of PoW, which is the introduction of the trusted honest ANs. Once the block has arrived to the ANs, they will validate it and all transactions inside it according to Algorithm 4, and immediately added it to their local chain. This means transactions will be confirmed without the need for an additional round of confirmation.

The probability of any transaction to arrive and confirm on the network can be measured based on the Poisson process, in which the outcome can arrive on a confirmed block with an arrival rate of λ.

$$P(T \leq t) = 1 - e^{-\lambda t} \tag{4.5}$$

The proposed HDPoA consensus mechanism does not require extra time to confirm the arrived block, as long as none of the authority nodes (ANs) initiate a block-rejection process. Based on this, λ can be defined as $\lambda = \frac{1}{t_m}$ block/s. The t parameter (i.e., time) relies on the number of blocks n the user needs to wait before the block carrying the transaction is confirmed on the network, the probability ($P(n)$) of the confirmation of any transactions can be calculated as:

$$P(n) = 1 - e^{-\left[\left(\frac{1}{\frac{D \times 2^{24}}{h_p}}\right) \times n\right]} \tag{4.6}$$

The number of the blocks n the user needs to wait for before the transaction is confirmed depend heavily on the t_m as HDPoA designed to allow for the immediate confirmation of any valid block as long as no AN initiate block rejection process. There are also two important transmission parameters that also has an effect on the Tx_{ct}. These parameters are the transactions propagation delay Tx_{pd} and the block propagation delay B_{pd}. Another factor that might effect the transaction confirmation time is how long an AN needs to perform the block validation process, which can be defined as t_v, this means the probability ($P(n)$) is calculated by:

$$P(n) = 1 - e^{-\left[\left(\frac{1}{\frac{D \times 2^{24}}{h_p}}\right) \times n \times \left(\frac{D \times 2^{24}}{h_p} + B_{pd} + t_v + Tx_{pd}\right)\right]} \tag{4.7}$$

Based on these factors and parameters and be rearranging 4.7, the Tx_{ct} can be calculated by:

$$Tx_{ct} = \frac{\ln(1 - P(n))}{\frac{-1}{\frac{D \times 2^{24}}{hp}}} + B_{pd} + t_v + Tx_{pd} \tag{4.8}$$

To provide an estimation of the network throughput (i.e transactions per second), assuming the block size is B_{size} and the transactions size is Tx_{size}, then the network

throughput can be calculated by:

$$Throughput = \frac{\frac{B_{size}}{Tx_{size}}}{Tx_{ct}} \tag{4.9}$$

4.3.3 Battery life and power consumption

The power consumed when the node is in sleep mode can be defined as P_s, and the power consumed during sending any data can be defined as P_{tx}, the consumption of power during data receptions can be defined as P_{rx}, and the consumption of power during the node participating in the mining process can be defined as P_m.

If a node performs N_w number of mining works per hour then its energy consumption E_n can be calculated by:

$$E_n = (P_s \times t_s) + (P_{rx} \times (t_{rx} \times N_w)) + (P_{tx} \times (t_{tx} \times N_w)) + \left(P_m \times (\tfrac{D \times 2^{24}}{h_p} \times N_w)\right) \tag{4.10}$$

Where t_s is the time when the node in sleep mode, t_{rx} is the time the node spends receiving data, t_{tx} is the time to send the data.

HDPoA is designed to allow devices with small batteries to participate in the mining process without substantial impact on these batteries' life. A battery with capacity B_c with cell voltage of V its estimate life in hours B_L can be calculated by:

$$B_L = \frac{B_c \times V}{(P_s \times t_s) + (P_{rx} \times (t_{rx} \times N_w)) + (P_{tx} \times (t_{tx} \times N_w)) + \left(P_m \times (\tfrac{D \times 2^{24}}{h_p} \times N_w)\right)} \tag{4.11}$$

4.3.4 Honesty level and workload

The HDPoA consensus mechanism allows for the implementation of a public blockchain platform where nodes can join freely. Each node once joins the network will have no trust and its honesty level H_i is zero. Through scalable work, nodes will increase this honesty level to reach a point where it is possible to act as an authority node, this is when $H_i > H_T$, where H_T is the network honesty threshold. Each work carries both a positive score S_p and a negative score S_n and the node that performs work honestly and produces a correct solution will be able to increase its H_i by S_p, conversely if it produces a wrong solution or behave maliciously its H_i will be reduced by S_n. For a total number of works w equal to j the positive honesty value H_p for node i can be calculated by:

$$H_p = \sum_{w=1}^{w=j} S_p(w) \tag{4.12}$$

Similarly, the negative honesty value H_n for node i can be calculated by:

$$H_n = \sum_{w=1}^{w=j} S_n(w) \tag{4.13}$$

Finally, a node's honesty level H_i can be calculated by:

$$H_i = H_p - H_n \tag{4.14}$$

All nodes during their presence on the network will be assigned a work to perform. The workload corresponds to the node honesty level; the greater H_i the node has the less the workload it will be assigned. If node i has a hash power factor of HPF then its honesty factor HF can be calculated by:

$$HF(i) = \begin{cases} \frac{H_T - H_i}{HPF} & \text{for } H_T > H_i \\ \frac{1}{HPF} & \text{for } H_T \leq H_i \end{cases} \tag{4.15}$$

Based on this, the assigned workload WL for node i can be calculated by:

$$WL(i) = W_{tot} \times \frac{H_F(i)}{\sum_{k=1}^{k=N} HF(k)} \tag{4.16}$$

In (4.16) W_{tot} is the total work that represents the maximum hash iteration (i.e nonce max range). As discussed above the minimum D in HDPoA is when the total leading zeros at the beginning of the hash is equal to 24 zeros. This means the number of valid hashes is 2^{232} based on this the probability P of any resulted hash to be valid is $P = \frac{2^{232}}{2^{256}} = 2^{-24} = 0.00000596\%$. Based on these facts, W_{tot} can be calculated by:

$$W_{tot} = \frac{100}{\frac{2^{(256-(24+D))}}{2^{256}} * 100} \tag{4.17}$$

Based on this, (4.16) can be rewritten as follow:

$$WL(i) = \left[\frac{100}{\frac{2^{(256-(24+D))}}{2^{256}} * 100} \right] \times \left[\frac{HF(i)}{\sum_{k=1}^{k=N} HF(k)} \right] \tag{4.18}$$

4.3.5 Honesty threshold

The network's honesty threshold defines the required honesty level for the promotion of nodes to the AN category; this is when $H_i > H_T$. The threshold depends on three important variables in the system: the battery life, the difficulty, and the available hash power. As more nodes joins the network the difficulty D will increase as there will be more hash power, as a result the honesty threshold will be decreased, thus allowing more nodes to be promoted to the AN category. Assuming that all WNs are performing honest calculations and increasing their honesty level by honesty score of

S_p. If the number of blocks the network produces per day is NB, then the network's honesty threshold H_T can be calculated by:

$$H_T = 10S_p \times e^{-\frac{1}{NB} \times D} \tag{4.19}$$

This means a network with 100,000 nodes would have lower honesty threshold H_T whilst mining blocks at a higher difficulty compared to a network with 10,000 nodes that has higher honesty threshold H_T whilst mining blocks at lower difficulty.

4.4 Security analysis

When designing HDPoA time and effort were directed to the security aspect. In the following subsections security analysis and qualitative risk assessments are presented covering all the important attacks that can cause problems to any HDPoA-based blockchain. The risk assessment was performed based on the NIST SP-800-30 [156] standard, the risk level determination from NIST SP-800-30 as shown in Table. 3.5 was adapted.

4.4.1 Malicious AN

It may be possible that an authority node is malicious or compromised. If this occurs, the HDPoA protocol needs to have the ability to address such a situation and defend the security of the network. Once the node is added to the AN category it will be able to vote on any changes on the network and currently there are two processes that require voting from ANs; block rejection process and malicious AN removal from AN category process. For any of these two processes to be successful the voting weight from all ANs should be at least 51% of the total voting weight on the network. A malicious AN can perform an attack on these processes; the first attack is forging a new block by defeating the voting process and the second is preventing removal of malicious AN from AN category. In HDPoA, once any AN receives the block, it will validate it using algorithm 4 and if valid it will add it to its local chain. Thus any impact of such an attack can be *low*, however, due to the fact that if nodes are not probably secure they could be compromised and their private keys can be stolen, thus, allowing the attackers to sign and propagate blocks easily. This means the likelihood of such an attack can be *very high*. For any malicious AN to be successful in control the network needs to control the voting process, this means the likelihood of controlling the voting process is *low*, however, the impact of successful voting process control is *very high*.

Probability of controlling voting process

Once any AN finds a new block is invalid it will initiate a block's rejection voting process. For this process to be successful the required voting weight needs to be more than half the total available weight on the network. Each AN has a voting weight

that is correspondent to its current H_i. Assuming that the maximum honesty level on the network is H_{max}, and the network honesty threshold is H_T any node that reaches this threshold will be added to the AN category. The honest AN voting weight can be defined by H_w and the malicious AN voting weight is M_w, which can be calculated by:

$$M_w = \gamma \times \frac{H_i}{\sum_{j=1}^{j=N} H(j)} \tag{4.20}$$

In 4.20 j is the total ANs on the network and γ is the voting weight factor and can be calculated by $\gamma = Hi/H_{max}$. Assuming the total number of ANs at time t on the network is j then the total weight the attacker can control A_s can be calculated by:

$$A_s = \frac{\left(\gamma \times \frac{H_i}{\sum_{j=1}^{j=N} H(j)}\right)}{\sum_{j=1}^{j=N} H_w(j)} \tag{4.21}$$

4.4.2 Dishonest WN

It is possible that a worker node can also misbehave or that it is compromised. If a worker node submits an invalid solution to a task the network can deal with it. Any new worker's solution will be valid by the AN that is in charge of the block generation process before accepting it and propagate the new block. Other available ANs on the network will also not accept any new block until they validate it according to Algorithm 4. The security of the network can be also enhanced by the scalable work concept in the case of misbehaving WNs. Honest ANs, if the majority agrees, can punish any misbehaving WNs by increasing the amount of the workload assigned, which could result in the destruction of the node's battery or the excessive use of its power source in non-mining-related work. Based on this, the impact level of any harm from dishonest WN will be low. However, due to the fact that if nodes are not probably secure they could be compromised and their private keys can be stolen, thus, allowing the attackers to submit wrong solutions. This means the likelihood of such an attack can be *very high*.

4.4.3 51% Attacks

HDPoA eliminates the impact of the attack that is associated with controlling the majority of the network hash power, as it first relies on AN to manage and validate the mining process, and second the hash calculation is performed by unrelated worker nodes. For attackers to successfully control the network, they would need to control at least 51% of the network authority nodes, unlike PoW, where an attacker needs to control 51% of the network's hash power. HDPoA introduces the concept of distributing mining work amongst network nodes to deter such an attack. However, while it is still possible that an attacker can control 51% of the ANs, this is very difficult because it is time consuming as the attacker needs to either build its nodes honesty level until they

all are promoted to AN. This is done while other honest nodes building their honesty at the same time, meaning attacker may needs to increase the number of nodes at his disposal, hence, more time. Based on this the likelihood of this attack is *low* and the impact level will be very high.

4.4.4 Forking

As with the original PoA [153], it is possible that small forks can occur; however, this can be efficiently addressed. First similar to [153], HDPoA implements a mechanism that only allows any AN to propagate a block at every $\frac{N}{2} + 1$ blocks, thus, at any time t there are at most $N - (\frac{N}{2} + 1)$ ANs allowed to propagate a block. Second, other available ANs on the network will validate any new received block according to algorithm 4. Another important mechanism in place to prevent forking is that each block that is propagated by the selected primary node will always have a higher weight compared to the elected secondary. The secondary node always adds a waiting time before releasing a new block. All nodes, including the secondary, will always validate and add the block with the higher weight if they receive more than one block, as long as it was minted by one of the elected nodes. This makes the likelihood of this attack is *high* and the impact level is low.

4.4.5 DoS Attack

HDPoA eliminates this attack, which is associated with PoA. First, the scalable work method allows all nodes to increase their honesty's levels, and all can be promoted to the category of authority node. This means there will be more ANs on the network to allow for a more decentralised approach. Hence, more ANs will be available to run and manage the mining process. Second, the time interval between blocks is not fixed, as in the PoA approach; mining time is always changing and depends on the number of available WNs and the difficulty, making it difficult to determine when each authority node will manage the mining process. This makes the likelihood of this attack is *Low* and the impact level is High.

4.4.6 Spamming Signer (AN) Attack

One interesting attack on the traditional PoA is the spamming signer attack, in HDPoA case this is spamming AN attack. In PoA this attack happens when malicious signers (in HDPoA case AN) inject new vote proposals into each block it mines to try and change the authorised signers lists. In HDPoA this attack is very difficult to perform. First the honesty level and scalable work mechanisms were implemented that guarantee nodes have to spend a long time building their honesty level before being promoted to the AN category. Second, even after the node is promoted to the AN category it will be difficult to carry out this attack. This is because HDPoA only allows removal from the list through the voting process and the attacker needs to defeat the network

total voting weight to be able to control the process which is very difficult to do. This attack likelihood is *Low* and the impact if this attack was to be executed is *High*.

4.4.7 Sybil Attack

In peer-to-peer networks sybil attack is defined by [160] as an attacker that creates multiple identities (i.e blockchain node ids or accounts) to appear as legitimate nodes on the network. In HDPoA an attacker might join the network using multiple nodes and perform according to the network rules and build up each node's honesty level until it is promoted to the AN category. Over time the attackers may end up owning multiple ANs then will try to control the network. This is a possibility, however the design of HDPoA will rely on the cost of time and the fact its a public network where many other honest nodes will join and also build up honesty and be promoted to AN as well. The more nodes on the network the more difficult and time consuming this attack will be. This is because the attacker needs to wait for a longer time to first gain trust then be promoted to AN (there is no need for nodes to compete for hash power as the consensus and distribution of tasks among WN works in a round robin process). Secondly, other honest nodes at the same time will do the same making it difficult for the attacker to control the total voting weight or the block generation process on HDPoA. While such an attack could be costly; the level of impact is *Very High*, it is very difficult and time consuming to perform such an attack making its likelihood *Moderate*. ***A summary of the risk level of the previously analysed attacks can be found in Table. 4.3.***

Table 4.3: Possible attacks on HDPoA and their likelihood, level of impact and risk's level

Threat	**Likelihood**	**Level of Impact**	**Level of Risk**
Forging new block	Very High	Low	Moderate
Control voting process	Low	Very High	Moderate
Malicious WN Submitting wrong solution	Very High	Low	Moderate
Spamming AN	Low	High	Low
Forking	High	Low	Low
DoS.	Low	High	Low
51% attacks	Low	Very high	Moderate
Sybil attack	Moderate	Very High	High

4.5 Implementation and experiment

In house blockchain platform for the implementation and validation of HDPoA was purposely created, and HDPoA was deployed. In the main phase of the experiment, HDPoA was deployed over a network where all the nodes are connected to each other over Wi-Fi (Wi-Fi deployment). In a secondary phase of the experiment HDPoA was

deployed over a hybrid network with both Wi-Fi and LoRa connectivities were used (hybrid LoRa deployment). In the following sections both deployment will be discussed in details.

4.5.1 Wi-Fi deployment

Figure.4.2 shows a topology where all the nodes are connected in a peer-to-peer network over Wi-Fi. Most of the experiments and testing was performed using this topology. The presence of a more reliable communication link, such as Wi-Fi, allows for lower latency when propagating blocks and transactions, as well as the distribution of mining tasks among worker nodes. This resulted in a more stable network and fully synced nodes within the global chain.

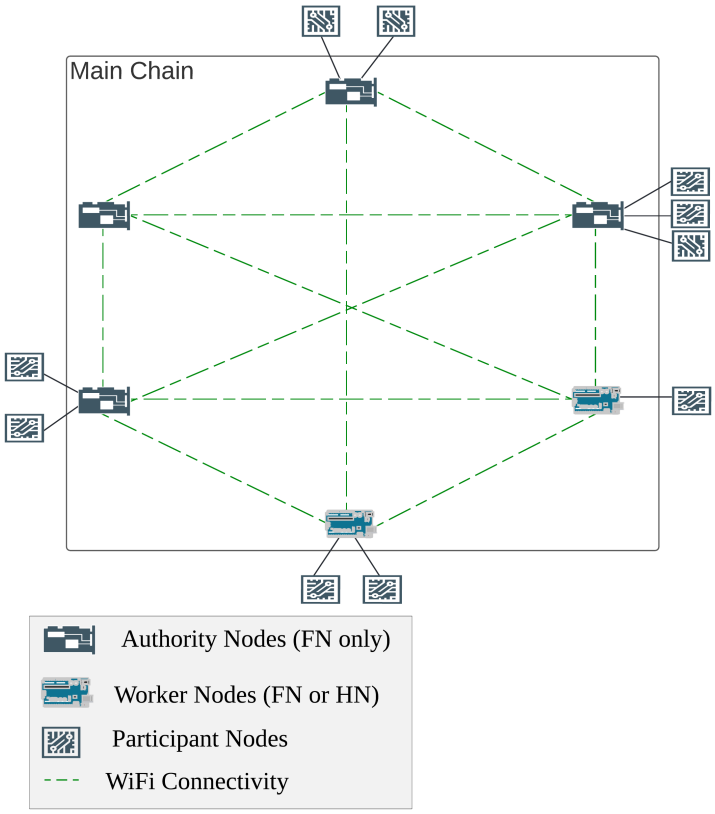


Figure 4.2: HDPoA Deployment over Wi-Fi only

Block Mining Process

The following steps explain the process of generating a new block, as illustrated in Fig.4.3:

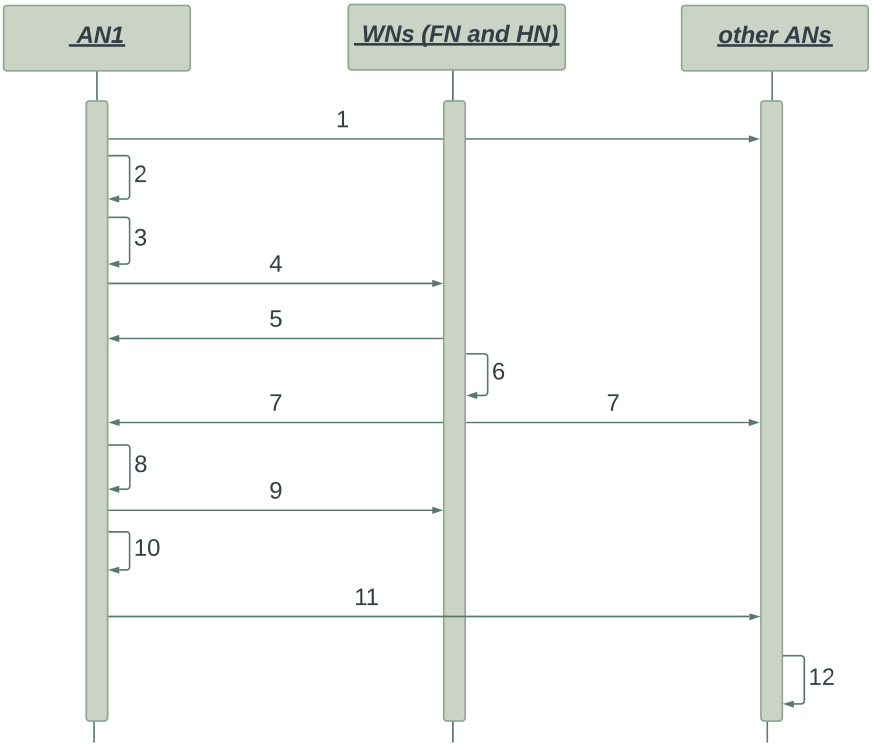


Figure 4.3: The different steps of the block mining process- Wi-Fi deployment.

1. The first authority nodes in the round-robin, *AN1*, will start the block generation process.

2. It will first collect and validate all transactions from the transaction's pool, and add them into a new pending block.

3. Then *AN1* will calculate MR for all added transactions and if needed it will adjust the difficulty D according to the target mining time and the available number of WNs [using (4.4)], the new value of D is then included in the block header and other ANs will use it for any future adjustment. The workload for each worker node will then be assigned according to (4.18).

4. *AN1* will distribute the mining tasks among the WNs.

5. Each WN will reply to *AN1* either by accepting the mining task or rejecting it.

6. The WN that accept the mining task will then start performing the search for the right nonce until either receives an abort message from *AN1*, or finishes the task at hand.

7. The worker node that finds the correct solution to the task (e.g WN1) will send its solution to AN1 and also to all other ANs for block validation in a later stage of the process.

8. When *AN1* receives the WN's solution, it will validate it by executing one hash calculation.

9. If the solution is valid, an abort request will be issue by *AN1* requesting other WNs to stop performing .

10. Then, *AN1* will create and sign the new block.

11. *AN1* will propagate the new block to all other authority nodes. AN1 will then issue a reward transaction to increase the $WN1$ honesty level by S_p, this transaction will be propagated to all ANs and WN1.

12. Finally, once other ANs receive the new block, they will first validate its hash using the previous block and the nonce received from the $WN1$. If the block and all transactions in it are valid, they will added to their local copy without further delay and accept the reward transactions for inclusion in the next block. If any AN thinks the block is not validate it will start a voting process and if the majorities of ANs (51%) agrees that the block is not valid then the block is rejected and the node that propagated the invalid block will be remove from the AN category.

4.5.2 Hybrid LoRa deployment

Deploying over LoRa will provides access to more hashing power by integrating LoRa nodes that do not have access to any Wi-Fi connectivity. However, this may cause an increase in the confirmation time as a result of distributing the mining tasks using less reliable LoRa connectivity compared to Wi-Fi. The network topology of this deployment is illustrated by Fig.4.4.

For a more stable network and lower latency, the prorogation of blocks and transactions were limited to only nodes that are connected by Wi-Fi, and only utilising the hash power of the LoRa side chains.

Block mining process

The process of generating a new block is slightly different than that of Wi-Fi deployment. These steps are illustrated by Fig.4.5 and they are as follow:

1. The first authority node in the round-robin, *AN1*, will start the mining process.

2. It will collect all transactions from the transaction pool, validate them, and add them into a new block. Then, it will calculate MR for all added transactions and

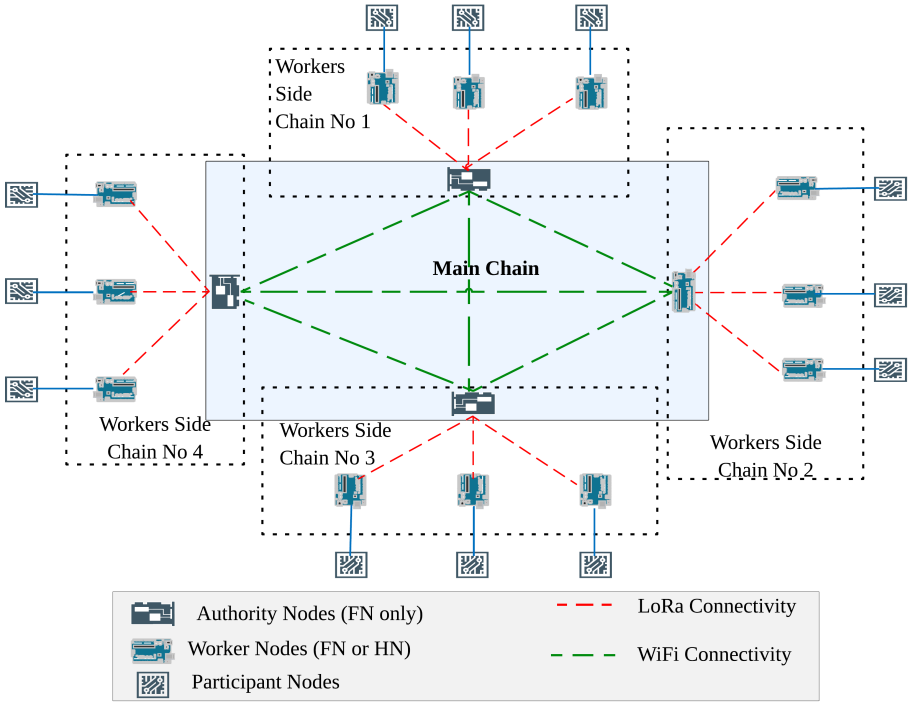


Figure 4.4: HDPoA deployment over hybrid Wi-Fi and LoRa

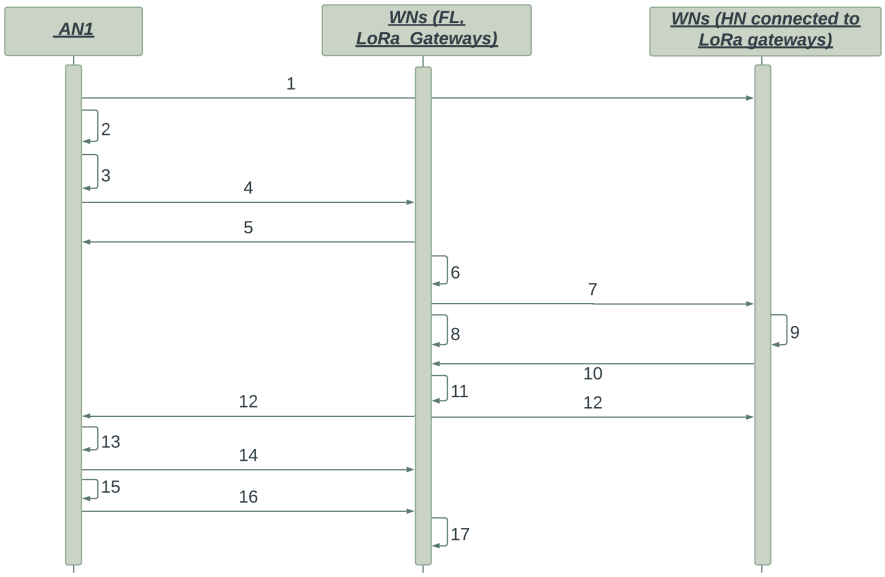


Figure 4.5: The different steps of the block mining process - hybrid LoRa deployment.

adjust the difficulty according to the network's requirements (i.e. mining time and the available number of WNs) [using (4.4)].

3. Using 4.18, it will then calculate the workload for each worker node.
4. *AN1* will then distribute the mining tasks among WN, in this case FNs that are acting as LoRa gateways.
5. Then, in return each WN will reply to *AN1* by accepting the mining task or rejecting it.
6. Gateways that have accepted the mining task will also adjust workload according to 4.18 and according to the number of connected WN (in this case they all could be HNs).
7. Then, they will distribute tasks to the connected nodes over the available LoRa link.
8. Gateways will then start performing their own mining tasks while also listening for any response from the WNs.
9. Each connected WN to the gateway that has accepted the task will start the mining process until it finds a solution, receives an abort message from its gateway, or completes the workload assigned to it.
10. If any gateway (or WN that are connected to a gateway in this case) finds the solution, it sends it to its own gateway.
11. When a gateway receives a solution, it will validate it.
12. If the solution is valid, it will forward it to *AN1* and also to all other ANs. Then the gateway will also send an abort message to WNs that are connected to it.
13. When *AN1* receives the solution, it will validate it by executing one hash calculation.
14. If the solution is valid, it will send an abort message to all assigned gateways and WNs.
15. Then, it create and sign a new block.
16. *AN1* will then propagate the new block to all other authority nodes.
17. Finally, when other authority nodes receive the new block, they will first validate its hash using the previous block and the nonce received from the WN. If the block is valid, they will also validate all transactions in it.

Table 4.4: Details of devices used in the experiment

Device	Specification	No & Roles
Raspberry Pi (R-Pi)	Raspberry Pi 3 Model B+ Broadcom BCM2837B0 Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz, 1GB LPDDR2 SDRAM	2 as AN 24 as WN
ESP32	ESP-32 Development Board Module - ESPRESSIF LORA32 868Mhz - wi-fi antenna -0.96 inch blue oled display- 20dBm output power -Clock Speed: 320 MHz up to 150 Mbps- 32-bit address space	4 as WN
ESP8266	Tensilica 32-bit RISC CPU Xtensa LX106 - Flash Memory: 4 MB- SRAM: 64 KB - Clock Speed: 80 MHz	1 used for evaluation and comparison
RF Transceiver Module RFM95W	LoRaTM Modem- 168 dB maximum link budget- Programmable bit rate up to 300 kbps	4 used forLoRa connectivity/

4.5.3 Network initialisation

When deploying HDPoA blockchain for the first time, all nodes will be joining as worker nodes with an honesty level H_i equal to zero. In order to allow the nodes to increase their honesty, the initial deployment requires the assigning of a few nodes as authority nodes. This means at the initial phase the AN category will be consider as private and the WN category will be consider as public. The network will run in this initial phase until it has sufficient nodes with an honesty level higher than the network's honesty threshold. Then, the initially assigned authority nodes will be removed from the network. This phase was simulated to discover how many blocks are needed to reach a level where the network has a sufficient number of nodes that managed to increase their honesty level through scalable work to allow them to be promoted to the AN category. The simulation was accomplished for different numbers of nodes, and the threshold was adjusted (using [4.19]) with a fixed mining time t_m of 10 minutes. Three different scenarios were simulated: 1) all nodes once they are promoted to the AN category are assumed to be honest and not compromised; 2) 10% of the promoted AN are assumed to be malicious or compromised; 3) 20% of the promoted AN are assumed to be malicious or compromised.

