Blockchain-based Authentication and Access Control Mechanism for Internet of Things (IoT)

by

MWRWAN ABDELRAZIG ABUBAKAR

Thesis submitted in partial fulfilment of the requirements

of Edinburgh Napier University, for the award of

Doctor of Philosophy

School of Computing, Engineering & The Built Environment

Edinburgh Napier University

SEPTEMBER 2023

Author's declaration

I hereby declare that the work presented in this thesis has not been submitted for any other degree or professional qualification, and that it is the result of my own independent work.

SIGNED: ....Mwrwan Abubakar..... DATE: ....30/06/2023....

Abstract

Authentication and access control are critical in addressing IoT security and privacy issues. However, due to resource overhead, most legacy authentication and authorisation mechanisms are not suitable for resource-constrained IoT devices (Meneghello et al., 2019). Another significant obstacle to IoT is the centralisation of efficient security solutions, such as Public Key Infrastructure (PKI), which presents scalability challenges in a system with thousands of connected nodes. Additionally, current authentication and access control standards rely on third parties. Scalability and deployment simplicity are advantages, but it requires trust in a third party to store customers' sensitive data, which is prone to misuse in the event of security breaches (Levi and Caglayan, 1997). However, blockchain and decentralised alternatives that do not rely on a third party can provide autonomous authentication and authorisation administration. Driven by the potential benefits of blockchain technology and the need to deliver reliable solutions that satisfy the demands of the IoT, this thesis aims to develop blockchain-based decentralised authentication and access control mechanisms for IoT to resolve security and privacy concerns in the current centralised paradigm and remove the need for a third party to maintain trust. The thesis goes further and explores the provision of decentralised identity management services such as secure and fair exchange, delegation, and revocation of credentials through the use of smart contracts. Additionally, the thesis looks into the inherent issue of designing secure, lightweight, and scalable decentralised systems that satisfy IoT needs by proposing a lightweight consensus mechanism. The contributions of this thesis are shown over all layers of the IoT architecture. For instance, the thesis proposed a blockchain-based two-factor authentication mechanism enabling authentication at the IoT applications layer. For authentication in IoT communication protocols, the thesis proposed a lightweight authentication and authorisation mechanism for the MQTT messaging protocol. Additionally, for authentication and access control at the devices layer, the thesis provided a decentralised authentication and access control for wearable medical devices. Finally, the thesis proposed a lightweight and scalable consensus mechanism that overcomes the resource overhead of distributed consensus and the complexity of blockchain in IoT. Further analysis of these approaches' usability, mainly CPU and memory usage, was conducted compared to the current security protocols. When subjected to security analysis and evaluation, the proposed approaches demonstrated performance improvements in data privacy levels, high security and lightweight access control design compared to the current centralised access control models.

Publications associated with this research

- Abubakar, M., Jarocheh, Z., Al-Dubai, A., Liu, X. (2022). A Survey on the Integration of Blockchain and IoT: Challenges and Opportunities. Big Data Privacy and Security in Smart Cities, 197-221.
- Abubakar, Mwrwan, Hisham Ali, Baraq Ghaleb, Isam Wadhaj, and William J. Buchanan. "An Overview of Blockchain-Based IoT Architectures and Designs." In International Conference on Emerging Technologies and Intelligent Systems, pp. 596-605. Cham: Springer International Publishing, 2022.
- Abubakar, M., Jaroucheh, Z., Al Dubai, A., Liu, X. (2022, May). A Lightweight and User-centric Two-factor Authentication Mechanism for IoT Based on Blockchain and Smart Contract. In 2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH) (pp. 91-96). IEEE.
- ABDELRAZIG ABUBAKAR, MWRWAN, Jaroucheh, Z., Al-Dubai, A., Liu, X. (2021, March). Blockchain-based identity and authentication scheme for MQTT protocol. In 2021 The 3rd International Conference on Blockchain Technology (ICBCT '21). Association for Computing Machinery, New York, NY, USA, (pp. 73-81). DOI:https://doi.org/10.1145/3460537.3460549
- Abubakar, M., Jaroucheh, Z., Al Dubai, A., Buchanan, B. (2021, August). A Decentralised Authentication and Access Control Mechanism for Medical Wearable Sensors Data. In 2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS) (pp. 1-7). IEEE.
- Abubakar, M., Jaroucheh, Z., Al-Dubai, A., Buchanan, B. (2020, December). PoNW: A Secure and Scalable Proof-of-Notarized-Work Based Consensus Mechanism. In Proceedings of the 2020 4th International Conference on Vision, Image and Signal Processing (ICVISP 2020). Association for Computing Machinery, New York, NY, USA, Article 58, (pp. 1-8). DOI:https://doi.org/10.1145/3448823.3448875