4.5.4 Experiment setup

A blockchain platform for the implementation and validation of HDPoA was designed and created. For full evaluation of HDPoA the platform was deployed and tested over a network where all the nodes are connected to each other via Wi-Fi. In the second phase of the experiment was conducted while the nodes were connected over Hybrid LoRa for some evaluations. The proof-of-concept permissionless blockchain system for testing and evaluating HDPoA uses a total of 30 different IoT devices, comprise of Raspberry Pis, ESP32, and ESP8266. Table. 4.4 shows the details of the devices used in this experiment. and Fig.4.6 shows picture of the devices during the experiment.

4.6 Results and Evaluation

In the following subsections the performance and security evaluation of HDPoA will be presented.

Figure 4.6: Pictures of the devices used in the experiment.

4.6.1 Initial deployment: latency to self-supported network

As described above, when deploying HDPoA blockchain network, an initial phase of deployment is required to allow nodes to build up their honesty level and the network becomes self-supported in terms of available authority nodes. A network can be describe as self-supported when there is an adequate number of authority nodes, based on the simulation this number is 5% of the total available nodes for a network with less than 8,000 nodes and 10% for a network with more than 8,000 nodes, at which half of the nodes have increased their honesty level to at least half of the threshold value. Figure.4.7c shows the required number of blocks needed for the network to be considered self-supported for different simulation scenarios; a network with 1,000 and a network with 10,000 nodes. It also shows the impact the malicious nodes could have on this initial phase if 10% of AN are malicious and if 20% of AN are malicious. Figure.4.7a and Fig.4.7b show the honesty level for all nodes at the end of the initial deployment phase for a network with 10,000 total nodes. It is clear from both figures at the end of this initial deployment more than half of the nodes have an honesty level that is more than half the required honesty threshold, this would allow the network after deployment to rapidly grows in terms of number of AN. For validations the required number of blocks was measured and compared with the simulated result for two networks; one with 10 nodes and another with 20 nodes, Fig.4.7d shows the measured and simulated results.

4.6.2 Hash Power

Each device's hash power (hash/sec) was measured, since this is an important parameter that can dictate both the mining time t_m and the difficulty D. Figure.4.8 shows the average hash power for three different IoT devices: the Raspberry Pi accomplished on average about 35 khash/sec, the averaged hash power the ESP32 can produce is

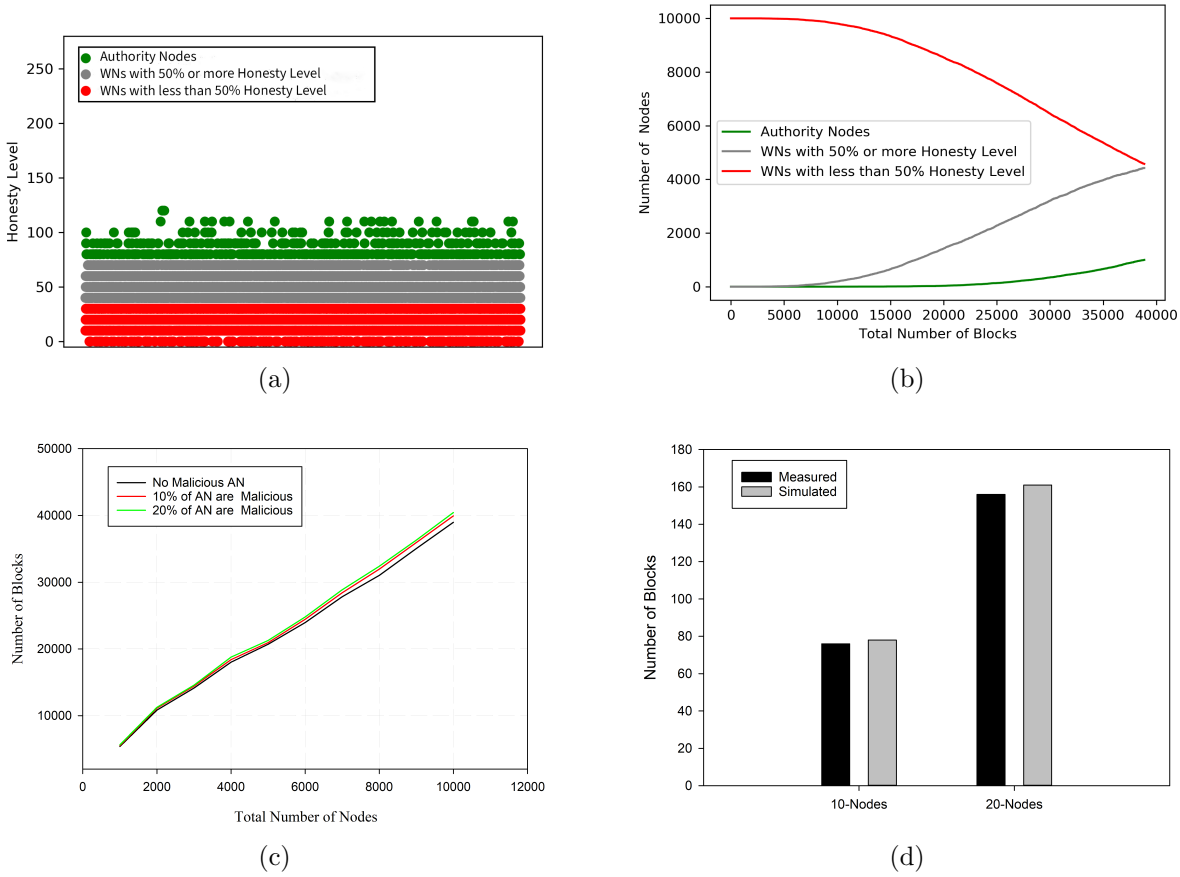


Figure 4.7: Self-supporting phase. (a) nodes' overall honesty level at the end of simulation (10,000 WNs). (b) Simulated total number of blocks requires to reach self-supported level (10,000 WNs). (c) Simulated required number of blocks for different networks configurations (1,000-10,000). (d) Measured total number of blocks for two networks 10 WNs and 20 WNs

about 17.5 khash/sec, and the ESP8266 averaged 5 khash/sec. These results show that with the right mechanism, all of these devices can be part of the mining process if their computational power is carefully managed so that there is no substantial effect on their power sources or batteries.

4.6.3 Mining time

The mining time depends on the available number of WNs and their hash power, the more devices available the less the t_m is, this allows for increasing the difficulty. Figure.4.9 shows the mining time for two difficulties ($D = 4$ and $D = 8$) using different number of devices, between 4-26 when testing the Wi-Fi deployment of HDPoA. The mining time can be seen decreasing for both difficulties as more WN participate in the block generation process until a network with 26 WN achieved a t_m of around 2

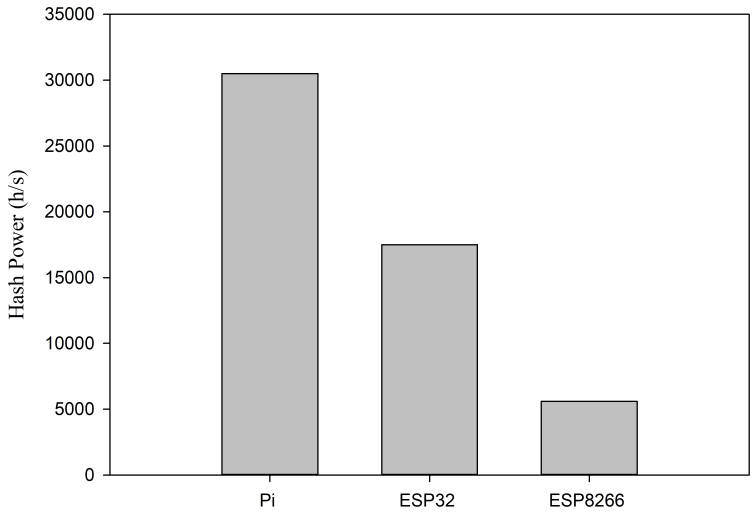


Figure 4.8: Average devices' hash power.

and 4 min for $D = 4$ and $D = 8$ respectively. It is clear that HDPoA has the ability to increase the difficulty according to the number of WNs, thus ensuring an added security through the adjustable difficulty.

HDPoA was also tested when deployed over Wi-Fi and LoRa (Hybrid LoRa Deployment) for a difficulty D = 1 and with total number of WN of up to 6 devices. Figure.4.10 shows the mining time when testing the hybrid LoRa compare to both the predicted time and the measured time when testing the Wi-Fi deployment. The hybrid LoRa deployment resulted in slight delay (an average of 700 ms round trip) caused by

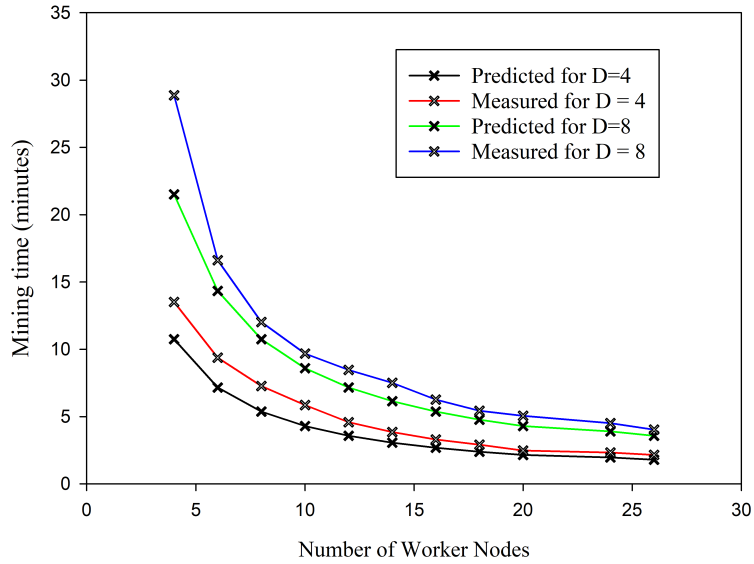


Figure 4.9: Measured and predicted T_m using different number of WNs.

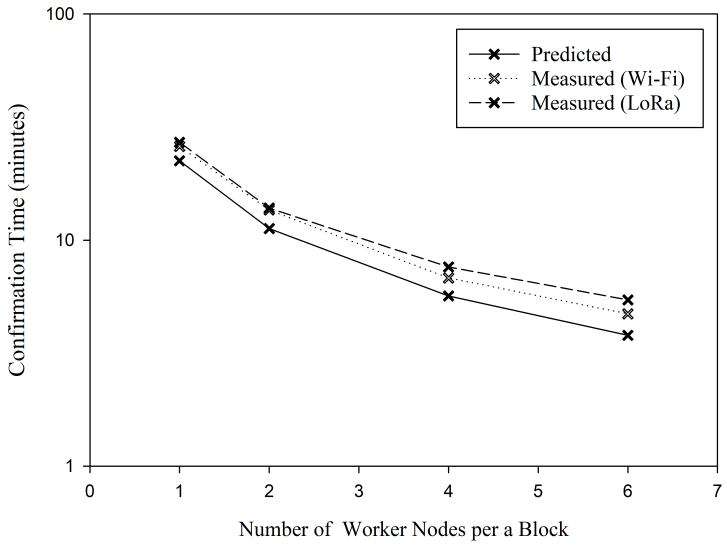


Figure 4.10: Mining time when testing the hybrid LoRa deployment compared to Wi-Fi deployment and the predicated time

the work distributions over LoRa link. However, this increase is acceptable as the expansion of the deployment over LoRa can open the door to access more computation power by utilising these small devices located behind LoRa gateways.

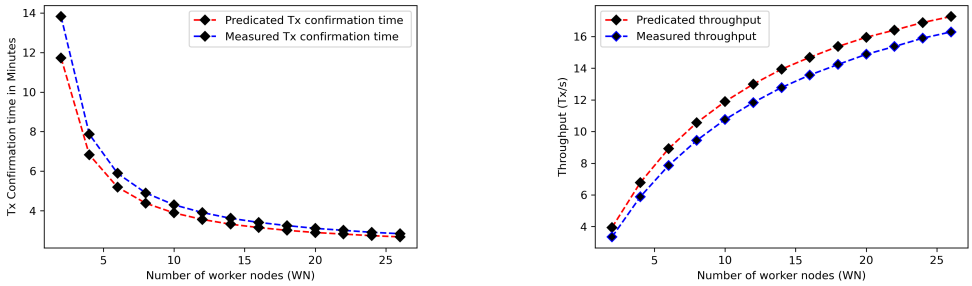
4.6.4 Confirmation time and throughput

The size of the block along with the transaction confirmation time dictate the throughput of HDPoA. According to the authors of [161]; who intensively investigated the impact of the block's size of on IoT blockchain systems; a size of 1 MB or less is highly recommended for IoT blockchain systems. In this work a max of 500 kB block was used, this is to enhance energy's efficiency and ensure limited impact on nodes' synchronisation. This allowed the increase of the number of WNs, which resulted in less mining time (i.e less Tx_{ct}) .

Tests to measure the Tx_{ct} and throughput were carried out using a block sized 500 kB while varying the number of WNs participating in the block generation process (2–26 WNs), the results are presented in Fig.4.11. From the figure it is clear that as the number of WNs increase (i.e network grows in number of nodes) the Tx_{ct} decreased resulting in higher throughput.

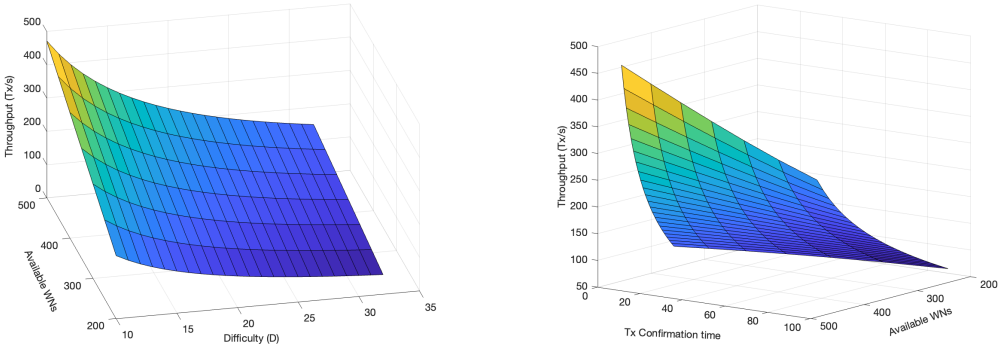
Due to the limited number of devices at hand, the impact of the number of nodes on all of Tx_{ct}, Throughput, and D was investigated and predicted (using [4.8 and 4.9]). The predicted results of Tx_{ct}, Throughput, and D are shown in Fig.4.12. From the figure, increasing the number of WNs that are participating in the mining process can have a positive impact on the network in terms of increasing the Throughput while

maintaining a good level of security by increasing D. Thus, showing how flexible and scalable the proposed HDPoA can be when deployed in public blockchain platform.



(a) (b)

Figure 4.11: Measured and predicted (a) Transaction's confirmation time (b) Throughput



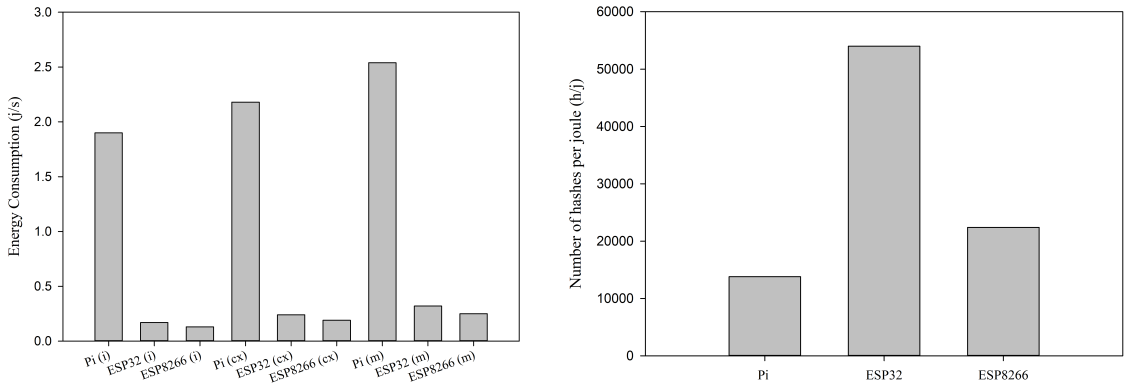
(a) (b)

Figure 4.12: Calculated throughput for different network setups. a) The number of WNs, difficulty, and throughput. b) The number of WNs, transaction's confirmation time, and throughput

4.6.5 Energy consumption and battery life

HDPoA was designed with the intention of ensuring that the mining process does not have a substantial impact on the power usage of the devices. The energy consumption for all three devices was measured, and Fig.4.13a shows the average energy consumption for these devices. First, when they are idle (i), then, when they are only connected to the Wi-Fi (cx), and finally, when they are performing the mining task (m). Most of the impact on the power was a result of only running the operating systems of the node (i). The power consumed by running the mining process on the devices was relatively small as can be seen in Fig.4.11.

Another important measurement is the number of hash calculations that a node can perform per a single joule. Figure.4.13b shows these number of hashes per joule for each devices. A Raspberry Pi will consume one joule of energy to produce on average a total of up to 13,800 hashes, similarly ESP32 can produces up to 54,000 hashes/j and the ESP8266 can achieve up to 22,400 hashes/j. It is clear from this that in terms of power small devices such as the ESP32 can be more efficient when performing such small mining tasks in comparison to more resources reach devices such as the Raspberry Pi, which is able to perform faster operation at small expense of energy.



(a) (b)

Figure 4.13: Energy and hash power measurements. (a) The average energy consumption for each device in different system sates. (b) The average number of hashes per joule each devices can produces

Battery Life. Table. 4.5 shows the estimated [using (4.11)] battery life (B_L) for a small button battery and a Li-Polymer battery with the minimum 30 days as a battery life target, for different network implementations using different number of WNs. It is clear from the table that devices with such batteries can participate in the mining process without any substantial impact on their batteries' life. In a network with more than 1000 nodes, a node may spend a month without being assigned a mining job. As shown in Fig.4.8 small IoT devices that can be run using this battery such as ESP-32 is more efficient in terms of power.

Figure.4.14 shows the battery life in days for different battery capacities (200 up to 3000 mA) when used in mining blocks in different network setups. This setup includes four different difficulties (1, 8, 16, 32) and different available WNs that ranges from 100 WNs up to 10,000 WNs with an assumption that blocks are mined within 10 minutes of each other $t_m = 10minutes$. The figure shows that even with small network at lower difficulty (D =1) batteries with small capacity (i.e 200 mA) can still participate in the network for couple of weeks without the need to change their batteries. However, as the network grow up in terms of the number of WNs the difficulty can be increased with minimal impact on devices' batteries, in network with 4000 WNs difficulty can be increased up to 32 and device's battery can be utilised for more than two months .

Table 4.5: Battery life analysis for a small button battery for different networks with different number of WN

Total Available WNs		**300**	**500**	**1000**	**5000**	**10000**
RS PRO CR2032 Button battery **BC= 225mAh** **V=3** **Price = £1.01**	Max-BL in days	68	110	225	1110	1830
	Minimum difficulty	1	1	1	1	1.2
	Mining tasks per month	3.8	2.3	1.1	0.2	0.14
	Min-BL in days	30	30	30	30	30
	Maximum difficulty	2.3	3.8	7.6	38	76
	Mining tasks per month	8.3	8.3	8.3	8.3	8.3
RS Li-Polymer battery **BC= 2000 mAh** **V = 3.7** **Price = £15.20**	Max-BL in days	195	930	1830	1830	1830
	Minimum difficulty	1	1	1	5.5	11
	Mining tasks per month	11.3	2.4	1.2	1.2	1.2
	Min-BL in days	30	30	30	30	30
	Maximum difficulty	6.8	33.8	67	337.5	675
	Mining tasks per month	73.5	73.5	73.5	73.5	73.5

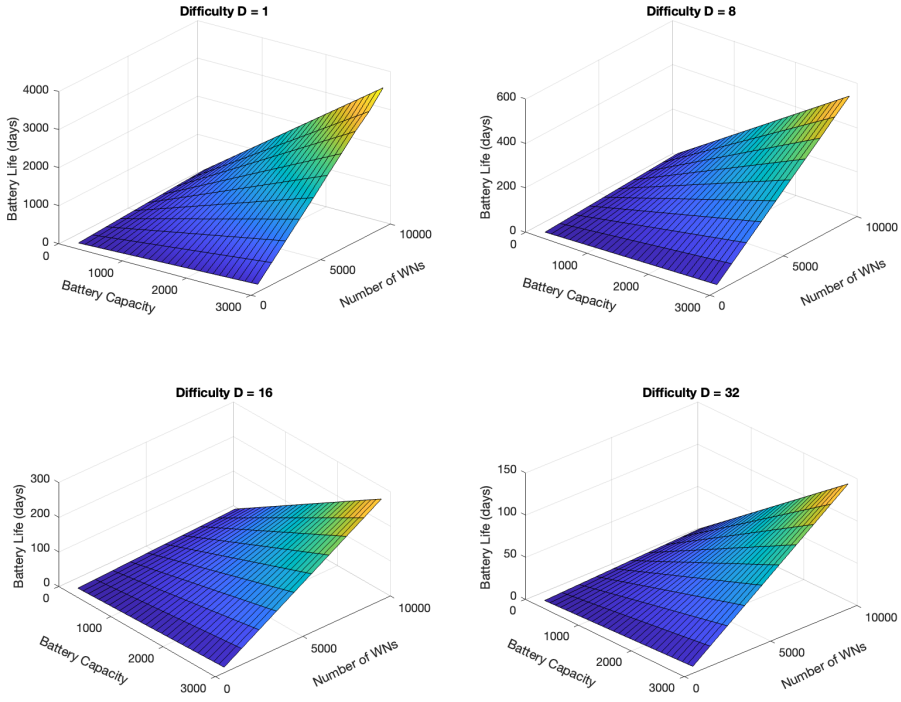


Figure 4.14: Battery life in days for different battery's capacities in different network's setups.

4.6.6 Security evaluation

In order for an attacker to successfully forge a new block or control the removal process of any AN from the AN category it should have total control over the voting process. The probability of any attacker taking control of the voting process was simulated using [4.20 and 4.21]. Figure.4.15 shows the maximum attack surface that can be exploited by any individual AN for five different networks each with a different number of ANs. It is clear from the figure that no individual node will be able to control the total voting weight. Making HDPoA a public consensus by allowing nodes to join and build up their honesty and be able to be promoted to the AN category is a source of strength on the network. As can be seen in Fig.4.15 the network with higher number of AN does realise the full potential of decentralisation making it difficult for any individual nodes to control.

A malicious user may intend to control the network by controlling more than half of AN (i.e 51%) through honest works. This requires the user or the entity to create enough full nodes that can participate in the mining process honestly for long periods of time until all of these nodes are promoted to the AN category and form the majority of them. This is very difficult and time consuming as HDPoA is an open and public consensus making it easy for nodes to join and build honesty level, thus malicious entities or users need to adjust the number of nodes at hand to make sure they have full control. The required number of blocks to be mined before a user can control HDPoA was simulated based on different network configurations in terms of available number of WNs (ranging from 1000-10000). No new node can join(i.e number of WN is not changing), and only 10% of WNs are FN were the assumptions. Figure.4.16 shows the required number of blocks before attacker's nodes form the majority on the AN category. Assuming the required mining time is 10 minutes, the times the attacker need to spend on this attack can be very long. For examples, a network with 500 nodes, the attacker needs to wait for more than a year to gain full control and this time is more than 2 years for a network with 10000 nodes .

4.6.7 Discussion

The results of the energy consumption tests successfully demonstrated that low-cost IoT devices are able to participate in the block generation process by carrying a small amount of the mining task without substantial impact on their batteries. This is a result of the way HDPoA was designed where it introduces an added layer of security through the trusted authority nodes allowing for the distribution of the mining computation among different trusted and less trusted WNs. HDPoA is a premissionless consensus where any node can freely join the network. This means more WNs are available to participate in the block mining process which in turn, as the network grows in size to may be more than 1,000 nodes, can imply a WN might spends weeks without performing any mining task (hence, less impact on the battery).

In HDPoA, transactions confirmation only requires one block to ensures the ***trans-***

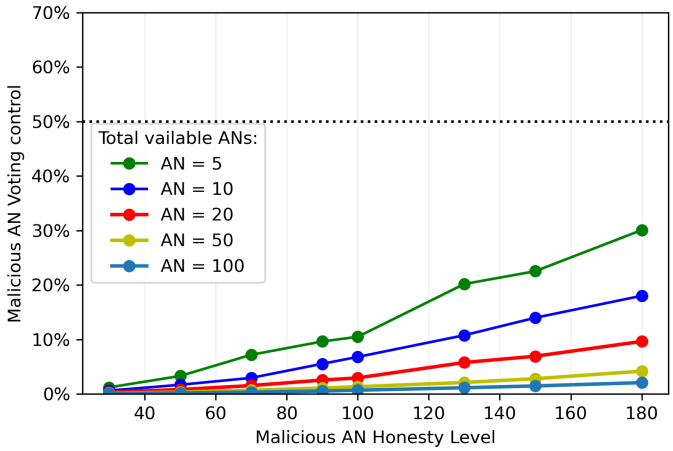


Figure 4.15: The attacker's maximum attack surface as a percentage of voting weight that it can control (need at least 51%)

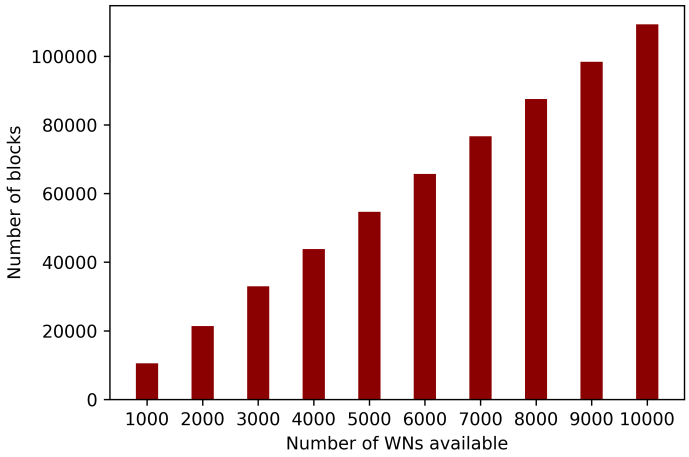


Figure 4.16: Number of blocks needed to control HDPoA through building honesty

action finality, providing lower latency for IoT applications on the network compared to the approximately six blocks required in the PoW approach for bitcoin. This is because HDPoA integrates PoA which does not require more than one block for confirmation [156, 160], hence, mitigating the PoW long confirmation time. In terms of throughput bitcoin which utilises PoW as a consensus protocol has an average of 5-7 Tx/s [162]. PoA is totally controlled by the network owner and can implement a block period of as low as one seconds, this means using B_{size} of 1Mbyte and Tx_{size} of 200 byte can result in throughput of more than 5000 Tx/s, however in an IoT network where devices have limited computation resources nodes will spend more than 100 seconds or

even a few minutes importing a block [156]. This means this throughput will not be achieved in practice.

In the HDPoA, when running a mining task for 82.9 sec on a Raspberry Pi, the total energy cost was 43.9 J and for the ESP32 it was 12.4 J, this compares to 54.9 J energy cost measured by [163] when executing PoW on a Raspberry Pi. This indicates that the proposed consensus mechanism of HDPoA is more efficient than PoW in terms of energy consumption, and that nodes are able to participate in the mining process while also being used for data analysis and sharing without substantial power cost. As the number of participating nodes increases, the effect on their individual power will decrease. In a network with a few thousand IoT devices, a node might spend a day without executing tasks.

Difficulty. Some previous studies, such as [69, 71, 149], have integrated PoW into IoT-blockchain systems by implementing the concept of nodes with high trust or credit scores mining blocks at *low difficulty*. In HDPoA, as the number of worker nodes increases, the difficulty can be increased, allowing for incorporation of the full security advantages provided by PoW.

Table 4.6 provides a performance comparison between HDPoA, PoW, and, PoA. As can been seen in the table, HDPoA scores better or similar (i.e top score) in all of the performance metrics used. This is due to the fact that it was designed to inherit

Table 4.6: Performance comparison between HDPoA, PoA, and PoW

	PoW	**PoA**	**HDPoA**
Tx Confirmation time	Require at least 6 blocks PS = 1	only one block as Blocks are propagated by trusted signer PS = 3	Similar to PoA as blocks are propagated by AN. PS = 3
Vulnerabilities	51% attack associated with controlling hashing power PS = 2	DoS and 51% can be easily mounted against the authorised signer due to their limited number and the fixed block period. PS = 1	Moderate risk to sybil attack which is an inheritance vulnerability from the IoT. However it is difficult and time consuming to mount such an attack PS = 2
Throughput (Tx/s)	7 -10 Tx/s PS = 1	It is based on the block period which can be as slow as 1s and the size of the block. This can results in higher throughput compare to most consensus protocols PS = 3	Can be more than 900 Tx/s in a network with 1000 nodes. PS = 2
Decentralisation realisation	Fully decentralised PS = 3	More of a centralised network as it is only suitable for permission and private network PS = 0	Fully decentralised PS = 3
Impact on IoT device's battery	Can be very costly. Not suitable for IoT devices PS = 0	Very low and limited PS = 1	The work is divided amongst devices. A device can spend days without work, meaning the impact is limited on its battery. PS = 2
Performance Total Score	**8**	**8**	**12**
PS = Performance Score. For each category the score is assigned between 0 to 3 where 0 is the worst in class and 3 best in class.			

the strength of both PoA and PoW and eliminate or effectively reduce the weaknesses which are resulted from the characteristics of any one of PoW and PoA.

4.7 Summary

In summary, a secure consensus mechanism based on the PoA and PoW algorithms, called honesty-based proof of authority (HDPoA) via scalable work was designed. It was found to be suitable for implementation within IoT-Blockchain applications and was implemented and validated. HDPoA was validated through real-world implementation using a dedicated testbed utilising different types of low-cost and low-power IoT devices with varying capabilities. The energy consumption and hash power measurements of the devices showed that HDPoA can utilise the hash power of these devices without a substantial impact on the life of their batteries.

The security analysis of HDPoA demonstrated its resilience and ability to defend the network against some of the common attacks on blockchain systems. HDPoA design approach also helps in reducing the number of confirmation blocks required to ensure transaction finality for the network.

Chapter 5

Blockchain in IoT: Supporting DAI Implementation and Ensuring Data Integrity

This chapter presents a distributed and decentralised architecture for the implementation of Distributed Artificial Intelligence (DAI) using hardware platforms provided by the Internet of Things (IoT). The proposed architecture was analysed, implemented, and tested using a dedicated testbed with low-cost IoT devices. Security analysis, and a quantitative measurement and performance evaluation of the system was conducted.

5.1 Introduction

The IoT provides an ideal distributed hardware and software platform that can be utilised for the implementation of a DAI engine. Another fact about the IoT is that it is a major source of big data, which is generated from the huge number of smart devices connected to the internet. Deploying an AI engine in a distributed approach over this platform would allow users to be able to generate valuable information and knowledge and produce an actionable outcome in close proximity to the data sources. This would enhance the overall performance of applications as they would avoid the long delay as a result of the utilisation of the centralised cloud entities.

5.1.1 Problem statement and background

A distributed system may be a collection of autonomous nodes communicating with each other over a communication channel. It has the ability to run software in parallel among these nodes closer to where computing is needed [19]. This attractive ability of the distributed approach helps process data in near real-time and reduces the communication overhead needed to transfer data from end devices to a central entity, such as cloud computing. To realise the benefits of true parallelism and distribution offered by

AI in a fully distributed computing system, a scalable hardware platform is required. The distributed nature of the IoT, where thousands of smart devices are available and can communicate with each other, offers such a platform [18].

Distributed computing requires each node in the system to carry out a parallel computation every round [20]. The number of rounds needed to complete the task and the number of messages exchanged between the nodes will result in a complex and undesirable situation. To avoid this complexity and reduce latency by providing the AI system with historical data to facilitate future decisions, the IoT system requires implementation in an architecture that combines both decentralisation and distribution. Blockchain technology is an ideal solution that enables distributed computing and achieves data storage in a large number of devices over a wide area network.

The integration of blockchain into IoT can provide reliable control of the IoT network's ability to distribute computation over a large number of devices [21]. It also allows the DAI system to use trusted data for analyses and forecasts while utilising the available IoT hardware to coordinate the execution of tasks in parallel, using a fully distributed approach.

The authors of [164] introduced a distributed AI system enabled by multiple layers of fog networking for smart shopping advertisements. They offloaded some of the AI analyses and data processing to fog layers while utilising cloud platforms to perform the main analysis and choose advertisements based on age and gender.

The works in [165, 166, 167] provided a framework and a simulation study to deploy a distributed Hopfield neural network through the use of a Wireless Sensor Network (WSN) as a hardware platform. While this work provided a robust architecture for utilising an IoT system for the implementation of DAI, validation in the form of practical deployment and a real-world use case is needed. The authors of [168] have proposed distributed deep neural networks (DDNNs) architecture, which consists of the cloud, the edge, and IoT end devices. Another work by [169] proposed a distributed machine learning (ML) architecture called Parallel-Channel Artificial Neural Networks (PCANN) for image recognition tasks on IoT devices.

Different works have proposed the combining of blockchain, AI, and IoT into one system. For example the NeuRoNt platform which is based on the Ethereum blockchain and an edge layer and consists of multiple agents powered by smart contracts for solving complex problems [132]. Using blockchain and AI the authors of [133] and [143] proposed solutions for IoT-enabled smart cities applications. ModelChain, proposed by [141], uses blockchain and machine learning for healthcare applications. BlockDeepNet is a framework that uses deep learning, blockchain, and smart contracts for data analysis in IoT [170], similarly the work by [142] proposed the DeepCoin framework for smart grids. Details analyses of these related works are provided in 2.6.

Although each of these proposed frameworks and architectures provide different advantages, none has yet exploited the potential provided by blockchain technology for supporting and facilitating the implementation of AI in a decentralised and fully distributed approach through IoT systems. Table 5.1 shows a comparison between the architecture proposed in this work and related research.

Table 5.1: Comparison between this proposed Architecture and other related works

Work	Main Contribution	Technologies Utilised	Distributed and Decentralised Implementation	Prototype and System Deployment	Measurement and Performance Analyses	Distributed Neurons Implementation on IoT devices
[133]	Blockchain and smart contract based framework for sharing economy in smart cities.	Fog, IoT, AI, Blockchain and cloud	No, blockchain for distributed data sharing but relies on Central AI engine	Yes, using smart phone, private blockchain and Amazon AWS	Yes, system's latency,	No
[141]	Blockchain-based framework to improve the robustness and security of distributed healthcare predictive modelling	Blockchain and Machine learning	No	No	No	No
[170]	Blockchain based Deep Learning collaborative algorithms for IoT.	Blockchain, Smart Contract, Edge.IoT, and AI	Partially, IoT devices relies on central edge server	Yes, using edge cloud, cluster, access point, Ethereum blockchain and smart contracts	Yes, Accuracy, Latency and memory and CPU Usage	No
[142]	Framework for smart . grids based on deep learning (as IDS) and blockchain	Blockchain and Recurrent neural Networks (RNNs)	No, uses Central RNN	Yes, using private blockchain	Yes , IDS detection rate and accuracy	No
[143]	Framework for enabling high performance and cost-effective computing resources for smart city applications.	Blockchain, Deep Learning, and IoT	No, central cloud for AI	Partial implementation, Corda and CordaDApp to simulate blockchain nodes setups	Yes, Latency and Scalability	No
[164]	AI-based smart shopping advertisements.	AI, Fog, and cloud computing	Partially, still relies on central cloud for AI implementation	Yes, using OM2M IoT platform and central cloud	Yes, latency and Data transfer	No
[165, 166, 167]	framework to deploy a distributed Hopfield neural network using WSN	AI and WSN	Distributed AI	No, simulation only	Simulated results only.	Yes
[168]	distributed deep neural networks (DDNNs) architecture which consists of the cloud, the edge and IoT end-devices.	AI, Cloud, Edge and IoT	Partially, needs aggregator and cloud for some processing	Yes, using six IoT devices and a cloud	Yes , Measured Accuracy	No
[169]	machine learning (ML) architecture called Parallel Channel Artificial Neural Networks (PCANN) for image recognition on IoT devices	AI and IoT	Partially, IoT devices relays on a controller	No, simulation only	Yes, Classification Accuracy	No
This work	A novel and secure blockchain architecture for supporting fully distributed AI on low-power and low-cost IoT devices.	Blockchain ,AI, Edge and IoT-end devices	Decentralised and fully distributed AI based on blockchain	Yes, using customised built blockchain that includes IoT-centric consensus mechanism and DAI over IoT devices (23 Raspberry Pis and six ESP32)	Yes, Latency, Accuracy, Devices Hash Power and Energy Consumption	Yes

5.1.2 Contribution of the chapter

The contributions of this chapter are summarised as follows:

- A novel, and secure blockchain architecture for supporting DAI on low-power and low-cost IoT devices.
- Practical implementation of DAI using scalable and distributed IoT hardware platform.
- Prediction and measurements of DAI using blockchain in IoT devices with performance analysis that includes, accuracy, energy consumption, and overall system latency utilizing data from trusted and robust platforms.
- A blockchain protocol that includes new transaction and block formats that help nodes handle DAI-based transactions and prediction requests.

Published papers

Based on this chapter a paper was published:

1. S. Alrubei, E. Ball and J. Rigelsford, "The Use of Blockchain to Support Distributed AI Implementation in IoT Systems," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2021.3064176.

5.1.3 Organisation of the chapter

The rest of the chapter is organised as follows. Section 5.2 presents the proposed architecture design. Section 5.3 presents the system analysis; this is followed by the system implementation and deployment in Section 5.4. Details of the results are in section 5.5, and finally, the summary of the chapter is presented in section 5.6.

5.2 Proposed architecture design

Artificial neural networks are a branch of artificial intelligence, and a multilayer perceptron (MLP) is a type of neural network. An MLP is usually made up of at least one input layer, one hidden layer, and an output layer [171]. The number of neurons in the input layer are equal to the features of the dataset, while the number in the hidden and output layers can vary depending on multiple factors, such as training and the type of implementation and problem at hand (e.g., regression or classification). Layers are fully connected, and neurons communicate their values to each other using synaptic connections represented by weights [171].

MLP is based on supervised learning techniques, and one of the learning algorithms that is used to train MLPs is the back-propagation algorithm. In this algorithm, the

first step is to initialise network weights to random values, then present the first input values from a training dataset to the network. This data is propagated through the network, and each node produces an output that is a function of the sum of the input values to the node and its weights, modified by an activation function, such as the sigmoid function ($S(x) = 1/1 + e^{-x}$). This is done by each neuron in the network until a final output is produced, then an error is calculated that compares the actual output to the target output. This error propagates back through the network and weights are adjusted to minimise the overall error. These steps are repeated until the overall error is satisfactorily small.

5.2.1 Design overview

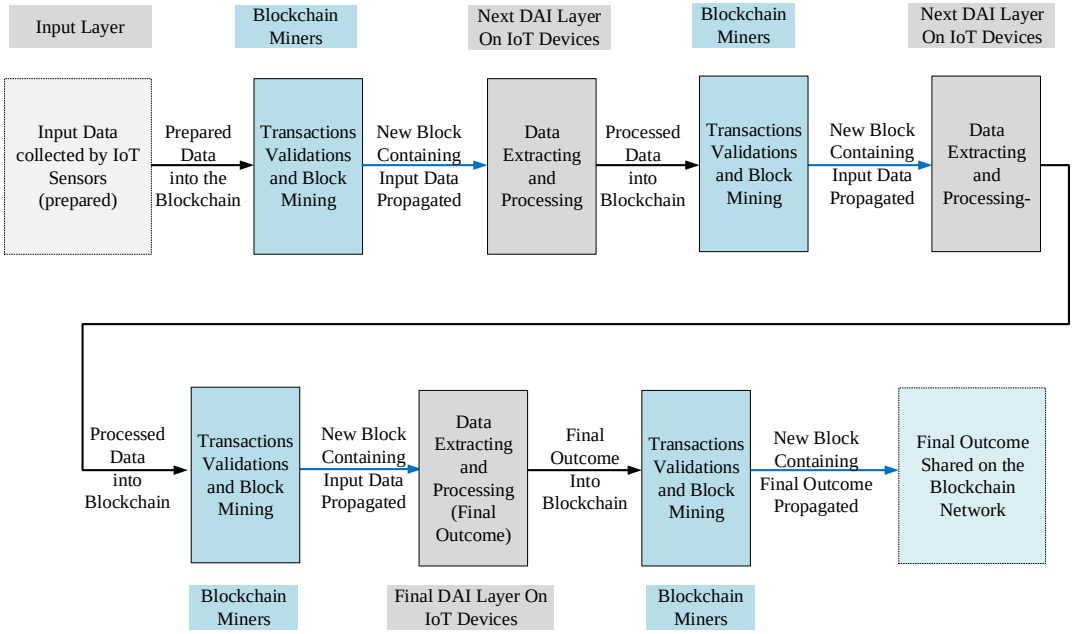


Figure 5.1: Proposed architecture - general workflow.

Distributed artificial intelligence (DAI) is an approach to exploit the resources provided by large-scale distributed computing. The aim of this design is to develop a blockchain platform that can support the implementation of a DAI that utilises the capabilities of IoT systems. The general workflow of the proposed architecture is presented in Fig.5.1. In this design, with the integration of a blockchain that is trustworthy, self-managed, and self-regulated, the DAI engine will have a platform that provides a secure way to handle and protect data, yielding a better decision-making process. Regardless of the type of DAI implemented, the data flow will be the same. Devices and nodes will also be able to interact and communicate with each other in a

secure way that will ensure efficient processing and flow of data through the system's different layers.

While the architecture has the ability to support any form of DAI implementation, for this work the implemented DAI is in the form of distributed multilayer perceptrons (DMLP). The main idea is to build an architecture that supports DAI implementation in general, and DMLP was selected as an example because it will allow deployment up to the neuron level, allowing for more focus on the evaluation of the blockchain aspects of this architecture. For DMLP, the exploitation of the resources provided by large-scale distributed IoT systems can be achieve by hosting one or more neurons on an IoT device. Each device would then act independently and utilize the blockchain platform to ensure the integrity of the processed data and its transfer to other devices. Processed data will then flow from one layer to another until the desired outcome is achieved. A trained DMLP will be implemented and tested over this blockchain platform.

Blockchain platforms are usually built around one of two main approaches: *On-Chain and Off-Chain*. On-chain is a transparent approach where transactions are executed and stored on the public blockchain. It is implemented by most of the well-known blockchain platforms, such as Bitcoin [13] and Ethereum [172]. This means all transactions and communications between the nodes are executed on the blockchain, and each transaction is stored on the chain. While this approach can result in increased latency, it provides a trusted method that makes AI predictions traceable and easy to understand, allowing users and organisations to determine how and why any decisions were made.

Off-chain, as described by [173], is intended to move some of the computational efforts from the main chain to an off-chain platform. The transactions are executed by the nodes off-chain, and only the final outcome is committed to the main chain. There are different implementations of the off-chain methods, e.g., Bitcoin's Lightning network [174] and Plasma of Ethereum [175]. While this approach reduces latency, it does not provide the complete trusted and transparent process intended by blockchain nor does it allow for full traceability. It may not provide validations for all transactions, which could compromise the system security.

In this design, the on-chain approach has been chosen where all transactions are executed, validated, and committed on the main chain. This method allows the blockchain platform to record all the AI transactions and variables that are used by the trained AI engine to make decisions. As the AI engine becomes smarter as the result of continuous training and is able to process large amounts of data, it becomes more difficult for scientists to understand how the AI systems came to specific conclusions and decisions. However, through the implementation of AI on blockchain platforms, they will have immutable records of all the data and variables used by AI for its decision-making processes. This will provide data scientists with the ability to easily audit and trace the entire process.

5.2.2 System components

The system proposed in this designed is a decentralised and fully distributed architecture, which provides added value to the computational ability of an AI system through the utilisation of a scalable IoT-hardware platform. This architecture takes advantage of the IoT sensing capabilities, blockchain immutability and trustworthiness, and the capabilities of DAI intelligence. This results in the design of a computationally intelligent, scalable, distributed, and decentralised DMLP architecture based on the blockchain (see Fig.5.2). The architecture consists of five main components.

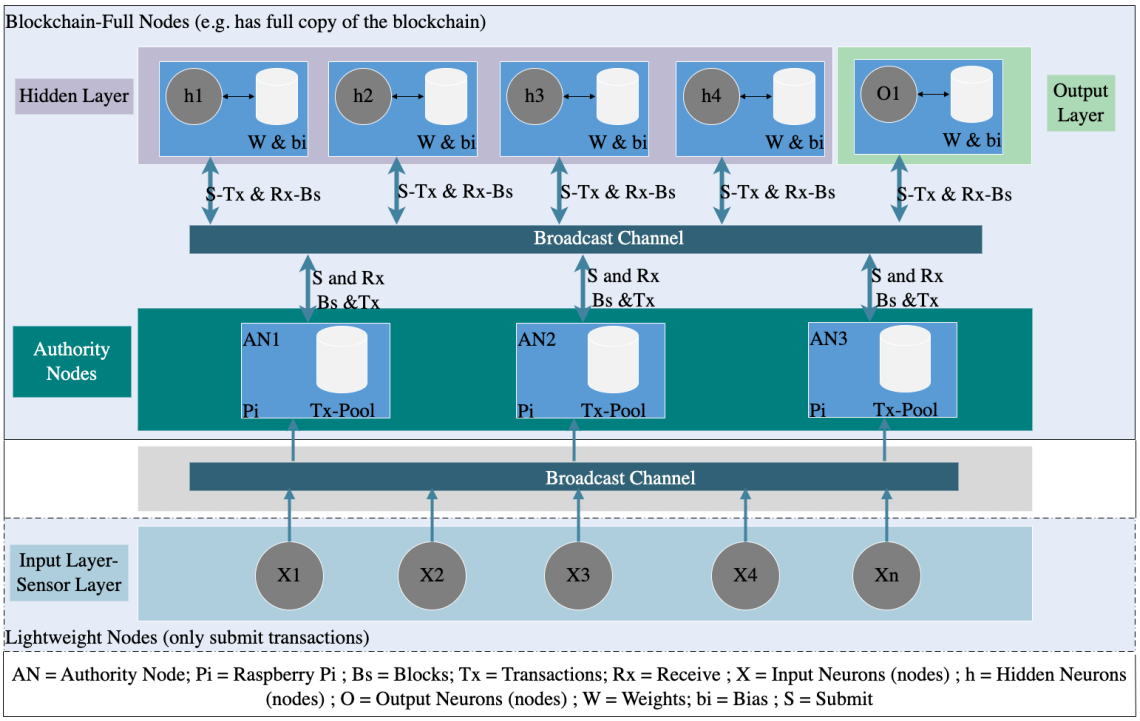


Figure 5.2: Proposed architecture - system components.

- *Input layer / Sensing layer*: This is where many small, low-power sensors can be used for monitoring and data collection. These sensors can be directly connected to other devices from the WNs or ANs. They can sense and collect data, which will then be locally prepared and submitted to the network for processing. These sensors are part of the blockchain network in the form of lightweight nodes that can submit transactions through the nodes they are connected to. They are the data feeder to the DMLP engine and the first layer in it.

- *Hidden layer*: These are low-cost devices, that will act individually as one or more neuron in the hidden layer. In this proposed architecture, there can be more than one hidden layer, but for the proof of concept implementation, only one hidden layer was deployed where each device acts as one neuron.

- *Output layer*: This is the final layer of the DMLP, and it also consists of low-cost devices. Each device can act as one or more neuron of the output layer. This design allows for the implementation of multiple output layers, where each one implements different activation functions and produces its own final predictions. This enhances the forecasting ability and allows for better performance.

- HDPoA consensus mechanism that was introduced and discussed in details in chapter 4 was utilised to secure the blockchain platform. Nodes in HDPoA are divided into two categories based on their role in the block generation process:

  - *Worker nodes (WN)*: These are low-cost and low-power devices, such as the Raspberry Pi and ESP32 microcontrollers. Within this system all nodes apart from the AN that currently manage the mining process will need to act as WN regardless of the value of their honesty level.
  - *Authority nodes (AN)*: These are low-cost devices, such as the Raspberry Pi, that form the heart of the blockchain network by acting as miners and validators. They store a full copy of the blockchain locally that is then synced with the latest block in the network. They have a high enough honesty level to act as coordinator of the mining process, validate the work of workers, sign and propagate blocks, and validate each other's newly propagated blocks. Authority nodes are the only nodes with a honesty level that allows them to run a blockchain network and handle all the tasks related to the DMLP engine in the network, which adds further trust to the handling of the data and prediction operations. More details about AN and WN were provided in 4.2.

Another important part of this architecture is the propagation of transactions and blocks. Broadcast is the most commonly used network operation in blockchain networks, where nodes issue transactions and blocks by broadcasting them in the network. In this architecture, assuming that all transactions and blocks that are related to the DMLP are broadcast to secure channels that are only accessed by the honest and trusted authority nodes. However, depending on the IoT application under consideration, there might be a need for the integration of a suitable encryption algorithms such as AES-128 [176] into the architecture to encrypt the DMLP-related data within transactions (i.e., transaction's payload).

5.2.3 Blockchain platform

Designing a blockchain platform that is reliable, secure, and has acceptable latency is an essential part of this design. All transactions are executed on-chain, ensuring full traceability, and the validation of all transactions before processing enhances the system's security. A public blockchain platform, including its consensus mechanism, transactions, and block formats, was designed. In this chapter HDPoA consensus

mechanism that was introduced and discussed in details in chapter 4 will be utilised to secure the blockchain platform.

Block and transaction formats

A key aspect of this blockchain platform is the format of blocks and transactions. First, block headers were designed to allow for the inclusion of both the worker node's public key and the authority's signature. It was also adapted to distinguish between blocks that carry DMLP related transactions and those that do not. This will help the neurons in each layer deal with the blocks accordingly. The transaction format is an essential part and was designed to allow for different transaction types; the different fields of both block headers and transactions will be discussed in detail in subsection 5.4.1.

5.2.4 Data flow in the system

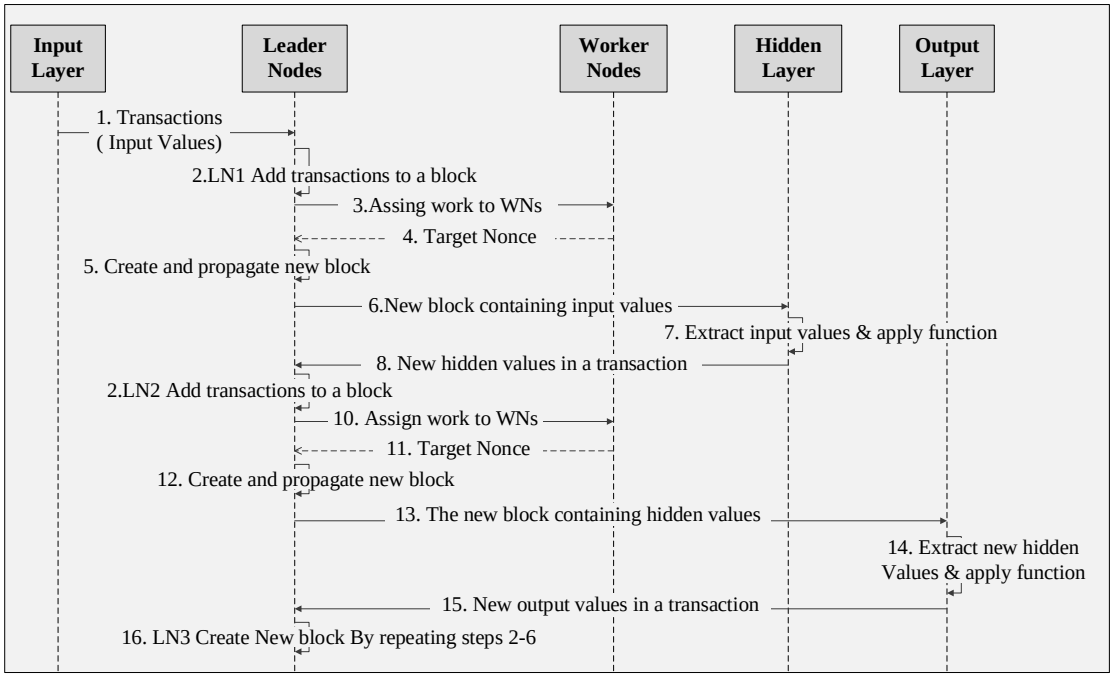


Figure 5.3: Flow of data between the system different components.

The data flow in the system is illustrated by Fig.5.3. The data is propagated through the network from one layer to another as follows:

1. First, the sensor nodes in the sensing layer collect data and pass it to the first neuron in the input layers. Neurons then process this data locally and create and submit their input transactions.

2. Transactions arrive at the miners' transaction pool. The first authority node in the round-robin (e.g., AN1) then collects all transactions, validates them, and adds them into a new block.

3. Next, AN1 creates mining tasks and sends them to the assigned WNs.

4. Once a worker node accepts the work, it will conduct the mining until either the node finds a solution, it receives an abort message from AN1, or it completes the iteration through all of the nonces assigned to it. If it finds a solution, it will submit its finding to all of the authority nodes.

5. When AN1 receives the WN solution (i.e., the nonce) to the task, it will validate the work, and if valid, it will send an abort message to all WNs. It will then sign and propagate the new block to the network.

6. When neurons in the Hidden Layer receive the new block, they will open the block, extract the relevant input values, perform the required calculation, and apply the activation functions. Their hidden values are then included in a new transaction and propagated through the broadcast channel to the miner nodes.

7. The next authority node in the round-robin (e.g., AN2) validate transactions and add them into a new block. It will then create mining tasks and send them to the assigned WNs.

8. Once a worker node accepts the work, it will start the mining process until it either finds a solution, it receives an abort message from AN2, or it completes the iteration. If it finds the target nonce, it will submit its finding to all of the authority nodes.

9. Once AN2 receives the WN solution, it will validate the work. If valid, it will send an abort message to all WNs. It will then sign and propagate the new block to the network.

10. After neurons in the output layer receive the new block, they will open the block, extract the relevant hidden values, perform the required calculation, and apply the activation functions. They then include their output value, which is the prediction of the DMLP expert engine, in a new transaction and propagate it through to the ANs.

11. The next authority node in the round-robin (e.g., AN3) validate transactions and add them into a new block. It will then create mining tasks and send them to the assigned WNs.

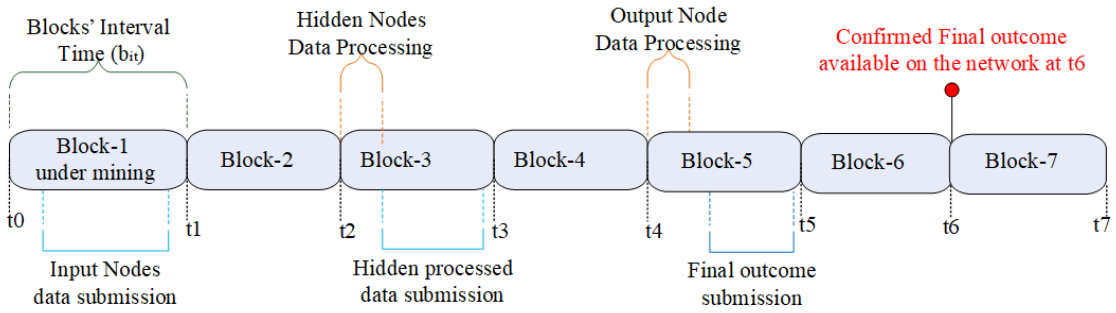
12. Once a worker node accepts the work, it will start the mining process until it either finds the correct nonce, it receives an abort message from AN3, or it complete the iteration. If it finds the target nonce, it will submit its finding to all of the authority nodes.

13. Once AN3 receives the WN nonce, it will validate the work. If valid, it will send an abort message to all WNs. It will then create and sign the block, and propagate it to the network.

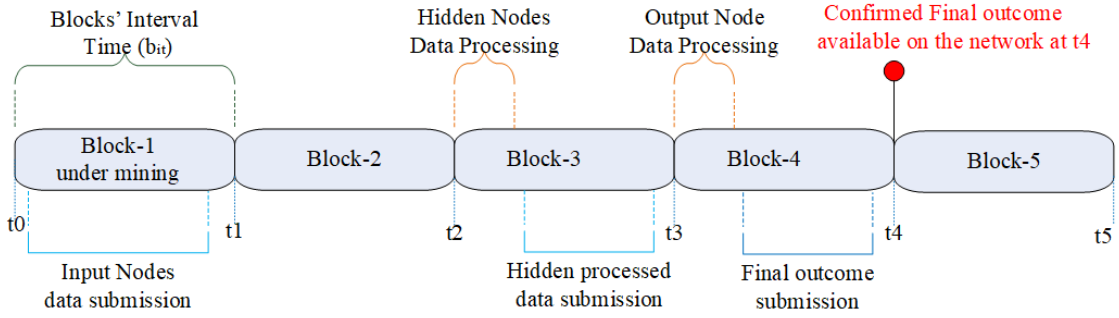
14. Finally, once the new block arrives and is validated in the network, the final outcome of the DMLP expert engine is now available to all interested nodes.

5.3 System analysis

The system under consideration is based on a blockchain technology where multiple nodes are connected to one another in a peer-to-peer network via wireless links. The data traffic generated on the network is from two main processes; both are broadcast transmissions: propagation of transactions and propagation of blocks to all nodes. Different types of transactions can be generated on the network; where DMLP related transactions always have the highest priorities when adding transactions into a new block. The parameters defined in Table. 4.2 will be used for this system analysis, however any new parameters that are specific to this system will be defined within the text as they are discussed. For systems analysis and implementation two different cases regarding the processing of DMLP related transactions were considered:



(a) Case I : Mining immediately without waiting



(b) Case II: Mining after waiting for time equal to ΔT

Figure 5.4: Block mining process.