Other Publications

The following publications are not part of this thesis but are closely related.

- Abubakar, M., Jaroucheh, Z., Al Dubai, A., Buchanan, B. (2021, October). Blockchain-Based Authentication and Registration Mechanism for SIP-Based VoIP Systems. In 2021 5th Cyber Security in Networking Conference (CSNet) (pp. 63-70). IEEE.
- M. Abubakar, P. McCarron, Z. Jaroucheh, A. Al Dubai and B. Buchanan, "Blockchain-based Platform for Secure Sharing and Validation of Vaccination Certificates," 2021 14th International Conference on Security of Information and Networks (SIN), 2021, pp. 1-8, doi: 10.1109/SIN54109.2021.9699221.

Acknowledgements

First and foremost, I give the Almighty God all the glory, honour, and adorations for all the strength, perseverance, and energy he bestowed upon me throughout the beginning to the completion of this thesis.

To those who supported my faltering steps, to you these words, my dear mother, the symbol of giving and love who has showered me with supplications and contentment. You have been my best supporter throughout my academic career. To the owner of the words that led me towards success, my father, who is always present despite his absence. I dedicate this research to my father's soul, hoping that it will be goodness and charity for him. This research would not have been possible without the support of my family, special thanks to my wife and son Abdo for the numerous sacrifices, continuous support, and endless love. I would like to extend my grateful to my brothers and sisters. I am eternally indebted to them for all their unconditional love and support. Many thanks must also be made to my friends who had to listen to all my tales of woe whilst I completed the write-up of this thesis.

I am thrilled to have the opportunity to pursue a PhD at Edinburgh Napier University (ENU). These years were a significant period of my life during which I gained invaluable experiences. The time I spent at ENU was pivotal in my development and provided me with priceless insights. I would first like to thank my supervisory team Dr. Zakwan Jaroucheh, Prof. Ahmed Al-Dubai and Prof. Xiaodong Liu, for their guidance and support throughout my PhD. They constantly encouraged me to explore novel concepts and provided me freedom at work, which helped me to widen my perspective and cooperate with others. I will be eternally grateful for their time, as well as their insightful and valuable feedback and instructions on my work. Additionally, I would like to express gratitude to Prof. Bill Buchanan for his precious support, which was really influential in shaping my research journey. Finally, I would like to thank all my collaborators for their insightful inputs into my research which include Dr Imed Romdhani, Dr Baraq Galib, Dr Isam Wadhaj, and Dr Jawad Ahmed. I would also like to thank all team members in the School of Computing, Engineering, and the Built Environment at ENU, Cybersecurity Research Group and BIL lab, which includes Dr. Owen lo, Dr. Pavlos Papadopoulos, and Dr Nikolaos Pitropakis.