- Case I: In this case the AN assigned to perform the next block mining process; once it receives a new block it will immediately start the mining process without waiting (see Fig. 5.4a).
- Case II: In this case the AN assigned to perform the next block mining process; once it receives a new block will not immediately start the mining process of the next block. A waiting time, ΔT, is introduced, which can be calculated by: $\Delta T = Tx_{pd} + Tx_p + Tx_v$. Where Tx_{pd} is the transactions propagation time, Tx_p is the time needed by the neurons in the network to process DMLP related transactions, and Tx_v is the time needed to validate transactions by the ANs. This is to allow neurons to perform necessary processing and submit their finding in new transactions, with the aim of reducing the overall system latency (see Fig.5.4b).

5.3.1 Overall confirmation time (OCT)

The probability of the final DMLP outcome arrival in the network is based on the Poisson process with arrival rate, λ.

$$P(T \leq t) = 1 - e^{-\lambda t} \tag{5.1}$$

The time t depends on the number of blocks (n) for which we need to wait before the final outcome arrives in the network, the block propagation delay B_{pd}, the block validation time t_v, and most importantly the total number of DMLP layers N_L. Two different cases were considered in the design as stated above, for both cases OCT has been analysed as follows:

Case I

As shown in Fig.5.4a, in this case AN start the mining process immediately, and using [4.4 and 4.6], the probability of the DMLP final outcome confirmation can be calculated by:

$$P(n) = \begin{cases} 1 - e^{-\left[\left(\frac{1}{\frac{D\times 2^{24}}{hp}}\right)\times\left(\frac{n}{N_L+1}\times\left(\frac{D\times 2^{24}}{hp}+t_v+B_{pd}+Tx_{pd}\right)\right)\right]} & \text{if } n \geq (2N_L) \\ 0 \quad \text{if } n < (2N_L) \end{cases} \tag{5.2}$$

This represents the probability of confirming the outcome of the DMLP including the confirmation of the input values and all hidden values. The total time for this confirmation process (OCT) can be calculated by:

$$OCT = (2 \times N_L) \times \left\lceil \frac{\ln(1 - P(n))}{\frac{-1}{\frac{D\times 2^{24}}{hp}}} \right\rceil \tag{5.3}$$

Case II

As shown in Fig.5.4b, in this case the AN was forced to wait for DMLP related transactions to be processed, by introducing the waiting time, ΔT. This means the probability of the AI final outcome confirmation can be calculated by:

$$P(n) = \begin{cases} 1 - e^{-\left[\left(\frac{1}{\frac{D\times 2^{24}}{hp(i)\times I}}\right)\times\left(\frac{n}{N_L}\times\left(\frac{D\times 2^{24}}{hp}+t_v+B_{pd}+Tx_{pd}+\Delta T\right)\right)\right]} & \text{if } n \geq (N_L+1) \\ 0 \quad \text{if } n < (N_L+1) \end{cases} \tag{5.4}$$

This represents the probability of confirming the outcome of the DMLP including the input and hidden values. The total time for this confirmation process (OCT) can be calculated by:

$$OCT = \left[(N_L+1)\times\left(\frac{\ln(1-P(n))}{\frac{-1}{\frac{D\times 2^{24}}{hp}}}\right)\right] + [\Delta T\times(N_L-1)] \tag{5.5}$$

5.3.2 Energy cost

For a system where P_s is the power consumption during the sleep mode, P_{tx} is the power consumption during data transmission, P_{rx} is the power consumption during data reception and preparation, and P_{AI} is the power consumption during the processing of DMLP transactions. Then the total power consumed by one node in the DMLP prediction process is:

$$P_n = P_s + P_{tx} + P_{rx} + P_{AI} \tag{5.6}$$

These powers depends on the time for each event assuming that: t_{rx} is the total time for receiving the block and extracting AI data. t_{tx} is the time required to prepare and transmit the transaction t_{AI} the processing time of the DMLP data taking a node to produce its own final calculation. Then the total energy of a node E_n can be calculated as:

$$E_n = (P_s\times t_s) + (P_{rx}\times t_{rx}) + (P_{tx}\times t_{tx}) + (P_{AI}\times t_{AI}) \tag{5.7}$$

Therefore the total energy required for the DMLP process is:

$$E_{AI} = \sum_{n=1}^{n=N} E_n \tag{5.8}$$

The amount of energy consumed depends on the node's state, the node will be in one of five different states. *Idle State (i)* where the node is on and not connected to the wireless channel, and energy consumed in this state is defined by E_i, this is the reference state. *Connection State (cx)* where the node is on and connected to the

available wireless channel (i.e. Wi-Fi connectivity), and energy consumed in this state is defined by E_{cx}. *Blockchain state (bc)*, where the node is connected to the blockchain network but is not performing any actions apart from submitting transactions and receiving and adding blocks to its internal storage, and energy consumed in this state is defined by E_{bc}. *AI State (AI)*, this is where the node, in addition to the actions performed in the blockchain state, is acting as one neuron of any of the DMLP layers, and energy consumed in this state is defined by E_{AI}. *Mining State (m)* where the node, in addition to the actions performed in the blockchain state, acts as a worker node and carries out some of the mining calculation, and energy consumed in this state is defined by E_m.

Based on these states the difference in energy consumption between two states δE can be calculated by:

$$\delta E = E_{state1} - E_{state2} \tag{5.9}$$

The i state represents the reference state, this will allow for the calculation of the energy consumed by a node when in any state in comparison to the reference state, for example, the energy consumed by a node when in the mining state is:

$$\delta E_m = E_m - E_i \tag{5.10}$$

5.4 System implementation and deployment

For practical trial purposes, a proof of concept system that consists of in house purposely built public blockchain platform and DMLP engine was developed and used. The following subsections describe the implemented system.

5.4.1 Blockchain implementation

A customised blockchain platform that implements the proposed consensus mechanism HDPoA discussed above were created. The deployed network consists of 23 Raspberry Pi devices used for both blockchain and DMLP implementations.

Block and transactions format

Table 5.2 shows the different fields of both transaction and block headers. Different fields have been introduced within the block header and the transaction that allow different neurons within the DMLP engine to distinguish between different values and easily extract relevant values in order to process them.

Different types of transactions have been identified. These include DMLP-related transactions such as input-layer transactions, hidden-layer transactions, output-layer transactions, notifications transactions, and parameter update transactions (administrative transactions). Non-DMLP related transactions include data transfer between nodes, payment transactions, and reward transactions.

The transaction type field is designed to allow the assigning of relevant types to each transaction. If the transaction is from an input neuron, then the type will have a value of one. Similarly, if the transaction is from a neuron in the first hidden layer then the value will be 2.0 etc. In this implementation, only one hidden layer was utilised, however, the transaction was designed to allow for a distinction between neurons if the

Table 5.2: Format of the block's header and the transaction

Field	Description	Size
Block Header		
Block Height	Number of blocks in order	4 bytes
Previous Hash	The hash of the previous/ parent block	32 bytes
Merkle Root	Merkle root of all transactions	32 bytes
Block Time	Time of the creation of the block	4 bytes
Difficulty	The difficulty level.	4-byte
Nonce	The target nonce that produced the desired difficulty level	4-byte
DMLP_Flag	Indicate the presence of DMLP data. Optional, always 1 if used	2 byte
Hash	The hash of this block produced by the AN	32 bytes
Public Key	The public key of the WN	33 bytes
Signature	AN should sign the block for validation	64 bytes
Transaction		
Transaction-ID	Unique transaction identifier	128 bits
Transaction Type	0, default. 1, DMLP input values. 2.0, DMLP values from the first hidden layer. 2.1, DMLP values from the second hidden layer. 3, DMLP values form the output layer.etc.	4 bytes
Reading Value	Sensors /Input reading, hidden, output values	1-4 bytes
Recipient	The address of the receiver of the transactions, if it is intended for DMLP then layer's name is used	2-33 bytes
Node Type	Distinguish between different neurons in different layers	1-4 bytes
Node Name	To help distinguish the flow of DMLP values from one layer to another	1-4 bytes
Timestamp	Time of the creation of the transaction	4 bytes
Signature	Sender should sign the transactions for validation	64 bytes

system has more than one hidden layer.

The block header was also designed with the DMLP in mind. The DMLP_ Flag field will be used by the AN in case the block carries any DMLP-related transactions, alerting other ANs and neurons once they receive the block.

5.4.2 DMLP implementation

A multilayer perceptron AI system was developed as a proof of concept and to test the validity of the architecture. It consists of three layers: input, hidden, and output. The process of developing the DMLP system spans three phases. The first phase encompasses the implementation and testing on a standalone PC running i5-8250U CPU @ 1.60GHz. This is then followed by the implementation and testing on a Raspberry Pi 3 Model B+ (1.4GHz 64-bit quad-core processor) as a standalone system. The trained DMLP was finally implemented as a distributed system on the blockchain network, where Raspberry Pi devices are used to act as individual neurons.

5.4.3 Dataset tested

For practical trial and testing purposes, three different datasets were used; the Iris flower dataset [177], air quality dataset from [178] and the Bot-IoT dataset from [179]. First the DMLP system was utilised to forecast the type of flower from the iris dataset; the system consists of four neuron in the input layer, five neurons in the hidden layer and three neurons in the output layer; each neuron was deployed on a Raspberry Pi. The data sets contained 150 Instances, with 4 features and one output. Then the system was configured to test future occurrences of air pollution that could affect air quality. It consists of 12 neurons in the input layer, 15 neurons in the hidden layer and three neurons in the output layer. This dataset contains a total of 9,358 Instances with 12 features, it has been prepared and normalised for the purpose of this implementation. A Raspberry Pi was used to emulate all input nodes, 15 Raspberry Pis used for the hidden neurons and one Pi for the output neurons.

Finally the system was configured to classify the type of attacks either DoS, distributed denial of service (DDoS), or reconnaissance based on the Bot-IoT dataset. The configured system consists of 10 neurons in the input layer, 12 neurons in the hidden layer and three neurons in the output layer. The used dataset contained 10,000 instance with 10 features and three outputs, and this data has been prepared and normalised for the purpose of this implementation, the complete description of this dataset can be found in [179]. Ten Raspberry Pis used as neurons for the input layer, 12 used in the hidden layer and one used for the neurons in the output layer.

5.5 Results

The system was tested while the nodes were deployed over two cities in the UK, Sheffield and Edinburgh, separated by a distance of about 310km. The following sections include the results and their evaluation.

5.5.1 Overall confirmation time

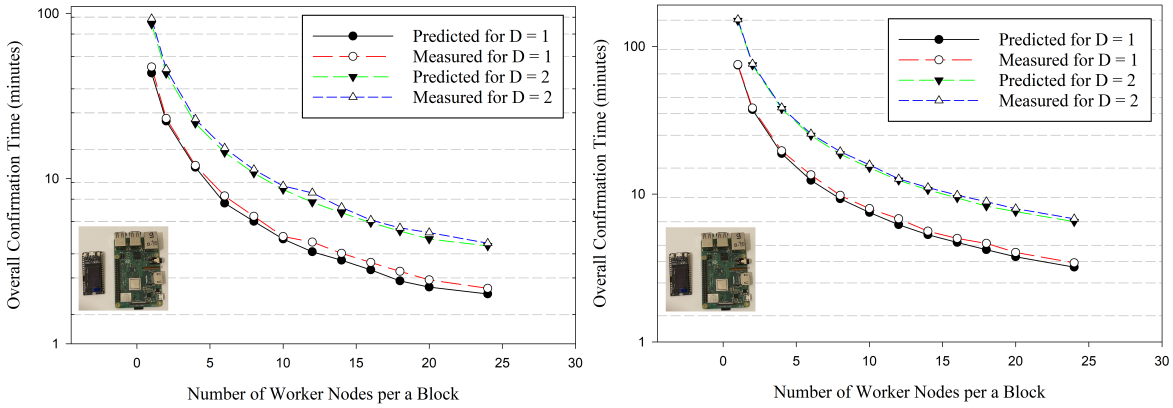
In both cases (see Fig.5.4), the system was tested using a different number of workers, three authority nodes, and different difficulty levels. As shown by Fig.5.5a and Fig.5.5b, the system was tested for two difficulties (D=1 and D=2). Both the predicted (using 5.3 and 5.5) and the measured OCT indicated that the overall system end-to-end confirmation time can be reduced as the network grows in size without compromising its security by reducing the mining difficulties.

In a network of 24 WNs with a difficulty of one, the OCT for Case I was 3.4 minutes and 2.16 minutes for Case II. For the same network with a difficulty of two, the measurements were 6.8 and 4.1 minutes, respectively. The t_m of the block has the most effect on the OCT, nevertheless the block and transaction propagation delays are important parameters. The results shown in Table 5.3 indicate that since the test was conducted with reliable Wi-Fi connectivity, the average of B_{pd} and Tx_{pd} have little effect on the OCT. However, as shown by the previous study in chapter 3, the B_{pd} is expected to have more influence on the OCT if the size of the block is increased or different communication methods, such as 3G and LoRaWAN, are used.

The network is small with regards to the number of workers; nevertheless, within IoT, with the presence of thousands of devices, the difficulty can be increased according to the number of available nodes in the network while maintaining an acceptable OCT according to the application.

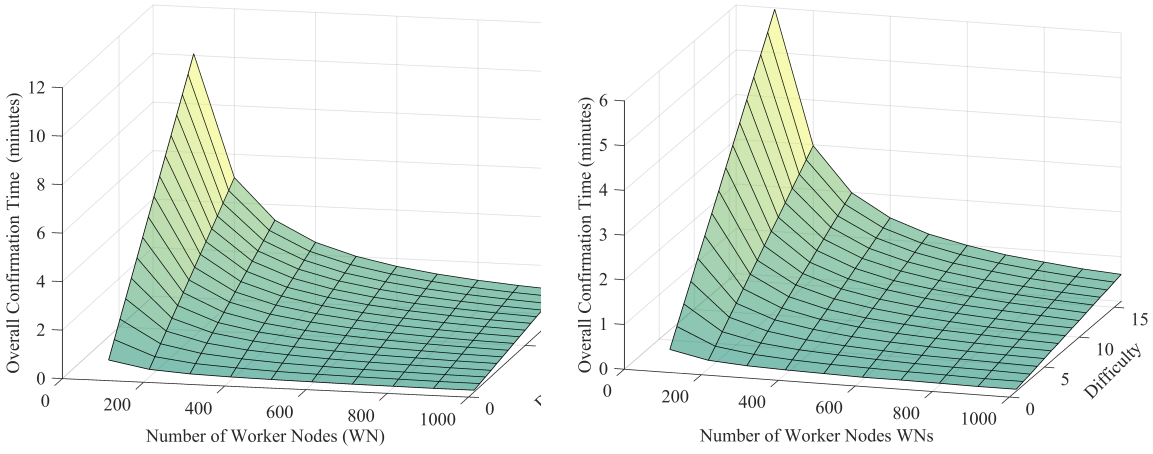
To better understand the effect on the OCT in the presence of hundreds of devices, OCT was calculated for different difficulty levels (1-16) using 5.3 and 5.5 and a different number of WNs, ranging from 100 to 1,000. A network that has more than 500 WNs can achieve OCT in less than two minutes for Case I and less than a minute for Case II. It is clear from both Fig.5.6a and Fig.5.6b that as the number of WNs increase, the OCT would be significantly lowered.

Another test was conducted to measure OCT using a network consisting of 20 workers for four different difficulties ($D = 1$, $D = 2$, $D = 4$, $D = 8$, and $D = 16$). As shown by Fig.5.7, when the difficulty increases, the OCT also increases; however, if more devices were available to use, the time could be lowered while maintaining an acceptable level of D.



(a) (b)

Figure 5.5: Overall confirmation time of the DMLP outcome.(a) Case I: Mining immediately. (b) Case II: Mining after waiting for time equal to ΔT



(a) (b)

Figure 5.6: Predicted OCT for different difficulties and WNs. (a) Case I: Mining immediately. (b) Case II: Mining after waiting for time equal to ΔT

Table 5.3: Latency measurements of important parameters

Parameter	Average	Maximum	Minimum
B_{pd}	295 ms	682 ms	227 ms
Tx_{pd}	66 ms	99 ms	41 ms
Tx_p	200 ms	220 ms	190 ms

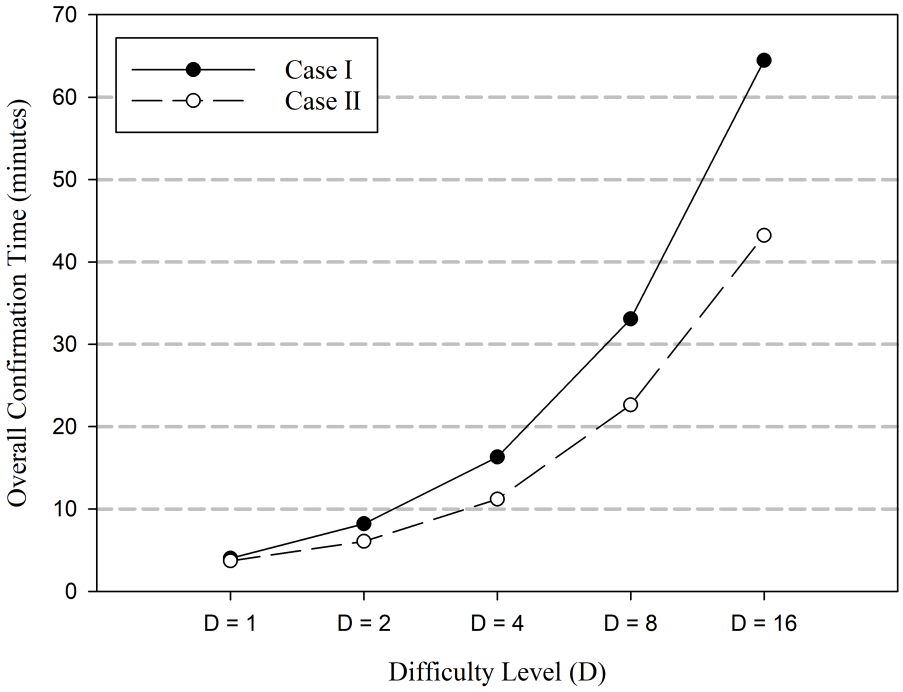


Figure 5.7: Overall confirmation time Using 20-WNs for different difficulties.

5.5.2 DMLP accuracy

The system for each dataset was first trained on the PC and tested on the Raspberry Pi. Each dataset was divided into 70% for training, 10% for testing on the PC, 10% for testing on the Raspberry Pi, and 10% for testing on the blockchain network, with the aim of comparing the resulting accuracies to ensure the trained DMLP results are consistent with the results from both implementations on the standalone PC and Raspberry Pi. The results showed that the accuracy resulting from testing over the blockchain is in line with both tests on the individual Raspberry Pi and the PC. All tests produced the same accuracy, 98% for iris dataset, 96% for the air quality dataset and 92% for the Bot-IoT dataset. This means it is practically possible to implement distributed AI engines on IoT devices based on blockchain technology with the accuracy unaffected.

5.5.3 Energy consumption

Energy is a vital aspect of this system; the system was designed with the intention of allowing low-power devices to be part of the blockchain and benefit from the DMLP services offered. This is done in exchange for a small amount of power, where these devices participate in the mining process and ensure the blockchain security.

The energy consumption was measured for each of the different system states when the system is implemented on different devices: a Raspberry Pi 3 Model B+ and an ESP32. Figure.5.8a shows the average energy consumption of both devices for different states. Most of the energy consumed is due to running the device's operating system (idle state). Using 5.9, it can be seen that the Raspberry Pi consumes 0.12 joules per second (J/s) when the system only performs DMLP-related tasks. However, when the system is in the mining state, it consumes 0.53 J/s, compared to only 0.15 J/s when mining using an ESP32 (see Fig.5.8b).

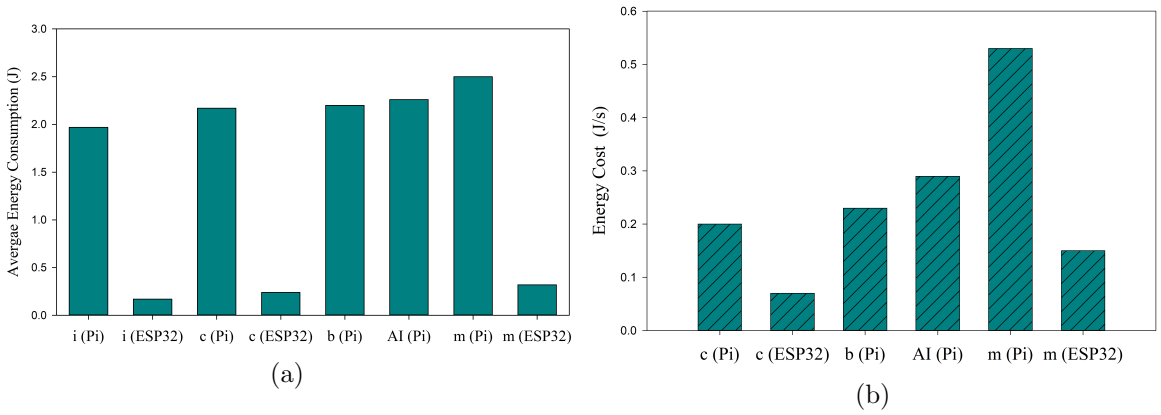


Figure 5.8: Energy measurements for R-pi and ESP32 when system in idle (i), connection (cx), blockchain (bc), AI, and mining (m) states . (a) Average energy consumption. (b) Energy cost δE during different states

5.5.4 Discussion

The DMLP performance on the blockchain network was not affected, and the results showed that it achieved the same accuracy as both of the standalone implementations. Overall confirmation time measurements and predictions showed that a network with thousands of available workers can achieve a reasonable difficulty to secure the network and manage the mining work between the workers to save energy and significantly reduce the OCT. The network currently consists of 23 Raspberry Pis and six ESP32, nevertheless all of the measured results, especially those related to OCT, were almost identical to the predictions that were based on the system analysis in section 5.3. This indicates that the proposed architecture is valid and the blockchain platform can support the distributed implementation of AI using IoT hardware capabilities.

The energy cost during different system states including during executing DAI related tasks and the mining process have been measured, however it was difficult to compare these measurements with any of the related works because all of them have not provided such measurements. As described in subsection 4.6.7, within HDPoA devices might spends days without performing any work.

In terms of security and user trust, this architecture provides a distributed public blockchain platform that ensures the security of data through the proposed HDPoA that is suitable for implementation within the IoT realm. Users can see that this is a trustworthy platform because it relies on an immutable, transparent, and secure blockchain. The system relies on HDPoA to ensure its security and data integrity. For the security analysis and risk determination of HDPoA please see section 4.4.

Table 5.4: Performance comparison with related works

Work	Communication overhead	Risk of Single Point of Failure?	Data Security (Integrity and Availability)	Accuracy
[143]	- Utilised blockchain for data handling from the edge layer only (+). - No local processing. - PS = 1	- Yes, Relies on cloud and data centre - PS = 1	- Blockchain is utilised to ensure the protection of Data starting from the edge (+). - PS = 1	- Not Measured - PS = 0
[170]	- Local learning model processing (+) . - Blockchain used to distribute parameters (+). - PS = 2	- Yes, need cloud offloading - PS = 1	- Blockchain is utilised to ensure the protection of data starting from IoT devices (++) - PS = 2	- 75% - PS = 1
[164]	- Uses two levels of Fog processing in a hierarchical structure (++) - PS = 2	- Yes, relies on central cloud for some of the AI Implementation - PS = 2	- No mechanisms in place - PS = 0	- Not Measured - PS = 0
[165, 166, 167]	- Nodes need to connect to each other from layer to another to exchange data. - PS = 0	- Yes, clusters' heads could be a problem. - PS = 2	- No mechanisms in place - PS = 0	- Not Measured - PS = 0
[168]	- Some local processing but there is a need to exchange data with aggregator and prone to bottleneck (+). - PS = 1	- Yes, needs aggregator and cloud - PS = 0	- No mechanisms in place - PS = 0	- Up to 97% -PS = 2
[169]	- Many local processing but devices need to exchange data with a controller. (++) -PS = 2	- Yes. devices relays on a controller - PS = 0	- No mechanisms in place - PS = 0	- Vary between 90%-96.7 - PS = 2
This work	- Some local processing (+). - Blockchain utilised to handle data starting from sensing layer (+). - Devices only need to propagate transactions once to miners nodes (+). - PS = 3	- Fully decentralised (+). - Any IoT devices can be utilised as a neuron (+). - It also allows for the processing of the same data by multiple DAI models (+). - PS = 3	- Blockchain is utilised to ensure the protection of data integrity and ensures its availability starting from the sensing layer (++). - PS = 2	- Vary between 92%-99% -PS = 2
PS = Performance Score. For each category the score is assigned between 0 to 3 where 0 is the worst in class and 3 best in class.				

In terms of robustness and redundancy, this proposed architecture does not rely on a third-party central entity to process and share data, which eliminates a single point of failure by leveraging distributed architecture.

Table 5.4 provide details performance comparison between this work and other related works in four performance metrics; communication overhead, risk of single point of failure, data integrity and availability, and DAI engine measured accuracy. Figure.5.9 shows the scored for each architecture in both data integrity and availability and communication overhead metrics along with the risk associated with each one of them. This proposed architecture is the best for communication overhead and has the lowest risk of single point of failure. It also ranked among the best architectures in both data integrity and availability and measured accuracy metrics. Finally, this architecture provides a trustworthy, self-managed, and self-regulated, public platform that can be utilised to integrate DAI into IoT hardware.

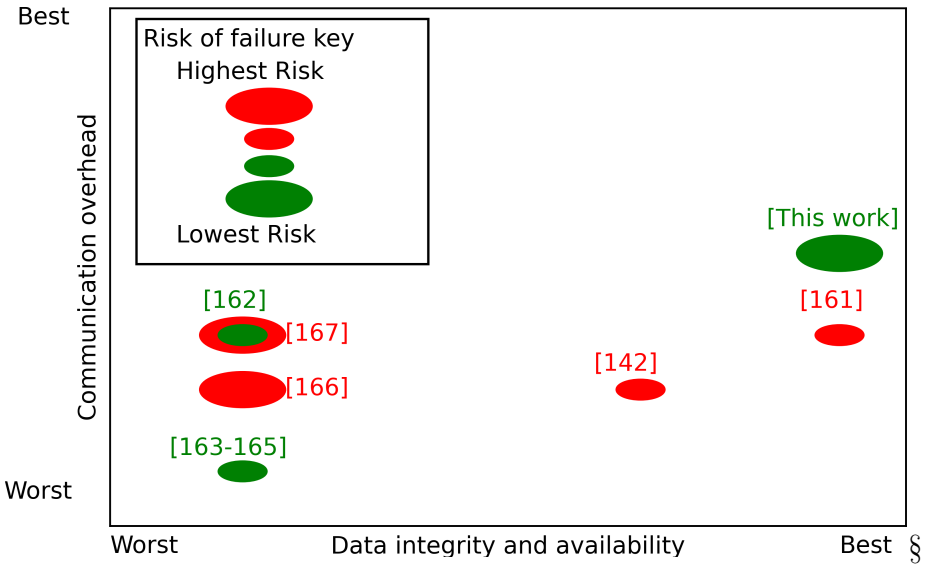


Figure 5.9: Illustration of Comparative Performance Showing Relative Best in Class/Worst in Class for Different Metrics as Listed in Table 5.4. Numbers in [ ] are the Paper's References.

5.6 Summary

A distributed, decentralised, and secure blockchain-based architecture for supporting DAI on low-power and low-cost IoT devices was proposed. A practical implementation of DMLP using a distributed IoT hardware platform was accomplished using a real-world example application. The results showed that, in terms of predication accuracy, it is possible to implement a DAI system over an IoT platform based on blockchain technology. It also showed that this architecture provides a secure, scalable, and dis-

tributed approach that utilises IoT devices as a platform for AI implementation with a minimal impact on their computational resources.

Chapter 6

Blockchain for Supporting AI Implementation and Securing IoT Applications at the Edge

In this chapter, a new blockchain protocol and a novel architecture that integrate the advantages offered by edge computing, artificial intelligence (AI), IoT end-devices, and blockchain were designed, developed, and validated. This new architecture has the ability to monitor the environment, collect data, analyse it, process it using an AI-expert engine, provide predictions and actionable outcomes, and finally share it on a public blockchain platform. For the use-case implementation, the pandemic caused by the wide and rapid spread of the novel coronavirus COVID-19 was used to test and evaluate the proposed system.

6.1 Introduction

Over the years IoT systems have been grown rapidly and increasingly used by many different organisations and users within different sectors, such as health-care and industry. One reason for this rapid increase is the fact that IoT is a major source of big data, which is generated from the huge number of smart devices connected to the internet. This data provides users with the ability to generate valuable information and knowledge. However, the handling of this data by organisations is often done by making IoT systems rely on central data processing entity such as the cloud for securing and managing IoT devices and for processing the data collected by these devices.

In chapter 5 an architecture based on blockchain for the implementation of a DAI engine was proposed, developed, and tested. In this chapter the proposed architecture will focus on the utilisation of edge nodes for the implementation of an AI engine and the use of the blockchain to support this implementation and ensure the integrity of the data, thus allowing the AI engine to access a trusted data. This means all analysis and validation is based on the idea that one node is running the full AI engine.

6.1.1 Problem statement and background

The approach of utilising a central entity such as the cloud has its own drawbacks, for instance it introduces the risk of single point of failure, communication overhead, and bottleneck. This can easily affect the system overall performance and security and makes users' experience unpleasant. It is essential for many IoT mission critical applications to have secure and reliable solutions that can provide low latency for data processing. In this regard, edge computing has grown fast to facilitate this kind of solution that provides faster data processing that allows for near real time actionable outcomes. The edge computing allows for location-awareness services that allow IoT applications to produce faster and reliable services for users. While edge computing technologies provide IoT systems with these great advantages, yet due to the heterogeneous nature of edge and IoT end devices the collected data may not be fully secured during transit and while stored [150].

Combining the edge technology with the blockchain technology can provide a solution that is decentralised, robust, and secure overing the IoT devices the ability to interact and share data among themselves and with users. The available resources on the edge devices would help in providing the required computation and storage resources for the blockchain technology.

With the integration of a distributed, self-managed, and decentralised network, both the dynamic and distributed IoT system and the intelligence AI engine will benefits greatly from such integration [11, 14]. With the presence of the edge computing these benefits includes; a) providing IoT networks with a reliable ability to control the distribution of computation requirements over large number of distributed devices, b) it will improve the security posture of the overall IoT system by enhancing its ability to ensure data integrity and availability and holds all participants nodes accountable of their actions [21], c) it will also enhance the intelligence AI engine ability to perform the required analyses and provide desires outcome using these trusted data.

Different related works have tried the integration of blockchain into the edge along with AI into one secure and intelligent system. For example The BlockDeepNet framework was proposed by [140] for data analysis within IoT systems. It combines blockchain technology, smart contracts, and deep learning. By utilising both Ethereum and smart contracts the authors of [133] proposed a mobile edge system for service sharing and data processing in smart-city IoT applications. The deepblockiotnet [180] introduced a blockchain-based secure deep learning approach for the IoT systems.

The DeepConin framework was introduced in [142] for fraudulent transaction detection and blockchain-attack prevention based on deep learning and blockchain within smart-grid applications. A similar framework based on blockchain was introduced in [143]. It uses deep learning and SDN to allow smart city applications to access and utilise cost-effective and high-performance computing resources. Many more authors as previously discussed in section 2.6 from chapter 2 have also tried the same concept of integration. Table 6.1 provides a summary of the important related work, including the solution provided, the applications, and the limitations in each work.

Table 6.1: Summary of the important related works

Work	Application	Solution	Technology-utilised	Limitations
[134]	Data-sensitive applications for example, healthcare.	Proposes and implements an Ethereum blockchain based architecture with edge artificial intelligence to analyse data at the edge of the network and keep track of the parties that access the results of the analysis	Blockchain, edge computing, and AI.	Scalability can be an issue because gateway can only support a limited number of end-devices. Requires additional scalability analysis.
[135]	Mission-critical applications	Blockchain based edge intelligence (EI) system for improved data security, privacy, and performance. It uses a public blockchain to ensure the communication security of consumer electronic devices (CEDs) and a private blockchain to ensure communication security among EI servers.	Blockchain, smart contract, edge computing, and AI.	Lacks analyses in terms of the effect of the overall system latency and scalability on mission critical applications and comparison with other related works.
[142]	Energy exchange for smart grids	An intrusion detection system (IDS)that employs recurrent neural networks (RNNs) to detect network attacks and fraudulent transactions in the blockchain-based energy network.	Blockchain, RNN, and smart grids.	The consensus process is long and this could have a negative impact on the transaction finality and the system latency
[143]	Smart city	Deep learning-based IoT-oriented infrastructure for a secure smart city where blockchain provides a distributed environment at the communication phase of CPS, and software-defined networking (SDN) establishes the protocols for data forwarding in the network.	Blockchain, DL, SDN, fog, cloud, and IoT.	The proposed system does not realise the full potential of the decentralisation approach provided by the blockchain as it relies on a central cloud entity
[146]	AI-envisioned Internet of Vehicles (IoV)-based smart city	Blockchain-based batch authentication scheme for IoV.	AI, blockchain, fog, cloud and Internet of Vehicles (IoV)	Network relies on cloud to convert and mine full blocks.
[147]	Healthcare applications (diseases' control)	Blockchain-based AI-empowered pandemic situation supervision scheme in which a swarm of drones embedded with AI is engaged to autonomously monitor pandemic outbreaks	Blockchain, AI, drones, edge computing, and cloud.	While this is a good solution for controlling the spread of diseases or viruses; it introduces the issue of human and data privacy.
[180]	Deep learning DL for object detection in IoT applications	Secure DL model based on blockchain to support collaborative DL in IoT.	DL, blockchain, smart contract, edge, and IoT.	Local models only, trained on local private data. Limit models' ability to access all data.

6.1.2 Contribution of the chapter

The contribution in this chapter can be summarised as follows:

- The design and development of an architecture that integrates four different technologies: IoT, AI, edge computing, and blockchain in one system that can monitor and sense the environment, learn, analyse data based on the requirements of the executed task, and produce actionable outcome. The proposed system is based on the integration of low-cost edge devices and takes full advantage of their available storage and all IoT devices' computation power to provide a data-processing and sharing public blockchain platform.

- This architecture was validated experimentally using 14 low-cost, flexible IoT hardware entities. Practical implementation and performance analyses in terms of system latency, system accuracy, and energy consumption of real-world applications in the form of an early warning system for the detection of COVID-19 in sewage water were carried out. The implementation of the Covid-19 application as a use case was motivated by the work by the Yorkshire Water company and the Department of Civil Engineering in regard to Covid-19 virus detection in sewage water.

Published papers

The following papers based on this chapter were published:

1. S. M. Alrubei, E. Ball and J. M. Rigelsford, "A Secure Blockchain Platform for Supporting AI-Enabled IoT Applications at the Edge Layer," in IEEE Access, vol. 10, pp. 18583-18595, 2022, doi: 10.1109/ACCESS.2022.3151370.

2. S. Alrubei, E. Ball and J. Rigelsford, "A Secure Distributed Blockchain Platform for Use in AI-Enabled IoT Applications," 2020 IEEE Cloud Summit, 2020, pp. 85-90, doi: 10.1109/IEEECloudSummit48914.2020.00019.

6.1.3 Organisation of the chapter

The rest of this chapter is organised as follows: section 6.2 presents the proposed system architecture. In section 6.3, an analysis of the system followed by the security analysis in section 6.4, and then the implementation and testing of the system-example application in section 6.5. Section 6.6 presents the results and the discussion, and section 6.7 presents the summary of the chapter.

6.2 System architecture

The proposed architecture provides a system for data collection, processing, and analysing and produces a sharable outcome among nodes. A general overview of this architecture is presented in Fig.6.1.

The platform operates according to the following three steps:

- The first step is the ***monitoring and collection***. In this step, the IoT system monitors the environment or situation and utilises its sensors at the lowest layer to collect the environmental or change data.

- The second step is the ***analysis and prediction***. In this step, the collected data is then propagate to the intelligent engine located at the edge nodes for analyses and providing predictions.

- The third and the final step is the ***sharing***. In this step, the produced outcome from the edge devices are shared among all participant nodes on the freely accessed public blockchain network.

6.2.1 Architecture different layers

To accomplish the three previously discussed steps and provide free access to the public blockchain platform, an architecture that consists of four different layers was designed. Figure.6.2 provides the layout of these layers.

Sensing layer. This is the lowest layer in the architecture and is the most important layer; it is the data feeder to the sharing platform. In this layer, a wide range of many low-cost, low-power, and small sensor devices are used for monitoring and data collection. The collected sensor data will then be submitted to the gateway, which can be in the form of low-cost devices (e.g., Arduino ESP-32), which can then be validated and prepared and then submitted to the next layer for processing. This aids in achieving the first step of the architecture, which is the *monitoring and collection*.

Network layer. The data submitted by the gateway is then transferred to the next layer. This is where the network layer takes part. In this layer different communication links can be utilised (for example, wireless connectivity, such as the Wi-Fi, LoRaWAN, or 5G, or a wired connectivity).

Processing layer. This layer is equipped with the necessary AI engine to perform the required analyses and is responsible for achieving the second step in the architecture, which is the *analyses and prediction*. Devices deployed at this layer can be low in both cost and power, and one example of such a device is the Raspberry Pi (R-pi). The data collected by the sensing layer arrived at this layer. The AI-expert engine located at these edge devices will then be used to process and analyse the data and then provide predictions and necessary outcomes that can be used to help the decision-making process. All nodes located in this layer should be a full nodes of the blockchain platform (see section 4.2). This means that these nodes will be able to share their collected data and AI outcomes instantly with the rest of the blockchain clients. In

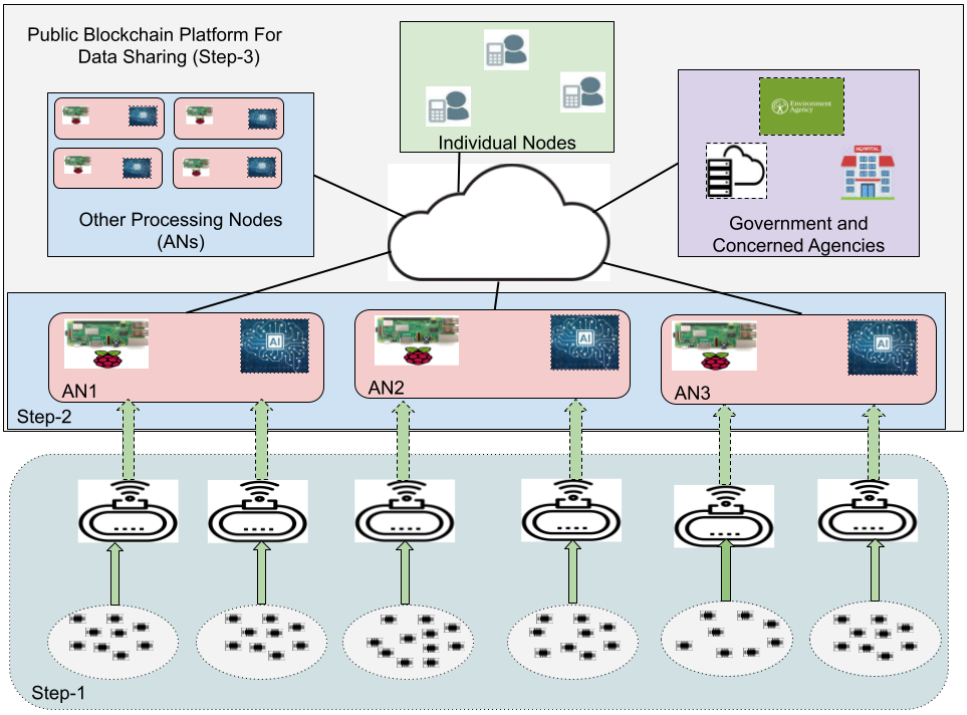


Figure 6.1: General Concept of the System Architecture.

doing so, the platform will have a continuous stream of data (collected by sensors and the outcome of the AI), allowing for a better performance of the system.

Sharing Platform. This is a freely accessible and a public blockchain platform and is responsible for achieving the *sharing* step which is the final and last step of the architecture. All the devices in the processing layer are part of the public blockchain. This would allow any organisations, users, or other concerned parties to be part of such platform and have the ability to freely access all processed AI and collected data.

6.2.2 Blockchain protocol

For the blockchain platform HDPoA consensus protocol will be used. In HDPoA The roles of the nodes are classified into two types of nodes; ANs (only from the FN) and WNs, which can be any node that is able to participate in the mining process (can be from both FN or HN); only the current AN that manages the mining process cannot be part of the WNs. All other nodes should make themselves available for the mining tasks. More details regarding how HDPoA works can be found in chapter 4.

6.3 System analysis

The proposed system utilises a free-access public blockchain network. In this network, the sources of data traffic are broadcast transmission processes of transactions and

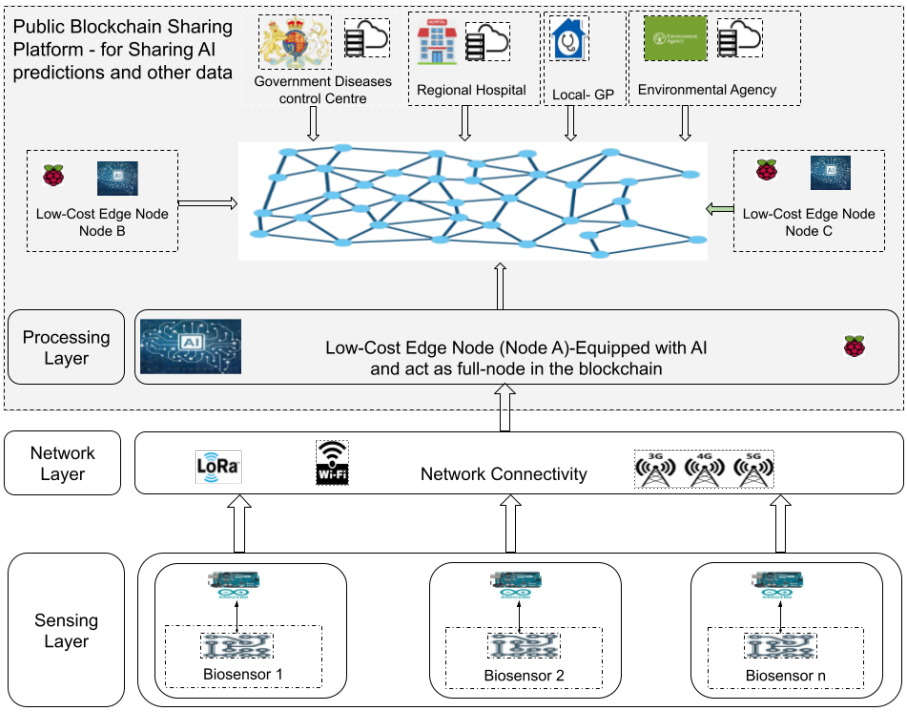


Figure 6.2: Example Application - Detailed system architecture.

blocks through the network. The important parameters used are listed in Table 4.2, and any new and system-specific parameters will be defined as they are discussed in this chapter.

6.3.1 System overall latency

Measuring the overall system latency (L); time from the submission of the AI input values until the final AI output value is confirmed on the blockchain network; is an important aspect of the system performance metrics. The probability of any transaction (including AI-related transactions) to arrive and confirm on the HDPoA-based blockchain network can be measured based on the Poisson process as described in subsection 4.3.2. Based on this, and using [4.6 and 4.7] the probability ($P(n)$) of the confirmation of all AI-related transactions-both input and output- can be calculated as:

$$P(n) = 1 - e^{-\left[\left(\frac{1}{\frac{D\times 2^{24}}{h_p}}\right)\times(n\times 2)\times\left(\frac{D\times 2^{24}}{h_p}+B_{pd}+t_v+Tx_{pd}\right)\right]} \tag{6.1}$$

This is based on the fact that there are two round of confirmation; one for the input values and one for the final outcome of the AI-engine.

To calculate (L), all AI-related transactions were assumed to arrive at the elected AN transaction pool on time to be included in the next block. As shown by Fig.6.3, transactions were assumed to be submitted before the time $t_1 - Tx_{pd}$ will be included in the next block (Block_n). Another important aspect of the proposed system that needs to considered is the fact that the AI input values and the final outcomes of the expert engine all will be validated and confirmed on the blockchain network. This means that two rounds of confirmation are needed before the arrival of the final outcome. Based on these assumptions and considerations, the overall system latency L can be calculated as follows:

$$L = 2 \times \left(\left[\frac{ln(1-P(n))}{\frac{-1}{D\times 2^{24}/h_p}}\right] + B_{pd} + Tx_{pd} + t_v\right) \tag{6.2}$$

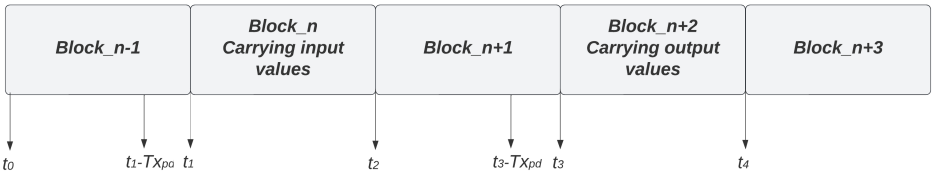


Figure 6.3: Overall system latency and the processing of AI data.

6.3.2 Power cost

The energy consumption of both the blockchain and the AI-engine is a significant parameter of the proposed architecture; therefore, it is important to analyse this parameter and identify its impact on the power sources of the devices. Each device is typically in one of the following states:

1. Sleeping state (s): In this state, the device will not perform any task. Instead, it will goes to sleep and wakes up by a timer or event. In this state, power consumption can be identified by P_s:

2. Connectivity state (cx): In this state, the device's operating system is active, and it is connected to the available connectivity link (i.e., Wi-Fi link) and does not perform any task. In this state, the power consumption can be identified by P_c. This is the reference state, in which the energy consumption of the other states is compared to.

3. Data exchange state (dx): During this state, the device will indeed be in data transmission or data reception. The power consumed during this state can be defined as P_{dx}.

4. Mining state (m): In this state, the device is engaged in the blockchain-mining activity by performing a small task of the block-mining process, in search of the nonce for the new block. In this state, the power consumption can be identified by P_m:

5. prediction state (pre): In this state, the device will receive the AI-input values and then will utilise its built-in AI-expert engine to process these values and produce an AI prediction. During this state, the power consumed is defined by P_{pre}:

Based on these states, the total power P_{tot} consumed by any device in the blockchain network can be calculated by:

$$P_{tot} = P_s + P_{cx} + P_{dx} + P_m + P_{pre} \tag{6.3}$$

The system will be in each state for a certain amount of time, and the time of the sleeping state can be identified as t_s; during connectivity state, it can be defined as t_{cx}; during the data exchange state, it can be defined as t_{dx}; during the mining state, it can be defined as t_m; and during the prediction state, it can be defined as t_{pre}. These values, along with the measured power in each state, can be used to calculate the energy consumption E_n of any device as follows:

$$E_n = (P_s \times t_s) + (P_{cx} \times t_{cx}) + (P_{dx} \times t_{dx}) + (P_m \times t_m) + (P_{pre} \times t_{pre}) \tag{6.4}$$

Based on the different system states, the cost of power (P_{cost}) (J/s) during the mining state (P_m), or the prediction state (P_{pre}), in comparison to when the system is

in the connectivity state (P_cx), the reference state, can be calculated using the following equation:

$$P_{cost} = (P_m, P_{pre}) - P_{cx} \tag{6.5}$$

6.4 Security analysis

The deployed blockchain platform is based on HDPoA consensus mechanism, this means the system security depends on how secure is HDPoA. In section 4.4 the security analysis of HDPoA was provided. In this section important attacks will be analysed based on their effects on the proposed system. For this, a qualitative risk assessment was carried out using the NIST SP-800-30 standard [181]. Table 3.5 shows the determination of the risk level based on attack likelihood and its impact level.

6.4.1 DoS attack

The architecture was designed to allow nodes to access the services provided by the AI-expert engine; however, a DoS attack against the node hosting the engine is possible. The system was designed to enhance the robustness of the AI engine by utilising the distributed approach provided by the blockchain. In this experiment, one node for hosting the AI engine was used, however, the system has the ability to allow any of the ANs to host the AI-expert engine, and each node can produce its own prediction value, as all nodes have access to the data in the blockchain. In fact, with the implementation of the blockchain and the bespoke protocol formats, it is possible to implement the AI in a distributed approach, in which each AN can host one layer or more of the engine, allowing for more transparency, as the flow of the data from one layer to another will be validated on the blockchain network. This makes the system robust against any attacks that target the services provided by the AI engine.

Therefore, although the likelihood of a DoS attack is *high*, its impact's level is *low* making the residual risk level of this attack *low*.

6.4.2 Data integrity attacks

The integrity of the data is very important for ensuring that AI prediction is performed on legitimate and fresh data. Nevertheless, the integrity of the data can be targeted and can be vulnerable to manipulation. In the proposed system, the blockchain platform is utilised to first validate new data before adding them to the system. Second, it ensures that the added data cannot be modified or deleted. This feature of the blockchain enhances the system's ability to ensure that the AI engine accesses only trusted and fresh data. However, there is still the risk that some sensors may feed the system a fabricated or untrue data. This might not be detected; however, once any node is

discovered behaving in a manner that could harm the data integrity, HDPoA consensus algorithm will block that node from feeding or accessing the data on the blockchain.

With the presence of HDPoA-based blockchain platform, the likelihood of any attacks that can harm the data integrity is *low*, and their impact can be *high*. This makes the residual risk level of any attack *low*.

6.4.3 Malicious AN

It is possible that one of the ANs can be malicious or that it can be compromised. A malicious AN can harm this architecture in two ways: either by forging a new block or by producing an untrue AI outcome. In both cases, the platform can manage this node. First, the block-mining process is performed by multiple unrelated WNs. Second, other ANs on the network only add and validate a new block produced by the elected miners (please see chapter 4 for more details). If such a block is not valid, then the node that produced the block will be eliminated from the AN category, and it will have to build its trust from zero. In terms of the AI prediction, the system was built to allow any trusted AN to host the AI-engine. This would allow the network to utilise more than one node to perform the AI prediction, allowing for more validation and outcome-consensus of any outcome before it is confirmed on the blockchain.

The impact of any attack from any malicious AN *high*; however, the likelihood that ANs can misbehave or become compromised is *moderate*, making the residual risk level of any attack *moderate*.

6.4.4 Attack on communication links.

Attacks on the communication links, such as jamming and DoS are possible. The main focus on this study was to evaluate the performance and security of the blockchain platform when utilised in supporting AI-enabled IoT applications. Hence, assuming that the network provider will has adequate security mechanisms and protection in place.

Even though attack likelihoods on communication links can be *moderate*; the impact of such attacks is *low* on the assumption that adequate protection is in place, making the residual risk level of this attack *low*.

6.5 Implementation and testing of example application: AI-enabled system for tracking viruses in sewage water

The worldwide pandemic caused by the novel coronavirus COVID-19 has wreaked havoc among organisations, governments, and businesses. The lack of robust and reliable tracking and early warning systems and platforms has resulted in the loss of

many lives and major economic losses. Technologies such as blockchain, IoT, and AI can provide governments with a secure, intelligent, and robust platform for tracking and tracing and for implementing early warning system. Such a system is a desirable solution that can help in tackling the spread of COVID-19 or other future viruses and allows governments to save lives and reduce economic impacts. In this system, the sensing and data-collection ability of the IoT can be combined with the decentralised and secure abilities offered by blockchain and, with the intelligence capabilities of the AI, can provide the best solution that can be utilised to tackle current and/or future pandemics.

According to the author of [182], wastewater-based epidemiology (WBE) offers an effective method for the early detection of possible viral infections before its actual spread by tracking and measuring the presence of viral genetic markers in wastewater. The proposed architecture discussed above is shown in Fig.6.2. It provides an affordable and more practical system that can be utilised to efficiently and securely predict the possibility of any viral infections. By continuously collecting and analysing the data, the system will serve as an early warning notification for concerned entities, such as disease-control agencies, allowing them to take effective and early actions to slow down or stop the spread of such a virus. Additionally, the system's ability to serve as an early warning notification platform can also help governments evaluate the effectiveness of other virus-control measures, such as social distancing, lockdown, and mass testing.

6.5.1 Blockchain implementation

A blockchain platform secured by the HDPoA consensus mechanism, which was previously developed, implemented, and tested, was deployed. For this platform to handle different types of transactions, including AI-related transactions, the bespoke transaction and block's header formats that were created and implemented and discussed in detailed in section 5.4.1 and table. 5.2 will be utilised.

In terms of the consensus mechanism ANs are responsible for ensuring the security of the blockchain by managing the mining process, validating transactions and blocks, validating any work performed by a WN, and validating each other's work.

6.5.2 Data flow

Figure.6.4 shows the different steps of the data flow in the system. These steps are as follows:

1. First, the sensors that are installed in the different sewage-water locations will sense and collect data in the form of readings of any presence of viral agents in sewage water. One sensor that can be used is a biosensor with a biological receptor [183]. These readings are submitted to a gateway that can be either a FN or a HN.

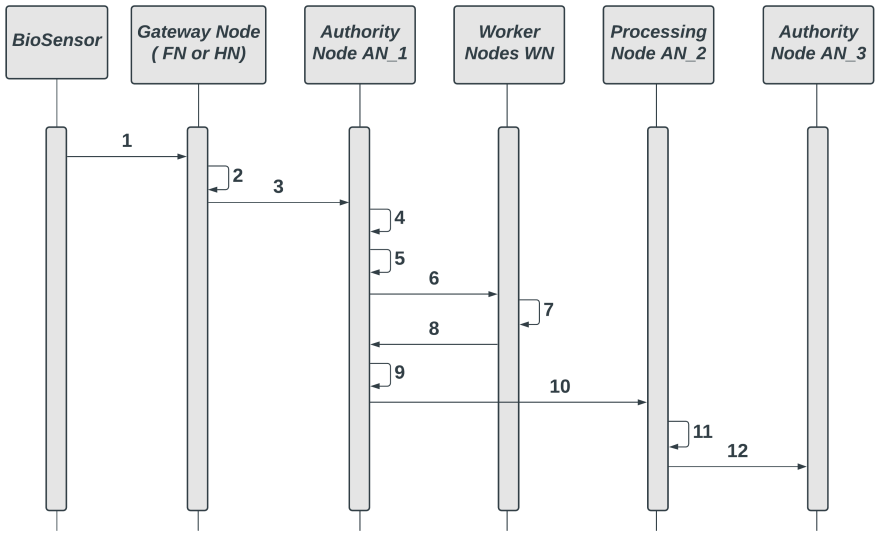


Figure 6.4: Data flow within the different layers of the architecture.

2. The gateway will validate the readings (if it was signed by the sensor), create a transaction, and label the type of this transaction as an AI-input value.

3. Then, the gateway will broadcast the transaction to all ANs on the blockchain network.

4. Assuming node AN_1 is responsible for the mining process of the next block (i.e., $block\_n$), it will collect transactions, validate them, and add them in a new block, and it will set the AI-flag to the appropriate value (0 or 1).

5. Then, in step 5 AN_1 will create mining tasks for all the available WNs and send the tasks to each one of them.

6. Upon receiving the task, any WN will accept it and begin performing the process of searching for the correct nonce that satisfies the current difficulty.

7. If any WN finds the nonce that satisfies the next block difficulty, it will forward it to AN_1 and all other ANs for future validation.

8. AN_1 receives the nonce and then will validate it by executing one hash.

9. If the nonce is valid, then $block\_n$ will be signed and propagated to the network.

10. Once $block\_n$ arrives at the processing node (assuming this node is AN_2), it will extract the relevant AI input values, feed them to the AI-engine, process them, and produce the final AI outcome (the prediction). This processing of the input values by the AI-engine occurs during the mining process of $block\_n + 1$.

11. AN_2 will then add this outcome to a transaction and propagate it to all ANs on the network.

12. Assuming the node responsible for managing the mining process of the next block ($block\_n + 2$) is AN_3, it will execute the same steps as 4–9, and will then propagate $block\_n + 2$ that carries the final AI outcome to the network. Now, the AI outcome is available on the public blockchain and can be accessed by any interested government entity or organisation.

6.5.3 Experiment and testing

To test the developed system, a blockchain network was deployed using 16 R-pis. Two were used as ANs and 14 were used as WNs. One AN was used for managing the mining process, and an AI-engine was developed, trained, and deployed on the other AN. The AI-engine consists of three inputs and three outputs. For the hidden layer, the tenser-flow keras dense function [184] was utilised, and for activation functions, Relu and SoftMax were used. Figure.6.5 shows the architecture of the AI-expert engine.

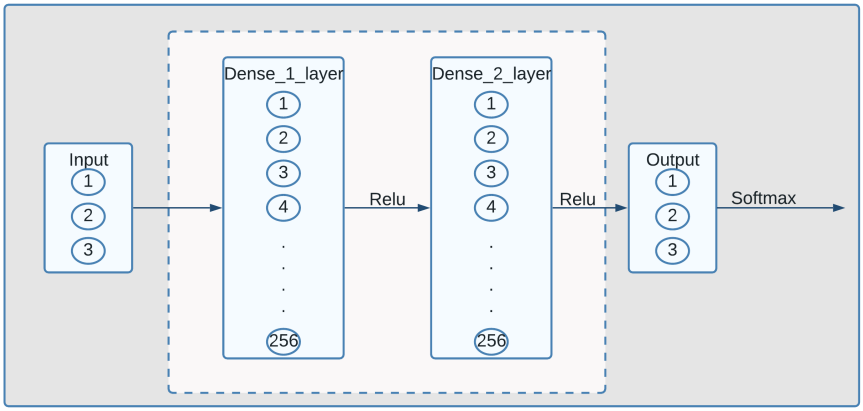


Figure 6.5: The architecture of the AI-expert engine.

It was very difficult to find any COVID-19 dataset related to wastewater, therefore, based on the literature, a new dataset was created. Based on [185], biosensors, such as electrochemical reaction biosensors, can be utilised to measure and detect the levels of viral nucleic acids, proteins, and small molecular antibodies.

Different studies investigated the use of biosensors for detecting COVID-19 in wastewater [185, 186, 187]. One common way to measure viruses and proteins using biosensors is the plaque-forming units PFU/mL; for COVID-19, this could be up to 16 PFU/mL [185]. The method used to create the dataset is based on the assumption that there are available biosensors to measure three different parts of the virus: viral proteins S, viral proteins N, viral genetic material RNA, and provide readings measured by PFL/mL. The dataset was created based on three input values: viral protein

S, viral protein N, and viral genetic material RNA. The higher the PFU of each input, the higher is the COVID-19 infection rate in a certain area. The AI outcomes was classified into three different categories: low risk, medium risk (needs attention), and high risk (needs immediate action). Table 6.2 presents an example of the test dataset, where the numbers in the table are representative of those found in [185, 186, 187].

Table 6.2: Example of the created test dataset.

Viral proteins S PFU/mL	Viral proteins N PFU/mL	Viral genetic material RNA PFU/mL	Output
1.28	13.12	9.92	1
1.12	8.80	8.00	0
5.92	16.00	10.24	2
6.08	13.12	2.24	1
0.64	0.96	13.12	0
2.72	0.96	1.92	0
12.16	14.40	14.08	2

6.6 Results and discussion

The following subsections will provide details of the results along with the discussion.

6.6.1 System latency

The system was tested while mining using different numbers of WN (1 to 12 WNs). Then the overall latency of the system for each test was measured. Figure.6.6 shows the average L. From the figure, it can be seen that as the number of WNs participating in the block mining process increases, the average L decreases. The overall latency was lowered from over 40 min when only one WN was used to approximately 4.3 minutes when the total WN utilised to mine one block was 12. If the number of WNs was increased, this time could have been reduced to less than one minute.

6.6.2 AI-accuracy

In terms of the AI engine accuracy, the system was first trained on 70% of the dataset, using R-pi, and produced a prediction accuracy of 97%. Then the system was tested on a stand-alone R-pi device, not connected to the blockchain network, using 15% of the dataset, resulting in a prediction accuracy of 95%. In the final test, the system was deployed on the blockchain network using one AN. Then, it was tested in three rounds. With each round, a 5% of the remaining dataset (the data were sent over the network as blockchain transactions, as described by the data flow in Fig.6.4) was used. All three rounds of the unseen dataset resulted in the same prediction accuracy of 95%, which is the same as when testing using the stand-alone system. This shows

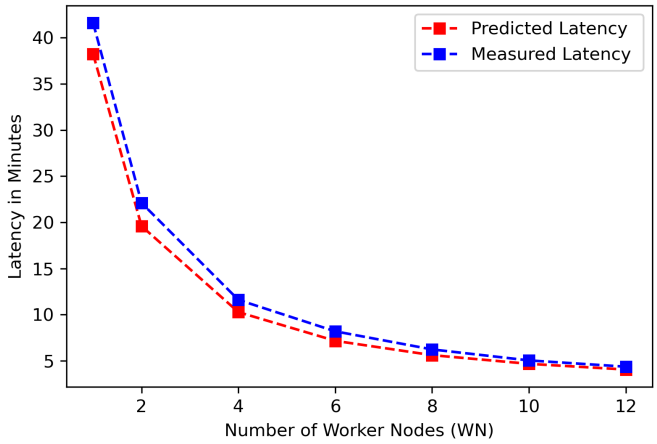


Figure 6.6: Measured and predicted system overall latency.

that utilising blockchain for better data security did not affect the AI-engine accuracy; Fig.6.7 shows the accuracy for both tests compared to the training.

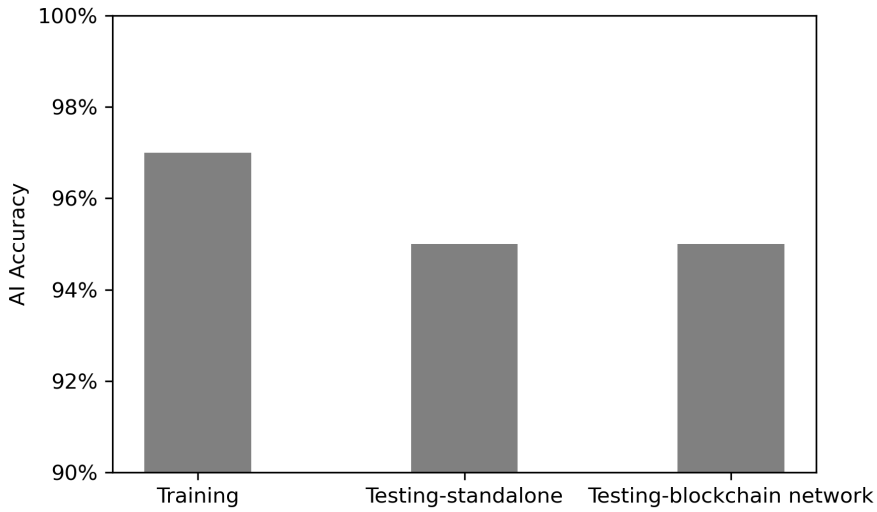
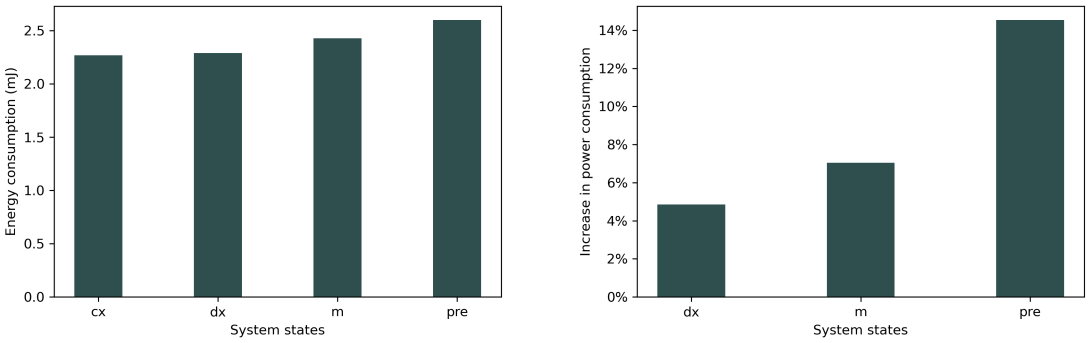


Figure 6.7: Accuracy of the AI-expert engine.

6.6.3 Power cost

An important aspect of the proposed system is the impact on the battery and power sources of devices. To investigate this impact, the power consumption during different system states was measured, including connectivity (cx), data exchange (dx), mining

(m), and prediction (pre). Figure.6.8a shows the consumed power by the R-pi during each of these states. It is clear that the impact of using the device for mining or hosting the AI engine is minimal, as most of the power is consumed when the system is running and connected to the Wi-Fi without performing any task. This is clear in Fig.6.8b, as it shows the percentage of the power increase when the system is utilised to perform blockchain mining, data exchange, or AI prediction. When using the R-pi for AI prediction, the power increase was 14%, and this increase was 7% when utilised for blockchain mining. However, in a network where the available number of WNs and ANs is a few hundred or even thousands, such an impact can be eliminated, as there would be more than enough nodes to perform different tasks in the network. This means that a device may spend a day without performing any task.



(a) (b)

Figure 6.8: Energy and power measurements. (a) Average energy consumption of the system states. (b) Power cost when the system is in the dx, m, and pre states compared to when the system is in the cx state, that is, the reference state.

6.6.4 Discussion

The proposed architecture provides a platform that is secure, robust, and effective in terms of power and accuracy to support AI-enabled IoT applications at the edge. The system is able to ensure continuous AI prediction, thus eliminating a signal point of failure, providing governmental entities and organisations with processed data and outcomes for better decision-making. It ensures data integrity by validating and securing all AI data (inputs and outcomes) using a secure, decentralised, and transparent blockchain platform.

Compared with other related studies, the proposed architecture provides a platform that is capable of ensuring AI data integrity through validation and transparency, allowing the deployment of a robust and redundant AI-engine without any impact on its accuracy. It achieves this by utilising edge devices and IoT end devices without a

substantial impact impact on the power of these devices. Table 6.3 shows a performance comparison with the important related works. The author acknowledge the difficulty of direct comparisons to other work due to differences in the presented assessment criteria. Furthermore, individual blockchain solutions can be tuned to enhance performance for a specific application.

Table 6.3: Performance comparison between this work and important related works.

Work	AI data integrity through validation and transparency.	AI engine Robustness and Redundancy	AI accuracy	Power Cost	
				AI performing prediction	Blockchain mining
This work	Yes. All input data and AI outcomes were validated and added to the blockchain	Yes. The AI can be deployed across different ANs, and each node can produce its own outcome.	95%	Power cost increased by 14%	Power cost increased by 7%
[134]	Partially. Only processed AI data validated on the blockchain	Yes	Not given	Not given	Not given
[135]	Partially. Some local processing at the consumer electronic device layer	Yes	Not given	Not given	Not given
[142]	No. Only trading data are recorded in the blockchain.	Yes	Up to 99%	Not given	Not given
[143]	Yes	Partially. Used centralised deep learning-based cloud	Not given	Not given	Not given
[146]	Yes	Partially. It relies on the cloud to convert and mine full blocks.	Not given	Not given	Not given
[147]	Yes	Yes	95.18%	Not given	Not given
[180]	Not all the data are validated. Local model trained on local data.	Partially. IoT devices rely on edge servers and the cloud.	Over 70%	CPU usage of 30%. No power consumption measured	CPU usage of 30%. No power consumption measured

6.7 Summary

In summary, a system that has the capability to combine the advantages of four important technologies—edge computing, blockchain, IoT end-devices, and AI—in one platform was proposed, designed, developed, implemented, and tested. This system incorporates the security advantages provided by blockchain to offer a publicly available platform that integrates the intelligence advantages provided by AI into an edge layer to facilitate a secure architecture capable of sensing, analysing, thinking, and producing actionable outcomes.

The results showed that the system provided reliable accuracy in terms of the AI prediction of COVID-19 occurrence in sewage water at an acceptable system latency for such an application. The results and analysis of the impact on the devices' power sources showed that it is possible to use low-cost and low-power devices to accommodate the requirements of AI and blockchain in a network of a few hundred nodes.

Chapter 7

Adding hardware security into HDPoA protocol

In this chapter, the viability of integrating a hardware security module (HSM) and a hardware wallet (HW) into HDPoA based IoT-blockchain will be evaluated. This chapter will provide Initial assessments of the energy consumption of nodes that are equipped with HSM and HW along with re-analysis of HDPoA security. Finally, an example application, namely, secure community energy trading based on the deployment of an HDPoA-based blockchain platform where all nodes are equipped with an HSM and HW will be presented and discussed.

7.1 Introduction

While the integration of blockchain into IoT (IoT-blockchain) can enhance the IoT security, yet due to the nature of IoT devices some vulnerabilities, such as those related to the physical security, key generation, management, and storage, and the lack of trust in host systems can exist. Hardware security can provide protection to the IoT devices and enhance the overall security on the IoT-blockchain platform and the data held on it. This make the evaluation of the viability of the integration of hardware-based security into IoT-blockchain and analyse its performance and resilience an important topic.

7.1.1 Problem statement and background

Due to the nature of IoT constrained devices, they can be vulnerable to various attacks. This means when these devices join a blockchain network this network will inherit any potential vulnerabilities that these devices have.

7.1.2 Potential vulnerabilities within IoT-blockchain

Some IoT devices lack the computation and storage resources to be able to host adequate security and protection mechanisms locally, they became vulnerable in term of security. One potential vulnerability is associated with the possibility of credential stealing, where an attacker can access a node's private key and compromise its identity. This is particularly dangerous when dealing with IoT-blockchain platforms where a node's private key can be used to sign data and make it legitimate. Data also can be vulnerable to modifications and confidentiality attacks; such attacks can damage trust in the blockchain platform and the data stored on it.

IoT devices are often left outside in an insecure place where attackers can physically access them and compromise their credentials and identity. They also lack adequate protection for their root file systems, making them vulnerable to attacks that might target proprietary software or sensitive data. The lack of a proper authentication mechanism among IoT devices makes it difficult to ensure the security of data and devices' software. Regarding IoT systems, some devices tend to use the sleep mode, where a device only wakes up to perform certain tasks triggered by an event or time; this means it is difficult to track the time accurately.

Another security issue within IoT systems is key generation and management. Most devices do not have proper key management in place. This raises concerns regarding what algorithms are used, the strength of the keys chosen, their expiration timeframe, how the key was generated, and whether it is prone to information leakage. Storing the key is another issue, specifically, what type of storage is in place and its trustworthiness. There may also be concerns regarding what type of crypto suites are in place, whether they have been confirmed to a well-known standard, and whether there is a true random number generator (TRNG) in place. These are important issues that must be addressed when dealing with IoT-blockchain applications.

Different authors have integrated the hardware security into IoT systems to solve some of these vulnerabilities. The work by [188] evaluate the use of hardware security to strengthen IoT devices against trojan and side-channel analysis attacks. The works by [189] and [190] proposed similar solutions based on hardware security to ensure the security of IoT medical devices against attacks such as side-channel analysis. Similarly the authors of [191] utilised hardware security to compact power analyses attacks on devices. The work of [192] has proposed the use of blockchain and hardware security to secure IoT systems. The authors of [193] have studied securing industrial IoT using the ARM TrustZone and the security controller. Based on their study, they proposed a hybrid solution to maximise the security gain from both.

In term of evaluating the energy and security impact of integrating hardware security into IoT-blockchain applications, none of the available literature has evaluated that. So there is a need to perform a study and an evaluation of the impact of hardware-security on the security of IoT-blockchain applications. There is also a need for analysing the impact on the performance in terms of energy consumption and events related latency.

7.1.3 Contribution of the chapter

The contribution in this chapter can be summarised as:

- Evaluate the usability of integrating hardware security modules and hardware wallets into IoT-blockchain platforms.
- Provide a performance evaluation regarding the impact of energy consumption following this integration and an analysis of the nodes' security and resilience to attacks while using a HSM and a HW were conducted.
- Finally, an example application, secure community energy trading, is proposed and discussed based on secure IoT-blockchain technology.

7.1.4 Organisation of the chapter

The rest of this chapter is organised as follows: section 7.2 presents the IoT-blockchain hardware security requirements, followed by the proposed solutions in section 7.3. In section 7.4, an example application called community-based secure energy trading was described, followed by the security analysis in section 7.5. The testing and evaluation was provided in section 7.6, followed by the chapter summary in section 7.7.

7.2 IoT-Blockchain hardware security requirements

Adding hardware security to an IoT-blockchain applications can enhance the security of these applications and protect them against attacks. Different security elements can be used to provide hardware security. Trusted platform module (TPM) is a secure single-chip coprocessor that provides different security functions [194]. TPM offers users different security features, such as allowing users to secure content without the need for a software-based operating system. It offers the user the ability to encrypt the hard disk, thus securing sensitive information. TPM provides users with a hardware authentication ability [194].

Hardware security module (HSM) is a hardware-based cryptographic module that is confirmed to the FIPS 140-2 standard [195]. It allows for secure generation, storage, use, and management of cryptographic keys and passwords. They are tamper-resistant, which helps ensure the physical security of devices. They allow for data encryption and signing, ensuring data confidentiality and integrity [196]. HSMs can be power-independent. For example, they can be powered using small batteries, allowing them to include a real-time clock for accurate time and date tracking and timestamping. Some modern HSMs, such as HSM6 from Zymbit [197], offer dedicated HW that can be used for key generation, management, and storage, making it ideal for securing devices for IoT-blockchain applications. All of these features make hardware security based on HSM solutions an ideal solution for IoT-blockchain applications.

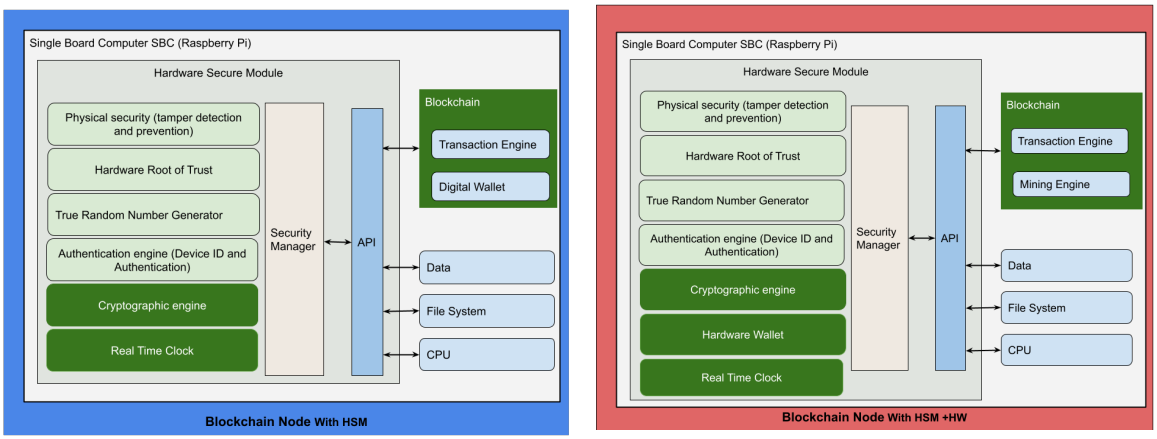
7.3 Proposed solutions

To address the above issues and vulnerabilities, the integration of the following hardware security features into IoT-blockchain applications are proposed:

- Physical security (tamper detection and prevention), utilising reliable sensors that can detect any physical activity targeting the device and subsequently protect the system and its credentials (e.g., by destroying the keys).
- Encrypting the root file system to protect the proprietary software, sensitive data, and WiFi credentials, keeping them immutable and protected against cloning.
- Using an authentication to secure the host system, for example by using a fingerprint of that system's components that is used to bind a specific hardware security module and the host computer, forming an immutable and permanent ID of the host system. This ID would be used each time when booting the system and/or at random times; any changes to the system's components would result in the failure of the authentication process.
- Using a cryptography engine where different services can be provided by this engine. First is the TRNG, the process of using a physical process to produce a random number. TRNG is used as a seed to generate private keys. The more random the creation of the sources of the private key, the more it is secured. Second, it provides users with the ability to encrypt files and data using some of the secure encryption algorithms such as ECC NIST P-256 (secp256r1), ECDSA (FIPS186-3), and AES-256 (FIPS 197).
- Using a hardware-based real-time clock where transactions(Tx) and blocks must include a timestamp for their creations. The timestamp helps to secure the blockchain network against double-spending attacks as only the last Tx is the only one count; the previous double-spending attempt is discarded [13].
- Using a secure offline device such as HW that can be used to generate, manage, and store cryptographic keys and are convenient to use when signing Tx or encrypting data on blockchain platforms.

Based on their computation and storage abilities, devices can have different hardware security features. In HDPoA, as discussed above, there are three node categories: FN, HN, and PN. Based on the nodes' roles in the consensus mechanisms, there are two different types of nodes: AN and WN. As shown in Fig.7.1a the security requirements for the HN was identified, which includes tamper detection and prevention, device authentication and identification, hardware root of trust and key storage, a TNRG, a real-time clock, and cryptographic primitives.

Only FNs can become ANs as they have sufficient computation power and storage capabilities to manage the consensus mechanisms and ensure the security of the network. This means that ANs are the heart of the security of the IoT-blockchain network,



(a) (b)

Figure 7.1: Hardware security solutions. (a) For hybrid nodes - using HSM. (b) For full nodes - using HSM and HW.

making it important for each node to have its own HW. Figure.7.1b shows the hardware security requirements for any FN nodes and, subsequently, ANs, which include tamper detection and prevention, device authentication and identification, hardware root of trust and key storage, a TNRG, a real-time clock, cryptographic primitives, and HW.

Figure.7.2 shows the HDPoA consensus mechanism with hardware security integrated into the nodes.

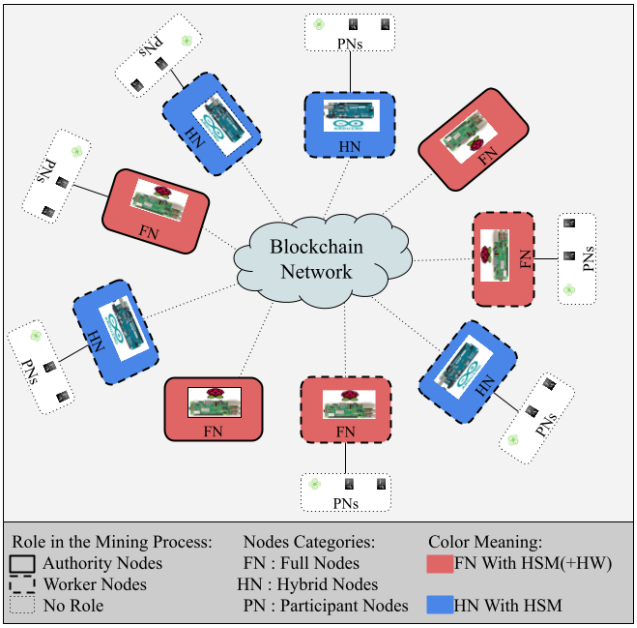


Figure 7.2: HDPoA consensus algorithm - overview

7.4 Example application: community-based secure energy trading

One application that can benefit greatly from integrating hardware security into IoT-blockchain nodes is community energy trading. Figure.7.3 shows the proposed design of IoT-blockchain energy trading platforms. As shown in the figure, different entities, homes, microgrids (MGs), energy storage, and factories can communicate with each other over a secure blockchain where all nodes are equipped with an HSM and their own HW.

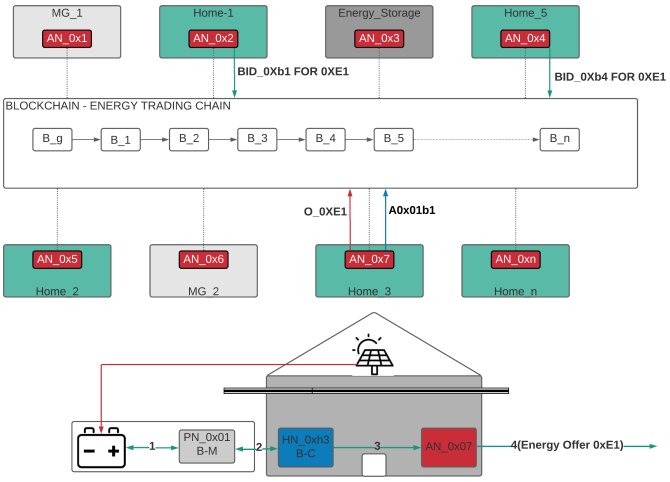


Figure 7.3: Example application - overview of secure community energy trading

In Figure.7.3 the home number 3 which is node AN 0x07 is assumed to have surplus energy that was generated using its solar panel. The node will announce the details for this surplus energy on the blockchain where other entities can securely receive trusted data and bid for the energy to buy it.

- First the PN (sensor) that monitors the energy storage once there is surplus energy sends a notification to HN_0xh1.
- Once it has received the message, HN_0xh1, which controls the battery or energy storage, validates the signature and sends an encrypted message (using HSM-4) to AN_0x07.
- Once AN_0x07 receives the message, it decrypts and validates its signature.
- Then AN_0x07 will create a blockchain Tx with ID O_0xE1 to advertise the amount of energy.

- Tx O_0xE1 will then arrive at all ANs pools and the node responsible for the next block mining process (AN_0x01) will validate all Txs, including O_0xE1, and add them to a new block.
- Then, AN_0x01 will manage the mining process according to HDPoA. Once the block has been mined, AN_0x01 propagates it to all ANs.
- Once the block has arrived at all ANs, the nodes that are interested in buying energy submit their bid. In this case, AN_0x02 will submit a bid Tx with an ID 0xb1 and AN_0x04 will submit a bid Tx with an ID 0xb4. Both nodes encrypt their bid using AN_0x07 public key (using HSM-6). Both Txs will then arrive at the next miner's pool (AN_0x03).
- Then, AN_0x03 will repeat the same process of Tx validation and block mining.
- Then, the new block containing both bids will be propagated to all ANs.
- Once AN_0x07 receives the new block, it will decrypt both bids and evaluate them and, if it accepted any of them, in this case, it accepted the bid from node AN_0x02, it will submit an encrypted acceptance Tx (A0x01b1).
- This transaction will then arrive at the next miner's pool (AN_0x05), which will then perform the mining process and produce the new block.
- Finally, the block that contains Tx A0x01b1 will arrive at AN_0x02 and energy transfer will begin.

7.5 Security analysis

In section 4.4 a security analysis and risk assessment of HDPoA using the NIST SP-800-30 standard [156] were provided. The results of these analysis and assessment are shown in Table. 4.3. In this section, security analyses and risk assessments will be provided regarding the hardware secured HDPoA nodes.

7.5.1 Compromised AN

An attacker can access the AN's private key and use it in a malicious way. ANs are responsible for managing the mining process, propagating new blocks, and validating new blocks; they are an integral part of network security. Different security measures are in place to defend against such attacks. First, all AN nodes should be equipped with an HSM and HW, ensuring the security of their private keys. Secondly, HDPoA consensus deploys a mechanism where nodes are only allowed to propagate a block every $\frac{N}{2}+1$ blocks; thus, at any time t there are, at most, $N-(\frac{N}{2}+1)$ ANs allowed to propagate a block. Any block that is propagated by any AN during its turn will be

validated by other ANs and if the majority of ANs agree that it is not valid, then the that AN can be removed from the AN category and subsequently from the network.

While the impact of a successful attack on an AN is *high*. As the node is equipped with an HSM and HW, the likelihood of an attacker compromising an AN is *low*; thus, the residual risk is *low*.

7.5.2 Compromised WN

A WN can be from either FN or HN categories, meaning some nodes will be equipped with HSM and other with HW. This will limit the attacker's ability to hijack a WN and compromise its credentials, including its private key. This along with the fact that these nodes work under the supervision of ANs that validate the work produced by any WN means the likelihood of an attack is *low*. If successful the impact of such an attack is *very low* as it will be limited and contained by the ANs. This means that the residual risk is *very low*.

7.5.3 51% attack

By design, it is not possible to carry out the 51% attack that is associated with controlling the majority of the hash power on HDPoA. This is because it implements two layers of security when performing the hash calculation: the first hash calculation is divided among the WNs and the second AN in charge of the block mining process always validates the WN hash's work, meaning no single node can control the hash of the network, regardless of its computation ability. However, there is the risk of an attacker attempting to control the majority of ANs (i.e., control 51% of ANs on the network). The first one is difficult as all nodes will have secure storage for the keys in the form of an HW and each node will have its security enhanced by the presence of an HSM, making such an attack very difficult. While the deployed measures make the likelihood of such an attack *low*, if successful, the impact of this attack is *high*, meaning the residual risk is *low*

7.5.4 Double spending

Double spending is an attack that is associated with data consistency and deceiving others about a transaction's state. It occurs when a node or user tries to spend the same digital currency twice [198]. In HDPoA, as all transactions are timestamped using the hardware-based real-time clock, only the last transaction is validated and all other transactions that have previously been submitted but not validated are disregarded. Another defending mechanism against this attack is the fact that all transactions are validated twice: first by the AN that currently manages the consensus and then by all other ANs which, after that transaction, will be final and cannot be deleted or modified. While the motivation and likelihood of such an attack can be considered

high, the impact can be controlled and eliminated in HDPoA. This means that its impact is *low* and the residual risk is *low*.

7.5.5 Data attacks

Ensuring the security of the data from its source to its destination is an important aspect of IoT-blockchain applications. Data can be vulnerable to different attacks that can compromise its integrity and freshness. By integrating an HSM into the blockchain, nodes have a cryptographic engine that allows them to sign data to protect its integrity in transit and at rest and allow other nodes to easily verify this. It can also encrypt the data (payload of transactions) to ensure confidentiality. The likelihood of attacks to compromise data integrity and confidentiality within IoT-blockchain applications is *high*, however, due to the measure provided by the HSM and the fact that blockchain is an immutable ledger that protects data by design, the impact of such attacks would be *low*, meaning that the residual risk is *low*.

7.6 Testing and evaluation

7.6.1 Expermint setup

Figure 7.4: Two R-pis, one equipped with the HSM4 and another with the HSM6.

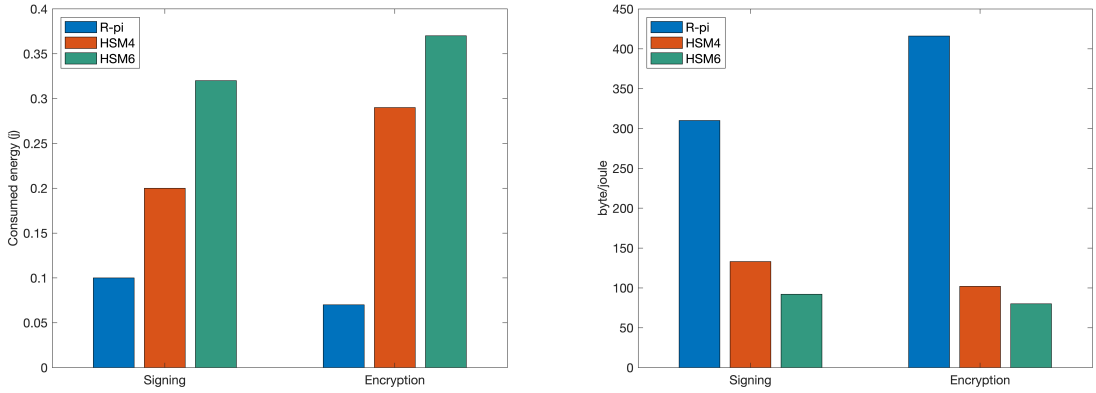
For the initial testing, two different HSMs will be used. The HSM-4 from Zymbit [199] which has all the security requirements that were identified for HN and the HSM-6 [197], which on top of the security features provided by the HSM-4, has its own dedicated HW, making it ideal for FN (i.e., AN). Both devices were tested as part of the in-house built blockchain platform using Raspberry pi (R-pi). Two different types of nodes were deployed on the blockchain platform; HN with HSM-4 and FN with HSM6, the FN were used as AN, Fig.7.4 shows the two nodes that were deployed. The test was carried out while both nodes are used to sign and propagate transactions, and

while encrypting some of the payload of transactions. The main aim with this testing was to evaluate its usability and its performance in terms of energy consumption and events execution time (latency).

7.6.2 Energy consumption

The energy consumption was measured when signing and encrypting data using R-pi, the HSM4, and the HSM6, results are shown in Fig.7.5a. From the figure, it is clear that the HSM-6 consumes the highest amount of power; this will have a limited impact on FN as these nodes will be utilized at the edge and will have access to an adequate power source. Additionally, the HSM-6 can be powered using dedicated small batteries.

Based on the measured energy consumption, the number of bytes that can be signed or encrypted was evaluated as a function of energy consumption. Figure.7.5b shows the total number of bytes that can be signed and encrypted based on 1 joule of consumed energy. While R-pi produces more encrypted or signed bytes per 1 joule, the security provided by hardware-based encryption and signing is worth the amount of energy consumed, particularly when an AN will be utilized at the edge layer, close to a reliable power source.



(a) (b)

Figure 7.5: Consumed energy. (a) Average energy consumption in (j) when signing and encrypting 30 byte of data. (b) Evaluated number of data (byte) as a function of energy (j).

7.6.3 Latency

The time needed to sign or encrypt a data of size 30 bytes was measured for R-pi, HSM4, HSM6, and results are shown in Fig.7.6. Both devices were communicating with the R-pi over I2C, meaning there is overhead due to the nature of the data rate when using I2C. Another important aspect of these devices is that they do not have

an encryption accelerator. This has resulted in both of them needing longer times to sign or encrypt data compared to the R-pi. Depending on the application in use, this delay can be neglected in exchange for the added security provided by these devices, ie. in the proposed example applications security is more important than delay.

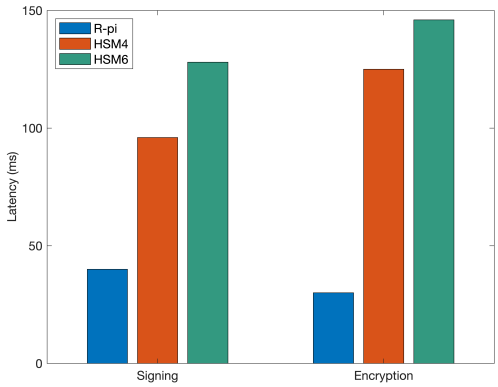


Figure 7.6: Latency of data signing and encryption.

7.7 Summary

In this chapter, the integration of HSM and HW into IoT-blockchain applications was explored and tested. The results showed that an HSM and HW can be utilised by the FN; an HSM can be utilised by an HN in the deployed HDPoA with minimal impact on these devices' power sources. The security analyses indicate that the added security features from this integration have a significant impact on reducing the risks associated with different attacks.

An example application was discussed that can greatly benefit from this integration, however, full deployment and testing of this application based on the securely enhanced HDPoA must be accomplished as a future research effort and further performance evaluations must be conducted.

Chapter 8

Conclusion and Future Works

The first objective of this thesis was to study and evaluate the integration of blockchain-IoT application using a real-world use case and to provide a performance analyses of the system latency, network synchronisation and stability, and energy consumption.

In **Chapter 3** of this thesis the performance evaluation of a real-world IoT-blockchain applications was conducted by providing a system model that predicts the system end-to-end latency, the energy consumption, and the stability of the network and the nodes synchronisation. The study was validated by practically implementing the flood detection system based on IoT-blockchain. The measured results are in line with the numerical analyses. Based on the conducted tests and analyses, the key findings can be summarised as follow:

- First when implementing the Ethereum clique protocol using PoA over a 3G cellular network the BPs of 1, 2, and 3 seconds are not recommended due to nodes synchronisation issues and longer block and transactions propagation delay. When implementing over Wi-Fi, while it is possible to implement the 1-second BP, it carries a lot of risk in terms of synchronisation and data freshness. However, in other application, such as tracking and traceability where the data will not be used to coordinate and automate the decisions, such BPs can be used.

- Ethereum PoA depends on the trusted nodes and their honesty in mining and propagating blocks. This renders it a more central network, which goes against the concept of decentralised blockchain. Many applications within IoT, however, require added security and privacy. Other consensus algorithms such as PoW provides a security consensus when implementing blockchain as a public network but requires more energy to find the target hash for each block, and this makes it not an ideal fit with its current form within the IoT realm.

- Another finding of the study, is that it is important to consider the size of the block when building IoT-blockchain network. Based on the conducted study, the latency of the events that are related to the block size such as announcing

the block, and importing the block is related to the block's size. As the block size increases the time the IoT devices take to execute them increases as well. For example, in the conducted study, the measured time for importing a block of size 20 kB was about 0.5 s over the Wi-Fi network and about 0.6s over the 3G network, and the time for importing a block of size 100 kB was about 3.3s over the Wi-Fi network and 3.9s over the 3G network. This means more energy consumption and could shorten the life of these devices' batteries

- By studying the implementation of blockchain networks over two different communication links, it is safe to say that Wi-Fi connectivity provides a reliable and fast link. Nevertheless it is not available all the time for many IoT applications. In this study we showed the possibilities of implementing blockchain over a 3G cellular network. However 4G and 5G networks are better in terms of latency. The authors of [200] who provided a comparison measurements between 3G and 4G that includes one way latency measurement showed that 4G is outperforming 3G in all measured parameters. For example, the 4G throughput tests resulted in maximum of more than 28 Mpbs, while the 3G resulted up to 4.8 Mbps. However, 3G provides much larger coverage making this technology difficult to neglect just yet. The 5G technology brings a great potential for IoT-blockchain implementations. Some IoT applications require low latency and higher data rate, which are two strong advantages of 5G, which will help facilitate this integration. IoT, blockchain and 5G together have great potential, while 5G provides a low latency connectivity cover for IoT devices, blockchain can be integrated to eliminate centralised third-party entities and ensures the protection of user and transaction data. This will potentially be a good integration as each part strengthens the other.

The second objective of the thesis was to design a consensus mechanism that is IoT centric, public, secure, and has limited impact on the device's individual power.

To achieve this aim, the HDPoA consensus mechanism that is suitable for implementation within IoT systems was proposed in **Chapter 4**. HDPoA is based on PoA and integrates PoW as an underlying security layer to make it a permission-less mechanism. A complete performance analysis of the most important metrics was conducted. The the key findings of the testing and evaluation of HDPoA mechanism can be summarised as follow:

- The results of HDPoA energy consumption tests successfully demonstrated that low-cost IoT devices are able to participate in the block generation process by carrying a small amount of the mining task without substantial impact on their batteries. This is a result of the way HDPoA was designed where it introduces an added layer of security through the trusted authority nodes allowing for the distribution of the mining computation among different trusted and less trusted WNs. HDPoA is a permission-less consensus where any node can freely join the

network. This means more WNs are available to participate in the block mining process which in turn, as the network grows in size to may be more than 1,000 nodes, can imply a WN might spends weeks without performing any mining task (hence, less impact on the battery). This allows IoT end-devices to be part of the blockchain platform and access services and data provided on the platform without worrying much about their batteries or energy sources.

- HDPoA is a permission-less consensus mechanism based on the concept of scalable work, where nodes joins freely as a WN and then perform a useful work on the network to build up their trust. This means WNs can be utilised to carry out different works other than participating in the mining process, for example be a part of an AI system, collect and provide data, perform data processing locally, and many more. This opens the door for its adaptation to secure blockchain platform that can be used to support AI implementation, data mining, and data trading.

- Within public blockchain platforms the ***transaction finality*** is an important aspect for system latency and data security. In HDPoA, transactions confirmation only requires one block to ensures the ***transaction finality***, providing lower latency for IoT applications on the network compared to the approximately six blocks required in the PoW approach for bitcoin. This is because HDPoA integrate PoA which does not require more than one block for confirmation [156, 160], hence, mitigating the PoW long confirmation time.

- In terms of throughput bitcoin which utilises PoW as a consensus protocol has an average of 5-7 Tx/s [162]. PoA is totally controlled by the network owner and can implement a block period of as low as one second. This means using B_{size} of 1Mbyte and Tx_{size} of 200 byte can result in throughput of more than 5000 Tx/s. However, in an IoT network where devices have limited computation resources nodes will spend more than100 seconds or even a few minutes importing a block [156]. This means this throughput will not be achieved in practice.

- The security analyses and the risk determination showed that HDPoA compared to PoA is very resilient against the DoS attacks due to the fact that it is a public consensus and there is no limit on the number of AN allowing for wide choice of nodes to manage and run the mining process, thus making it difficult for an attacker to target all of them at the same time. The analysis showed that HDPoA eliminates the 51% attacks that are associated with the control of the hash power that PoW is vulnerable to. This is because it implements a mechanism where ANs control the mining process and the work is distributed amongst multiple WN and only valid solutions are accepted.

- In permissionless consensuses, the difficulty is usually adjusted (increased or decreased) according to the available hash power to maintain the security of the

blockchain platform. Some previous studies, such as [69, 71, 149], have integrated PoW into IoT-blockchain systems by implementing the concept of nodes with high trust or credit scores mining blocks at *low difficulty*. In HDPoA, as the number of worker nodes increases, the difficulty can be increased, allowing for incorporation of the full security advantages provided by PoW without compromising the decentralisation of the network as it allows for unlimited number of nodes to be promoted to the AN category as long as they are behaving honestly. For example, with a total of 20 WNs, the network was able to increase the mining difficulty up to 16 and mine blocks.

The third objective of the thesis was to design, develop and implement a secure architecture that can be utilised to support the implementation of a DAI engine that can exploit the available distributed IoT hardware and ensures the data integrity within the IoT systems.

In **Chapter 5** a secure architecture was developed and tested where an DMLP engine deployed using different IoT devices where each one of these devices host one neuron or more of the DMLP engine. A blockchain protocol that includes transactions and block formats was designed and tested to allow for the handling of the DAI related transactions securely. The key findings of studying, testing, and evaluation this architecture can be summarised as follow:

- The architecture is based on a distributed and decentralised approach, thus allowing for the exploitation of the resources provided by large-scale distributed IoT systems.
- It is built around the on-chain processing of transactions. This means it is based on a trusted method that makes DAI predictions traceable and easy to understand, allowing users and organisations to determine how and why any decisions were made. This is because each neuron once it finishes the processing of the received data will share the results on the blockchain using the transactions that are capable of carrying AI related data.
- It integrates the IoT devices at the sensing layer as partially part of the blockchain platform to ensure the integrity of the data collected by these devices by including them directly into blockchain transactions that are validated and stored into the blockchain. This means the DMLP engine only have an access to secure and trusted data, yielding a better decision-making process
- It utilises HDPoA consensus which means it is resilient against DoS and 51% attacks. HDPoA encourage the nodes to perform useful work on the network in order for them to be promoted to AN category. This means with the presence of large number of WNs nodes deployment of redundant DMLP can be easily achieved.
- The distributed nature of the architecture allows for the implementation of different DAI engines over the IoT devices that all can access the same

trusted data and each one of them implementing its own AI processing mechanism (i.e different activation functions or different concept of the AI such as convolutional neural network (CNN)). Then each engine will share its outcome over the blockchain network. This would allow for the implementation of the AI-consensus on one outcome, thus more AI assurance regarding decision-making process.

The fourth objective of this thesis was the design and development of an architecture that combines IoT, AI, blockchain, and edge computing to create a system that is secure and able to monitor the environment and collect data, analyse it, and produce an outcome to support an AI-enable IoT applications.

In **Chapter 6** an architecture was designed, developed, implemented, and validated experimentally using 14 low-cost, flexible IoT hardware entities based on the Covid-19 detection in sewage water use case. It combines blockchain, AI, IoT end device, and edge layer into one system. This system have the ability to monitor the environment, collect data, analyse it, process it using an AI-expert engine, provide predictions and actionable outcomes, and finally share it on a public blockchain platform. For the use-case implementation, the pandemic caused by the wide and rapid spread of the novel coronavirus COVID-19 was used to test and evaluate the proposed system. Based on the testing of the use case and the result and the analysis of the architecture the key findings are as follow:

- The system is able to ensure continuous AI prediction, thus eliminating a single point of failure, providing governmental entities and organisations with processed data and outcomes for better decision-making.
- The architecture integrates a new blockchain protocol for handling the communication aspects of the system and securely handle the AI-related data through a new transaction and block formats.
- Compared with other related studies, the proposed architecture provides a platform that is capable of ensuring AI data integrity through validation and transparency, allowing the deployment of a robust and redundant AI-engine without any impact on its accuracy.
- The architecture is suitable for integration into the edge. It allows the IoT devices to be part of it by implementing the concept of the hybrid nodes that can submit transactions and access the stored full chain through the full nodes.
- The results showed that the system provided reliable accuracy in terms of the AI prediction of COVID-19 occurrence in sewage water at an acceptable system latency for such an application.

– The results and analyses of the impact on the devices' power sources showed that it is possible to use low-cost and low-power devices to accommodate the requirements of AI and blockchain in a network of a few hundred nodes.

The final objective of this thesis was to evaluate the viability of integrating a hardware secure module (HSM) and a hardware wallet (HW) into IoT blockchain applications by adding HSM and HW to the nodes that are part of HDPoA-based blockchain network.

Chapter 7 provided an evaluation of the viability of integrating HSM and HW into IoT blockchain applications and provide a performance analyses in terms of power consumption and relevant security evaluation. Nodes within HDPoA-blokckchairn based on their roles within the consensus mechanism are equipped with either HSM or both HSM and HW. In the chapter an analysis of the nodes' security and resilience to attacks while using a HSM and a HW were conducted. The HSM6 from Zymbit was added to the ANs (i.e only FNs) and HSM4 from Zymbit was added to the WNs. A performance evaluation regarding the impact of energy consumption following this integration and an analysis of the nodes' security and resilience to attacks while using a HSM and a HW were conducted. A community energy trading use-case was described including how nodes would utilise the hardware securities to create and receive transactions. This was the final work of the research and due to the limited time only initial testing and results were carried out. The key findings of this initial testing and evaluation are as follow:

– The results showed that HSM6 and HSM4 can sign more than 130 bytes of data and more than 100 bytes of data, respectively while consuming one joule. The HSM6, which included the HW as well, produces more than 90 bytes when signing and more than 80 when encrypting for the cost of one joule.
– Both HSM4 and HSM6 where slower when performing the signing and encryption of 30 bytes of data compared to the R-pi. HSM4 needed 96 ms to sign and 125 ms to encrypt while HSM6 needed 128 ms to sign data and 146 ms to encrypt it. This is can be related to the fact that both devices were communicating with the R-pi over I2C, meaning there is overhead due to the nature of the data rate when using I2C and both do not have an encryption accelerator.
– The evaluation of the results and the security analysis showed that nodes from both categories, AN and WN, within HDPoA can benefit from the added security and trusted provided by both the HSM and HW in exchange for small amount of energy, particularly when an AN will be utilised at the edge layer, close to a reliable power source. Different security advantages provided by the integration of HSM6 and HSM4 for example:

    - These hardware secure devices provided the node with the ability to generate, manage, and store cryptographic keys and are convenient to use when signing Tx or encrypting data on blockchain platforms.
    - They allow nodes to detect any physical activity targeting the device and subsequently protect the system and its credentials.
    - They can deter attackers especially those related to credential theft and node compromisation.
    - They provide a hardware-based real-time clock for nodes to utilise to add a timestamp when creating transactions (Tx) and blocks. The timestamp helps to secure the blockchain network against double-spending attacks
    - They provide an ability for encrypting the nodes' root file system to protect the proprietary software and sensitive data.

In many IoT applications blockchain can provide great benefits, for example to resolve the issues surrounding the use of a central entity for better system performance by eliminating single point failure, and provide means for devices and user identification and authentication and preserve data integrity. An example of these IoT applications is tracking and traceability within both supply chain and healthcare systems. The tracking and traceability within healthcare can greatly benefit from immutable systems such blockchain to protect against medicine and drugs counterfeiting, to monitor the environmental conditions of pharmaceuticals including donated blood. Also, within industrial IoT blockchain can be utilised for better machine automation - especially ensuring decisions executed by machines are based on true data.

To conclude, in this thesis many contributions were provided that are important in helping researchers and organisations to integrate blockchain into IoT systems. Thus allowing them to greatly benefit from a resilient, distributed, decentralised, self-managed, robust, and secure IoT-blockchain systems.

8.1 Limitations

With any research there are always limitations with proposed works and solutions and this thesis is no different. The limitations in the works provided in this thesis are as follow:

- The first limitation is related to the scalability of HDPoA in term of throughput (transactions/seconds). While HDPoA compared to PoW as a public consensus can provide higher throughput yet within high scale applications where thousands of nodes generate many transactions it is ability to process high volume of transactions (thousands per a second) can be limited in such applications.

This limitation does effect the proposed architectures in this work as both, the architecture to support DAI implementation and the architecture to support AI-enabled IoT applications at the edge, utilises HDPoA as the consensus mechanism for securing them.

- The second limitation is regarding HDPoA's ability to allow the end devices to access the full chain directly, thus allowing them to benefits from the available data. This is a limitation resulted from the fact that some devices within IoT especially at the sensing layer lack adequate storage resources.

- In regard to the security, while HDPoA is resilient against the DoS attacks and the 51% attacks, due to the fact it can be implemented within IoT system, it inherits the vulnerability in regards to the sybil attacks. However this attack against HDPoA can be time consuming and attackers needs to spend long time building the honesty level of its nodes to be able to have them all promoted to the AN category where different attacks can be launched against the network. This sybil attacks vulnerability is also a limitation of both the architecture to support DAI implementation and the architecture to support AI-enabled IoT applications at the edge

- The final limitation is in regard to the overall energy foot-print resulted form implementing the concept of hash solving similar to PoW. While as shown by the results and the analysis the impact of HDPoA on the IoT devices' power sources can be limited, yet the overall energy cost can be an issue for the environment , unless these sources are batteries or from renewable energy.

8.2 Future Work

In term of the HDPoA consensus mechanism a deployment over a large area with more nodes for a longer field trial is needed. Further testing over LoRa, is required within different IoT contexts to fully evaluate the performance of HDPoA. Within the IoT devices, the storing of the chain locally could potentially create a problem in term of the devices' storage capabilities, this need to be tested and fully analysed and a mechanisms to deal with the storage of the full chain need to be in place. For example, some applications at certain points in time do not need to access old data. This could allow for the data overwriting that would allow for flexible storage of the chain.

Currently the scalable work takes into account the mining work that is carried out by nodes when dealing with the node's honesty score and the network threshold. For future work when a node perform any useful work such as AI related processing tasks or data pre-processing or mining should be incorporated towards increasing the node's honesty score.

Within IoT devices at the sensing layer, nodes can sense and collect a useful amount of data. These nodes need to be allowed to trade this data either by rewarding them

with an honesty score or for money (may be for cryptocurrency). This can be another future work for improving the HDPoA-based blockchain platform.

Another future work related to the architecture for AI-enabled IoT applications at the edge is the integration of biosensors into the system. Further study of their impact on the overall system performance and the security of the collected data is needed. Future work should include full deployment of the system around different sewage water sources to collect and analyse real-world data.

To evaluate the blockchain performance when supporting the deployment of a DAI over IoT hardware, only a simple DMLP engine was deployed. For future work a deployment of another type of a more complex AI, such as a Convolutional Neural Network (CNN) is needed to further evaluate the architecture.

Adding the hardware secure model and the hardware wallet into HDPoA enhances its security. However as the research is at an early stage, the future work should include a longer field trial to further assess the impact of the HSM and HW and conduct more security analyses. Another future work is the full deployment and testing of the energy trading example application, which was previously discussed, based on the securely enhanced HDPoA including further performance evaluations.

Bibliography

[1] S. Li, L. D. Xu, and S. Zhao, "5G internet of things: A survey," *Journal of Industrial Information Integration*, vol. 10, pp. 1–9, 2018.

[2] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, Eds., *Vision and Challenges for Realising the Internet of Things*. Publications Office of the European Union, 2010.

[3] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.

[4] O. Vermesan and P. Friess, Eds., *Internet of Things: From Research and Innovation to Market Deployment*, ser. River Publishers Series in Communication. River, 2014.

[5] P. P. Ray, "A survey on internet of things architectures," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, 2018.

[6] D. Fredrik, P. Mark, R. Alexander, and S. Jonathan, "Growing opportunities in the internet of things," 2019, accessed: Aug 24, 2022. [Online]. Available: https://www.mckinsey.com/industries/private-equity-and-principal-investors/our-insights/growing-opportunities-in-the-internet-of-things

[7] J. Wurm, K. Hoang, O. Arias, A.-R. Sadeghi, and Y. Jin, "Security analysis on consumer and industrial IoT devices," in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016, pp. 519–524.

[8] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[9] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013, including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services & Cloud Computing and Scientific Applications — Big Data, Scalable Analytics, and Beyond. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X13000241

[10] R. R. Yager and J. P. Espada, Eds., *New Advances in the Internet of Things*. Luxembourg: Springer, Cham, 2018.

[11] N. Kshetri, "Can blockchain strengthen the internet of things?" *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017.

[12] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," *Journal of Cryptology*, vol. 3, pp. 99–111, 1991.

[13] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009, accessed: Jun 12, 2022. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[14] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE International Congress on Big Data (BigData Congress)*, 2017, pp. 557–564.

[15] A. F. Rocha Neto, F. C. Delicato, T. V. Batista, and P. F. Pires, "Distributed machine learning for IoT applications in the fog," *Fog Computing: Theory and Practice*, pp. 309–345, 2020.

[16] W. Chong, N. Desai, A. Govindjee, X. Wang, and S. Witherspoon, "What is distributed AI?" 2021, accessed: Jun 26, 2022. [Online]. Available: https://developer.ibm.com/learningpaths/get-started-distributed-ai-apis/what-is-distributed-ai/

[17] S. Ponomarev and A. E. Voronkov, "Multi-agent systems and decentralized artificial superintelligence," 2017. [Online]. Available: https://arxiv.org/abs/1702.08529

[18] G. Serpen, J. Li, L. Liu, and Z. Gao, "WSN-ANN: Parallel and distributed neurocomputing with wireless sensor networks," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 2013, pp. 1–8.

[19] M. van Steen and A. S. Tanenbaum, "A brief introduction to distributed systems," *Computing*, vol. 98, no. 10, pp. 967–1009, 2016. [Online]. Available: https://doi.org/10.1007/s00607-016-0508-7

[20] N. A. Lynch, *Distributed Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996.

[21] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1508–1532, 2019.

[22] S. K. Lo, Y. Liu, S. Y. Chia, X. Xu, Q. Lu, L. Zhu, and H. Ning, "Analysis of blockchain solutions for iot: A systematic literature review," *IEEE Access*, vol. 7, pp. 58 822–58 835, 2019.

[23] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1508–1532, 2019.

[24] A. Sunyaev, *Distributed Ledger Technology*. Cham: Springer International Publishing, 2020, pp. 265–299. [Online]. Available: https://doi.org/10.1007/978-3-030-34957-8_9

[25] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, p. 382–401, jul 1982. [Online]. Available: https://doi.org/10.1145/357172.357176

[26] L. Lamport and M. Massa, "Cheap PAXOS," in *International Conference on Dependable Systems and Networks, 2004*, 2004, pp. 307–314.

[27] G. Hileman and M. Rauchs, "Global blockchain benchmarking study," 2017, accessed: Mar 7, 2022. [Online]. Available: https://ssrn.com/abstract=3040224orhttp://dx.doi.org/10.2139/ssrn.3040224

[28] E. Benos, R. Garratt, and P. Gurrola-Perez, "Bank of England staff working paper no. 670: The economics of distributed ledger technology for securities settlement," 2017, accessed: Mar 15, 2022. [Online]. Available: https://www.bankofengland.co.uk/-/media/boe/files/working-paper/2017/the-economics-of-distributed-ledger-technology-for-securities-settlement.pdf?la=en&hash=17895E1C1FEC86D37E12E4BE63BA9D9741577FE5

[29] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology Proceedings of Crypto 82*, D. Chaum, R. Rivest, and A. Sherman, Eds., 1983, pp. 199–203.

[30] D. Jackson and B. Downey, "E-gold statistics." accessed: Mar 7, 2022. [Online]. Available: https://web.archive.org/web/20041014062818if_/http://www.e-gold.com:80/unsecure/aboutus.html

[31] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in *Advances in Cryptology — CRYPTO' 92*, E. F. Brickell, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 139–147.

[32] M. Jakobsson and A. Juels, *Proofs of Work and Bread Pudding Protocols(Extended Abstract)*. Boston, MA: Springer US, 1999, pp. 258–272. [Online]. Available: https://doi.org/10.1007/978-0-387-35568-9_18

[33] A. Back, "Hash cash postage implementation," 1997, accessed: Mar 7, 2022. [Online]. Available: http://www.hashcash.org/papers/announce.txt

[34] W. Dai, "B-money," 1998, accessed: Mar 7, 2022. [Online]. Available: https://en.bitcoin.it/wiki/B-money

[35] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, Sep. 1997. [Online]. Available: https://firstmonday.org/ojs/index.php/fm/article/view/548

[36] V. Buterin, "Ethereum white paper a next generation smart contract & decentralized application platform," 2014, accessed: Jul 4, 2022. [Online]. Available: https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf

[37] H. Pervez, M. Muneeb, M. U. Irfan, and I. U. Haq, "A comparative analysis of DAG-based blockchain architectures," in *2018 12th International Conference on Open Source Systems and Technologies (ICOSST)*, 2018, pp. 27–34.

[38] S. Popov, "The tangle," *White paper*, 2018, accessed: Nov 20, 2021. [Online]. Available: https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf

[39] A. Churyumov, "Byteball: A decentralized system for storage and transfer of value," accessed: Mar 7, 2022. [Online]. Available: https://obyte.org/Byteball.pdf

[40] Y. Ribero, "Dagcoin whitepaper," accessed: Mar 7, 2022. [Online]. Available: https://prismic-io.s3.amazonaws.com/dagcoin/f4e531e1-a5db-43b6-930c-14bf705e65ee_Dagcoin_White_Paper.pdf

[41] D. Cheatoshin, "The dagger crypto currency: white paper v0.3," accessed: Mar 7, 2022. [Online]. Available: https://github.com/XDagger/xdag/blob/master/WhitePaper.md

[42] L. Baird, M. Harmon, and P. Madsen, "Hedera: A public hashgraph network & governing council, whitepaper v.2.1," accessed: Mar 7, 2022. [Online]. Available: https://hedera.com/hh_whitepaper_v2.1-20200815.pdf

[43] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173–190, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17329205

[44] N. M. Kumar and P. K. Mallick, "Blockchain technology for security issues and challenges in IoT," *Procedia Computer Science*, vol. 132, pp. 1815–1823, 2018, international Conference on Computational Intelligence and Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187705091830872X

[45] M. Maroufi, R. Abdolee, and B. M. Tazekand, "On the convergence of blockchain and internet of things (IoT) technologies," *Journal of Strategic*

Innovation and Sustainability, vol. 14, no. 1, Mar. 2019. [Online]. Available: https://articlegateway.com/index.php/JSIS/article/view/990

[46] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," Tech. Rep., oct 2018. [Online]. Available: https://doi.org/10.6028%2Fnist.ir.8202

[47] B. Furht, Ed., *The RSA Public-Key Encryption Algorithm*. Boston, MA: Springer US, 2006, pp. 757–757. [Online]. Available: https://doi.org/10.1007/0-387-30038-4_206

[48] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, p. 36–630, 2001.

[49] M. Qu, "Sec 2: Recommended elliptic curve domain parameters," *Certicom Res., Mississauga, ON, Canada, Tech. Rep. SEC2-Ver-0.6*, 1999.

[50] T. Lange, "Koblitz curve cryptosystems," *Finite Fields and Their Applications*, vol. 11, no. 2, pp. 200–229, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1071579704000395

[51] H. Guo and X. Yu, "A survey on blockchain technology and its security," *Blockchain: Research and Applications*, p. 100067, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2096720922000070

[52] B. Project, "Bitcoin developer reference." [Online]. Available: https://developer.bitcoin.org/reference/

[53] B. wiki, "Difficulty," 2021, accessed: Jul 4, 2022. [Online]. Available: https://en.bitcoin.it/wiki/Difficulty

[54] btcinformation.org, "Bitcoin developer guide." [Online]. Available: https://btcinformation.org/en/developer-guide#block-height-and-forking

[55] E. Community, "Ethereum homestead documentation." [Online]. Available: https://www.ethdocs.org/en/latest/index.html

[56] H. W. P. W. Group, "An introduction to hyperledger," 2018, accessed: May 18, 2021. [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2018/07/HL_Whitepaper_IntroductiontoHyperledger.pdf

[57] J. P. M. Chase., "Goguorum," 2018, accessed: Jun 26, 2022. [Online]. Available: https://github.com/ConsenSys/quorum

[58] K. J. O'Dwyer and D. Malone, "Bitcoin mining and its energy footprint," in *25th IET Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014)*, 2014, pp. 280–285.

[59] K. Sunny and N. Scott, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," 2012. [Online]. Available: https://decred.org/research/king2012.pdf

[60] N. Houy, "It will cost you nothing to kill a proof-of-stake crypto-currency," *Economics Bulletin*, vol. 34, no. 2, pp. 1038–1044, 2014. [Online]. Available: https://ideas.repec.org/a/ebl/ecbull/eb-14-00114.html

[61] B. Chase and E. MacBrough, "Analysis of the XRP ledger consensus protocol," *CoRR*, vol. abs/1802.07242, 2018. [Online]. Available: http://arxiv.org/abs/1802.07242

[62] F. Yang, W. Zhou, Q. Wu, R. Long, N. N. Xiong, and M. Zhou, "Delegated proof of stake with downgrade: A secure and efficient blockchain consensus algorithm with downgrade mechanism," *IEEE Access*, vol. 7, pp. 118 541–118 555, 2019.

[63] B. Lashkari and P. Musilek, "A comprehensive review of blockchain consensus mechanisms," *IEEE Access*, vol. 9, pp. 43 620–43 652, 2021.

[64] K. Karantias, A. Kiayias, and D. Zindros, "Proof-of-burn," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science, J. Bonneau and N. Heninger, Eds. Springer-Verlag, Jul. 2020, pp. 523–540, 24th International Conference on Financial Cryptography and Data Security 2020, FC 2020 ; Conference date: 10-02-2020 Through 14-02-2020. [Online]. Available: https://fc20.ifca.ai/

[65] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *CoRR*, vol. abs/1710.09437, 2017. [Online]. Available: http://arxiv.org/abs/1710.09437

[66] L. Lamport, "PAXOS made simple," *ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001)*, pp. 51–58, December 2001. [Online]. Available: https://www.microsoft.com/en-us/research/publication/paxos-made-simple/

[67] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *2014 USENIX Annual Technical Conference (USENIX ATC 14)*. Philadelphia, PA: USENIX Association, Jun. 2014, pp. 305–319. [Online]. Available: https://www.usenix.org/conference/atc14/technical-sessions/presentation/ongaro

[68] J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu, and P. Zeng, "Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3680–3689, 2019.

[69] L. Bahri and S. Girdzijauskas, "When trust saves energy: A reference framework for proof of trust (PoT) blockchains," in *Companion Proceedings of the The Web Conference 2018*, ser. WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, p. 1165–1169. [Online]. Available: https://doi.org/10.1145/3184558.3191553

[70] G. Sagirlar, B. Carminati, E. Ferrari, J. D. Sheehan, and E. Ragnoli, "Hybrid-IoT: Hybrid blockchain architecture for internet of things - PoW sub-blockchains," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2018, pp. 1007–1016.

[71] R. Xu, Y. Chen, E. Blasch, and G. Chen, "Microchain: A hybrid consensus mechanism for lightweight distributed ledger for IoT," 2019.

[72] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang, "PoBT: A lightweight consensus algorithm for scalable iot business blockchain," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2343–2355, 2020.

[73] L. Lao, X. Dai, B. Xiao, and S. Guo, "G-PBFT: A location-based and scalable consensus protocol for IoT-blockchain applications," in *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2020, pp. 664–673.

[74] Raghav, N. Andola, S. Venkatesan, and S. Verma, "PoEWAL: A lightweight consensus mechanism for blockchain in IoT," *Pervasive Mob. Comput.*, vol. 69, p. 101291, 2020.

[75] E. K. Wang, R. Sun, C.-M. Chen, Z. Liang, S. Kumari, and M. Khurram Khan, "Proof of X-repute blockchain consensus protocol for iot systems," *Computers & Security*, vol. 95, p. 101871, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404820301449

[76] D. Puthal, S. P. Mohanty, P. Nanda, E. Kougianos, and G. Das, "Proof-of-authentication for scalable blockchain in resource-constrained distributed systems," in *2019 IEEE International Conference on Consumer Electronics (ICCE)*, 2019, pp. 1–5.

[77] S. Nick, "Smart contracts: Building blocks for digital markets," 1996, accessed: Jun 26, 2022. [Online]. Available: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html

[78] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[79] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F.-Y. Wang, "Blockchain-enabled smart contracts: Architecture, applications, and future trends," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 11, pp. 2266–2277, 2019.

[80] D. Ali, S. K. Salil, and J. Raja, "Blockchain in internet of things: Challenges and solutions," *CoRR*, vol. abs/1608.05187, 2016. [Online]. Available: http://arxiv.org/abs/1608.05187

[81] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "LSB: A lightweight scalable blockchain for IoT security and anonymity," *Journal of Parallel and Distributed Computing*, vol. 134, p. 180–197, Dec 2019. [Online]. Available: http://dx.doi.org/10.1016/j.jpdc.2019.08.005

[82] ——, "Blockchain for IoT security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017, pp. 618–623.

[83] O. Alphand, M. Amoretti, T. Claeys, S. Dall'Asta, A. Duda, G. Ferrari, F. Rousseau, B. Tourancheau, L. Veltri, and F. Zanichelli, "IoTChain: A blockchain security architecture for the internet of things," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6.

[84] Y. Zhang and J. Wen, "The IoT electric business model: Using blockchain technology for the internet of things," *Peer-to-Peer Networking and Applications*, vol. 10, no. 4, pp. 983–994, 2017. [Online]. Available: https://doi.org/10.1007/s12083-016-0456-1

[85] H. Shrobe, D. L. Shrier, and A. Pentland, "Chapter 15 enigma: Decentralized computation platform with guaranteed privacy," in *New Solutions for Cybersecurity*, 2018, pp. 425–454.

[86] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.

[87] A. Ouaddah, A. Abou Elkalam, and A. Ait Ouahman, "Fairaccess: a new blockchain-based access control framework for the internet of things," *Security and Communication Networks*, vol. 9, no. 18, pp. 5943–5964, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1748

[88] T. Le and M. W. Mutka, "Capchain: A privacy preserving access control framework based on blockchain for pervasive environments," in *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2018, pp. 57–64.

[89] H. Shafagh, L. Burkhalter, A. Hithnawi, and S. Duquennoy, "Towards blockchain-based auditable storage and sharing of IoT data," 2017. [Online]. Available: https://arxiv.org/abs/1705.08230

[90] Z. Gong-Guo and Z. Wan, "Blockchain-based IoT security authentication system," in *2021 International Conference on Computer, Blockchain and Financial Development (CBFD)*, 2021, pp. 415–418.

[91] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu, "Blockchain based data integrity service framework for IoT data," in *2017 IEEE International Conference on Web Services (ICWS)*, 2017, pp. 468–475.

[92] Y. Lin, J. Li, S. Kimura, Y. Yang, Y. Ji, and Y. Cao, "Consortium blockchain-based public integrity verification in cloud storage for IoT," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3978–3987, 2022.

[93] W. M. Dahmane, S. Ouchani, and H. Bouarfa, "Guaranteeing information integrity through blockchains for smart cities," in *Model and Data Engineering*, C. Attiogbé and S. Ben Yahia, Eds. Cham: Springer International Publishing, 2021, pp. 199–212.

[94] M. Altulyan, L. Yao, S. S. Kanhere, X. Wang, and C. Huang, "A unified framework for data integrity protection in people-centric smart cities," *Multimedia Tools and Applications*, vol. 79, no. 7, pp. 4989–5002, 2020. [Online]. Available: https://doi.org/10.1007/s11042-019-7182-7

[95] J. Cui, F. Ouyang, Z. Ying, L. Wei, and H. Zhong, "Secure and efficient data sharing among vehicles based on consortium blockchain," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2021.

[96] W. Zhang, Y. Bai, and J. Feng, "TIIA: A blockchain-enabled threat intelligence integrity audit scheme for IIoT," *Future Generation Computer Systems*, vol. 132, pp. 254–265, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X22000723

[97] K. Biswas and V. Muthukkumarasamy, "Securing smart cities using blockchain technology," in *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2016, pp. 1392–1393.

[98] R. Xu, L. Hang, W. Jin, and D. Kim, "Distributed secure edge computing architecture based on blockchain for real-time data integrity in IoT environments," *Actuators*, vol. 10, no. 8, 2021. [Online]. Available: https://www.mdpi.com/2076-0825/10/8/197

[99] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 840–852, 2018.

[100] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: decentralized computation platform with guaranteed privacy," 2015, accessed: Aug 11, 2022. [Online]. Available: https://arxiv.org/abs/1506.03471

[101] S. Cha, J.-F. Chen, C. Su, and K.-H. Yeh, "A blockchain connected gateway for BLE-Based devices in the internet of things," *IEEE Access*, vol. 6, pp. 24 639–24 649, 2018.

[102] Y. Zhang and J. Wen, "The IoT electric business model: Using blockchain technology for the internet of things," *Peer-to-Peer Networking and Applications*, vol. 10, no. 4, pp. 983–994, 2017. [Online]. Available: https://doi.org/10.1007/s12083-016-0456-1

[103] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, "Towards a novel privacy-preserving access control model based on blockchain technology in IoT," in *Europe and MENA Cooperation Advances in Information and Communication Technologies*, Á. Rocha, M. Serrhini, and C. Felgueiras, Eds. Cham: Springer International Publishing, 2017, pp. 523–533.

[104] Y. Rahulamathavan, R. C.-W. Phan, M. Rajarajan, S. Misra, and A. Kondoz, "Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption," in *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2017, pp. 1–6.

[105] N. Foukia, D. Billard, and E. Solana, "PISCES: A framework for privacy by design in IoT," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, 2016, pp. 706–713.

[106] G. Zyskind, O. Nathan, and A. S. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*, 2015, pp. 180–184.

[107] M. Conoscenti, A. Vetrò, and J. C. De Martin, "Peer to peer for privacy and decentralization in the internet of things," in *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, 2017, pp. 288–290.

[108] A. D. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, "A decentralized privacy-preserving healthcare blockchain for IoT," *Sensors*, vol. 19, no. 2, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/2/326

[109] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1676–1717, 2019.

[110] F. Lombardi, L. Aniello, S. De Angelis, A. Margheri, and V. Sassone, "A blockchain-based infrastructure for reliable and cost-effective IoT-aided smart

grids," in *Living in the Internet of Things: Cybersecurity of the IoT*, 2018, pp. 1–6.

[111] Filament, "Enterprise blockchain solutions, IoT filament," 2016. [Online]. Available: https://filament.com/index.html.

[112] B. Lee and J.-H. Lee, "Blockchain-based secure firmware update for embedded devices in an internet of things environment," *J. Supercomput.*, vol. 73, no. 3, p. 1152–1167, mar 2017. [Online]. Available: https://doi.org/10.1007/s11227-016-1870-0

[113] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending bitcoin's proof of work via proof of stake," *SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 3, p. 34–37, dec 2014. [Online]. Available: https://doi.org/10.1145/2695533.2695545

[114] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.

[115] S. Ibba, A. Pinna, M. Seu, and F. E. Pani, "Citysense: Blockchain-oriented smart cities," in *Proceedings of the XP2017 Scientific Workshops*, ser. XP '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3120459.3120472

[116] W. Viriyasitavat, L. D. Xu, Z. Bi, D. Hoonsopon, and N. Charoenruk, "Managing QoS of internet-of-things services using blockchain," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1357–1368, 2019.

[117] D. E. Kouicem, Y. Imine, A. Bouabdallah, and H. Lakhlef, "Decentralized blockchain-based trust management protocol for the internet of things," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 1292–1306, 2022.

[118] A. Bahga and V. Madisetti, "Blockchain platform for industrial internet of things," *Journal of Software Engineering and Applications*, vol. 09, pp. 533–546, 01 2016.

[119] L. Bai, M. Hu, M. Liu, and J. Wang, "BPIIoT: A light-weighted blockchain-based platform for industrial IoT," *IEEE Access*, vol. 7, pp. 58 381–58 393, 2019.

[120] P. Veena, P. Sanjay, N. Sumabala, and B. Paul, "Empowering the edge: Practical insights on a decentralized internet of things." [Online]. Available: https://www.ibm.com/downloads/cas/2NZLY7XJ

[121] P. K. Sharma, M.-Y. Chen, and J. H. Park, "A software defined fog node based distributed blockchain cloud architecture for IoT," *IEEE Access*, vol. 6, pp. 115–124, 2018.

[122] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, "Blockchain-based dynamic key management for heterogeneous intelligent transportation systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1832–1843, 2017.

[123] A. Stanciu, "Blockchain based distributed control system for edge computing," in *2017 21st International Conference on Control Systems and Computer Science (CSCS)*, 2017, pp. 667–671.

[124] J. Pan, J. Wang, A. Hester, I. Alqerm, Y. Liu, and Y. Zhao, "EdgeChain: An edge-IoT framework and prototype based on blockchain and smart contracts," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4719–4732, 2019.

[125] A. Jindal, G. S. Aujla, and N. Kumar, "SURVIVOR: A blockchain based edge-as-a-service framework for secure energy trading in SDN-enabled vehicle-to-grid environment," *Computer Networks*, vol. 153, pp. 36–48, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S138912861831106X

[126] R. Chaudhary, A. Jindal, G. S. Aujla, S. Aggarwal, N. Kumar, and K.-K. R. Choo, "BEST: blockchain-based secure energy trading in SDN-enabled intelligent transportation system," *Computers & Security*, vol. 85, pp. 288–299, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016740481831201X

[127] P. K. Sharma and J. H. Park, "Blockchain based hybrid network architecture for the smart city," *Future Generation Computer Systems*, vol. 86, pp. 650–655, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X1830431X

[128] A. Fitwi, Y. Chen, and S. Zhu, "A lightweight blockchain-based privacy protection for smart surveillance at the edge," in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 552–555.

[129] Z. Cai, G. Yang, S. Xu, C. Zang, J. Chen, P. Hang, and B. Yang, "RBaaS: A robust blockchain as a service paradigm in cloud-edge collaborative environment," *IEEE Access*, vol. 10, pp. 35 437–35 444, 2022.

[130] X. Zhang, R. Li, and B. Cui, "A security architecture of vanet based on blockchain and mobile edge computing," in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, 2018, pp. 258–259.

[131] Y. Lu, J. Zhang, Y. Qi, S. Qi, Y. Zheng, Y. Liu, H. Song, and W. Wei, "Accelerating at the edge: A storage-elastic blockchain for latency-sensitive vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2021.

[132] W. De Brouwer and M. Borda, "Neuron: decentralized artificial intelligence, distributing deep learning to the edge of the network," 2017, accessed: Jul 17, 2022. [Online]. Available: https://neironix.io/documents/whitepaper/0c59985c14a53024558894ef06e08de2.pdf

[133] M. A. Rahman, M. M. Rashid, M. S. Hossain, E. Hassanain, M. F. Alhamid, and M. Guizani, "Blockchain and IoT-based cognitive edge framework for sharing economy services in a smart city," *IEEE Access*, vol. 7, pp. 18 611–18 621, 2019.

[134] A. Nawaz, T. N. Gia, J. P. Queralta, and T. Westerlund, "Edge AI and blockchain for privacy-critical and data-sensitive applications," in *2019 Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, 2019, pp. 1–2.

[135] R. Gupta, D. Reebadiya, S. Tanwar, N. Kumar, and M. Guizani, "When blockchain meets edge intelligence: Trusted and security solutions for consumers," *IEEE Network*, vol. 35, no. 5, pp. 272–278, 2021.

[136] J. Xu, S. Wang, A. Zhou, and F. Yang, "Edgence: a blockchain-enabled edge-computing platform for intelligent IoT-based dApps," *China Communications*, vol. 17, no. 4, pp. 78–87, 2020.

[137] C. Qiu, H. Yao, X. Wang, N. Zhang, F. R. Yu, and D. Niyato, "AI-Chain: blockchain energized edge intelligence for beyond 5G networks," *IEEE Network*, vol. 34, no. 6, pp. 62–69, 2020.

[138] X. Lin, J. Li, J. Wu, H. Liang, and W. Yang, "Making knowledge tradable in Edge-AI enabled IoT: A consortium blockchain-based efficient and incentive approach," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6367–6378, 2019.

[139] Q. Wang, Y. Guo, X. Wang, T. Ji, L. Yu, and P. Li, "AI at the edge: Blockchain-empowered secure multiparty learning with heterogeneous models," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9600–9610, 2020.

[140] S. Rathore, Y. Pan, and J. H. Park, "BlockDeepNet: A blockchain-based secure deep learning for IoT network," *Sustainability*, vol. 11, no. 14, 2019. [Online]. Available: https://www.mdpi.com/2071-1050/11/14/3974

[141] T. Kuo and L. Ohno-Machado, "ModelChain: Decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks," *CoRR*, vol. abs/1802.01746, 2018. [Online]. Available: http://arxiv.org/abs/1802.01746

[142] M. A. Ferrag and L. Maglaras, "Deepcoin: A novel deep learning and blockchain-based energy exchange framework for smart grids," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1285–1297, 2020.

[143] S. K. Singh, Y.-S. Jeong, and J. H. Park, "A deep learning-based IoT-oriented infrastructure for secure smart city," *Sustainable Cities and Society*, vol. 60, p. 102252, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221067072030473X

[144] D. Unal, M. Hammoudeh, M. A. Khan, A. Abuarqoub, G. Epiphaniou, and R. Hamila, "Integration of federated machine learning and blockchain for the provision of secure big data analytics for internet of things," *Computers & Security*, vol. 109, p. 102393, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404821002170

[145] P. Ramanan and K. Nakayama, "Baffle : Blockchain based aggregator free federated learning," in *2020 IEEE International Conference on Blockchain (Blockchain)*, 2020, pp. 72–81.

[146] P. Bagga, A. K. Sutrala, A. K. Das, and P. Vijayakumar, "Blockchain-based batch authentication protocol for internet of vehicles," *Journal of Systems Architecture*, vol. 113, p. 101877, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762120301569

[147] A. Islam, T. Rahim, M. Masuduzzaman, and S. Y. Shin, "A blockchain-based artificial intelligence-empowered contagious pandemic situation supervision scheme using internet of drone things," *IEEE Wireless Communications*, vol. 28, no. 4, pp. 166–173, 2021.

[148] A. Islam and S. Young Shin, "A blockchain-based secure healthcare scheme with the assistance of unmanned aerial vehicle in internet of things," *Computers & Electrical Engineering*, vol. 84, p. 106627, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790620304821

[149] J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu, and P. Zeng, "Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3680–3689, 2019.

[150] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the internet of things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.

[151] S. Huh, S. Cho, and S. Kim, "Managing IoT devices using blockchain platform," in *2017 19th International Conference on Advanced Communication Technology (ICACT)*, 2017, pp. 464–467.

[152] A. Bahga and V. Madisetti, "Blockchain platform for industrial internet of things," *Journal of Software Engineering and Applications*, vol. 09, pp. 533–546, 01 2016.

[153] P. Szilágyi, "Eip-225: Clique proof-of-authority consensus protocol," 2017, accessed: Feb 13, 2021. [Online]. Available: https://eips.ethereum.org/EIPS/eip-225

[154] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[155] E. Wiki, "Light client protocol," accessed: Nov 20, 2021. [Online]. Available: https://eth.wiki/en/concepts/light-client-protocol

[156] R. Ross, "SP 800-30 Rev 1, guide for conducting risk assessments." Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, 2012. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-30r1

[157] B. Ford, P. Srisuresh, and D. Kegel, "Peer-to-peer communication across network address translators," accessed: Nov 19, 2021. [Online]. Available: https://bford.info/pub/net/p2pnat/

[158] N. Kshetri, "Can blockchain strengthen the internet of things?" *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017.

[159] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE International Congress on Big Data (BigData Congress)*, 2017, pp. 557–564.

[160] J. F. Buford, H. Yu, and E. K. Lua, "Chapter 14 - Security," in *P2P Networking and Applications*, J. F. Buford, H. Yu, and E. K. Lua, Eds. Boston: Morgan Kaufmann, 2009, pp. 319–340. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123742148000143

[161] G. Sagirlar, B. Carminati, E. Ferrari, J. D. Sheehan, and E. Ragnoli, "Hybrid-IoT: Hybrid blockchain architecture for internet of things - PoW sub-blockchains," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2018, pp. 1007–1016.

[162] B. Anshika, "Top cryptocurrencies with their high transaction speeds," 2022. [Online]. Available: https://www.blockchain-council.org/cryptocurrency/top-cryptocurrencies-with-their-high-transaction-speeds/

[163] A. Elsts, E. Mitskas, and G. Oikonomou, "Distributed ledger technology and the internet of things: A feasibility study," in *Proceedings of the 1st Workshop on Blockchain-Enabled Networked Sensor Systems*, ser. BlockSys'18. New York, NY, USA: Association for Computing Machinery, 2018, p. 7–12. [Online]. Available: https://doi.org/10.1145/3282278.3282280

[164] K. L. Cai and F. J. Lin, "Distributed artificial intelligence enabled by oneM2M and fog networking," in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2018, pp. 1–6.

[165] G. Serpen and J. Li, "Parallel and distributed computations of maximum independent set by a Hopfield neural net embedded into a wireless sensor network," *Procedia Computer Science*, vol. 6, pp. 390–395, 2011, complex adaptive sysytems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050911005382

[166] J. Li and G. Serpen, "nesC-TinyOS model for parallel and distributed computation of max independent set by Hopfield network on wireless sensor network," *Procedia Computer Science*, vol. 6, pp. 396–401, 2011, complex adaptive sysytems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050911005394

[167] ——, "TOSSIM simulation of wireless sensor network serving as hardware platform for Hopfield neural net configured for max independent set," *Procedia Computer Science*, vol. 6, pp. 408–412, 2011, complex adaptive sysytems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050911005412

[168] S. Teerapittayanon, B. McDanel, and H. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 328–339.

[169] T. Bi, Q. Liu, T. Ozcelebi, D. Jarnikov, and D. Sekulovski, "PCANN: Distributed ANN architecture for image recognition in resource-constrained IoT devices," in *2019 15th International Conference on Intelligent Environments (IE)*, 2019, pp. 1–8.

[170] S. Rathore, Y. Pan, and J. H. Park, "BlockDeepNet: A blockchain-based secure deep learning for IoT network," *Sustainability*, vol. 11, no. 14, 2019. [Online]. Available: https://www.mdpi.com/2071-1050/11/14/3974

[171] C. Arouri, E. M. Nguifo, S. Aridhi, C. Roucelle, G. Bonnet-Loosli, and N. Tsopzé, "Towards a constructive multilayer perceptron for regression task using non-parametric clustering. a case study of photo-z redshift reconstruction," 2014, accessed: Jul 17, 2022. [Online]. Available: https://arxiv.org/abs/1412.5513

[172] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[173] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *IEEE Access*, vol. 8, pp. 16 440–16 455, 2020.

[174] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016, accessed: Jul 17, 2022. [Online]. Available: https://www.bitcoinlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf

[175] P. Joseph and B. Vitalik, "Plasma: Scalable autonomous smart contracts," 2017, accessed: Jul 17, 2022. [Online]. Available: https://plasma.io/plasma-deprecated.pdf

[176] M. J. Dworkin, E. B. Barker, J. R. Nechvatal, J. Foti, L. E. Bassham, E. Roback, J. F. Dray Jr *et al.*, "Advanced encryption standard (aes)," 2001, accessed: Jul 17, 2022. [Online]. Available: https://www.nist.gov/publications/advanced-encryption-standard-aes?gclid=cj0kcqjwudb3brc9arisaea-vuvw_18-e5i49b218fc7tfn5_fr-hdaj9s-mqglxel3fsormn_ydg-aaar5gealw_wcb

[177] R. Fisher, "Iris dataset, UCI machine learning repository," 1988, accessed: Jan. 15, 2021. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/iris

[178] V. Saverio, De, "Air quality dataset, UCI machine learning repository," 2016, accessed:Jul. 5, 2020. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Air+quality

[179] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18327687

[180] S. Rathore and J. Park, "DeepBlockIoTNet: A secure deep learning approach with blockchain for the IoT network," *Trans. Ind. Inform*, vol. 11, no. 14, p. 3974, 2019.

[181] R. S. Ross *et al.*, "Guide for conducting risk assessments," 2012, accessed:Jul. 15, 2022. [Online]. Available: https://www.nist.gov/publications/guide-conducting-risk-assessments?pub_id=912091

[182] K. Mao, H. Zhang, and Z. Yang, "Can a paper-based device trace covid-19 sources with wastewater-based epidemiology?" *Environmental Science & Technology*, vol. 54, no. 7, pp. 3733–3735, 04 2020. [Online]. Available: https://doi.org/10.1021/acs.est.0c01174

[183] Z. Yang, B. Kasprzyk-Hordern, C. G. Frost, P. Estrela, and K. V. Thomas, "Community sewage sensors for monitoring public health," *Environmental Science & Technology*, vol. 49, no. 10, pp. 5845–5846, 05 2015. [Online]. Available: https://doi.org/10.1021/acs.est.5b01434

[184] keras.io, "Tensorflow keras: Developer guides," accessed:Jul. 17, 2021. [Online]. Available: https://keras.io/guides/

[185] D. Barceló, "Wastewater-based epidemiology to monitor COVID-19 outbreak: Present and future diagnostic methods to be in your radar," *Case Studies in Chemical and Environmental Engineering*, vol. 2, p. 100042, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666016420300402

[186] K. Mao, H. Zhang, Y. Pan, and Z. Yang, "Biosensors for wastewater-based epidemiology for monitoring public health," *Water Research*, vol. 191, p. 116787, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0043135420313208

[187] O. M. Abdeldayem, A. M. Dabbish, M. M. Habashy, M. K. Mostafa, M. Elhefnawy, L. Amin, E. G. Al-Sakkari, A. Ragab, and E. R. Rene, "Viral outbreaks detection and surveillance using wastewater-based epidemiology, viral air sampling, and machine learning techniques: A comprehensive review and outlook," *Science of The Total Environment*, vol. 803, p. 149834, 2022.

[188] J. Dofe, J. Frey, and Q. Yu, "Hardware security assurance in emerging IoT applications," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016, pp. 2050–2053.

[189] K. Nomikos, A. Papadimitriou, G. Stergiopoulos, D. Koutras, M. Psarakis, and P. Kotzanikolaou, "On a security-oriented design framework for medical IoT devices: The hardware security perspective," in *2020 23rd Euromicro Conference on Digital System Design (DSD)*, 2020, pp. 301–308.

[190] J. He, Y. Zhao, X. Guo, and Y. Jin, "Hardware trojan detection through chip-free electromagnetic side-channel statistical analysis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2939–2948, 2017.

[191] R. Agrawal and R. Vemuri, "On state encoding against power analysis attacks for finite state controllers," in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2018, pp. 181–186.

[192] A. K. Singh and N. Kushwaha, "Software and hardware security of IoT," in *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 2021, pp. 1–5.

[193] C. Lesjak, D. Hein, and J. Winter, "Hardware-security technologies for industrial IoT: TrustZone and security controller," in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, 2015, pp. 002 589–002 595.

[194] R. Philip, S. Vouthanack, T. Kenneth, U. David, V. Michael, and W. Amy, "Trusted platform module," 2012, accessed:Jul. 18, 2022. [Online]. Available: https://researchedsolution.wordpress.com/2013/09/14/trusted-platform-module/

[195] D. L. Evans, P. Bond, and A. Bement, "Fips pub 140-2: Security requirements for cryptographic modules," *Federal Information Processing Standards Publication: Gaithersburg, MD, USA*, vol. 12, p. 965, 2002.

[196] M. A. Mehrabi, C. Doche, and A. Jolfaei, "Elliptic curve cryptography point multiplication core for hardware security module," *IEEE Transactions on Computers*, vol. 69, no. 11, pp. 1707–1718, 2020.

[197] Zymbit, "Hsm6," accessed:Feb. 21, 2022. [Online]. Available: https://www.zymbit.com/hsm6/

[198] C. Pérez-Solà, S. Delgado-Segura, G. Navarro-Arribas, and J. Herrera-Joancomartí, "Double-spending prevention for bitcoin zero-confirmation transactions," *International Journal of Information Security*, vol. 18, no. 4, pp. 451–463, 2019.

[199] Zymbit, "Hsm4," accessed:Feb. 21, 2022. [Online]. Available: https://www.zymbit.com/hsm4/

[200] F. Fresolone, R. Kloibhofer, A. Ralbovsky, P. Farkas, M. Rakus, and T. Palenik, "Throughput and one-way latency measurements in a 3G/4G live-network hi-mobility uplink," in *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, 2016, pp. 44–49.