TABLE OF CONTENTS

List of Tables

Tables **Page**

List of Figures

First Chapter

Introduction

1.1 Overview

The Internet of Things (IoT) has transformed our lives and represented a significant step forward in how we will use technology. According to an IBM blog (Teicher, February 7, 2018), the idea of connecting sensing devices to the network return back to early 1980s. Before there was even a modern Internet, a group of students in Carnegie Mellon University's computer science department installed a board sensing the status of a soda machine in their department to track the machine content remotely. This was achieved by allowing a line from the board run to a gateway for the department's main computer, connected to the ARPANET, the precursor for today's Internet. However, the first concept for the Internet of things was put forward in 1999 by Kevin Ashton (Ashton et al., 2009) as a bind between the Internet and radio frequency identification. Technically, it refers to objects that can be connected to each other over the Internet. These objects could be any device embedded with software, electronics, or sensors. IoT enables real-time sensing capabilities, empowering various areas such as agriculture, transportation, healthcare, energy, smart homes, supply chain etc. The cause for this growth is the rising demand for industrial automation, which supports the development of IoT solutions. According to (Vailshery, 2021), an estimated 30.9 billion IoT connected devices are expected to be installed worldwide by 2025. This will

almost certainly increase the use of cutting-edge IoT applications and solutions. However, questions and concerns regarding security and privacy have arisen due to this technology's adoption. Despite its various applications, rapid expansion, and numerous far-reaching visions for the future, the IoT faces significant obstacles. Since IoT communication is wireless, it is subject to message manipulation, message eavesdropping, and identity spoofing. Every time a system is connected, an attacker has a new set of opportunities to find and exploit weaknesses. The more services provided via the Internet, the more services it can be attacked. This is referred to as the attack surface. Reducing the attack surface is one of the first stages in safeguarding a system (Wójcicki et al., 2022).

1.1.1 IoT security

IoT security being a sizzling topic for researcher today, there is a myriad of publication indicating security and privacy issues in IoT. The IoT must gain users' trust to be broadly accepted by the industry. Serious problems regarding people's privacy are brought about as a result of the increasingly invisible, dense collecting, processing, and distribution of data amid people's private life. The lack of key security measures in many of the initial generations of IoT systems currently available on the market contributes to an increase in the privacy concerns posed by the IoT (Barua et al., 2022). Digital security risks exist at every stage of the IoT journey, and hackers stand by to take advantage of any system flaws. Without sufficient security measures, attackers have the potential to seize control of critical elements of our lives. For instance, if an attacker successfully breaches the security of a smart healthcare system, it might result in the loss of many patients' lives, whereas a successful security attack on an intelligent transportation system can result in financial loss and the loss of human lives (Raghuvanshi et al., 2022). As a result, secure authentication and access control solutions are required. Authentication and access control technologies are known as the central elements to address the security and privacy problems in IoT (Trnka et al., 2022).

1.1.2 IoT security requirements

IoT architecture consists of different layers of technologies supporting IoT. According to (Iqbal et al., 2020), the main hurdle in the centralised environment of IoT is the inadequacy of privacy and security of sensitive data that is transmitted when devices communicate with each other or with the cloud. The security requirements of each layer are depicted in Figure 1.1 and further described below.

1.1.2.1 IoT devices' layer security

Protecting the devices layer is a difficult task, given the vast number of IoT assets and the potential vulnerabilities they include. Therefore, sensors in the IoT systems must be able to prove their identity to maintain authenticity. Furthermore, the integrity of the data send from devices in this layer must be signed and encrypted to prove that is not been tampered or changed. In addition, privacy is one of the requirements which must be considered when collecting data from sensors in this layer as many applications can provide traceable data, which can be considered as a privacy issue. (Qi et al., 2020).

1.1.2.2 The network layer security

This layer represents the connectivity between the devices layer and the cloud. Secure communication between things and cloud is one of the most important requirements. It is worth indicating that the current security technology that has been widely used is SSL/TLS encryption. However, the issue with solutions such as TLS and VPNs in resource-constrained IoT devices is that they consume more resources, such as memory, due to the requirement to allocate additional buffers and to have sufficient free heap memory to establish a TLS connection. Since cryptographic procedures are required, the processor overhead is also increased, especially when a certificate employs a long key length. (Ramesh et al., 2022).

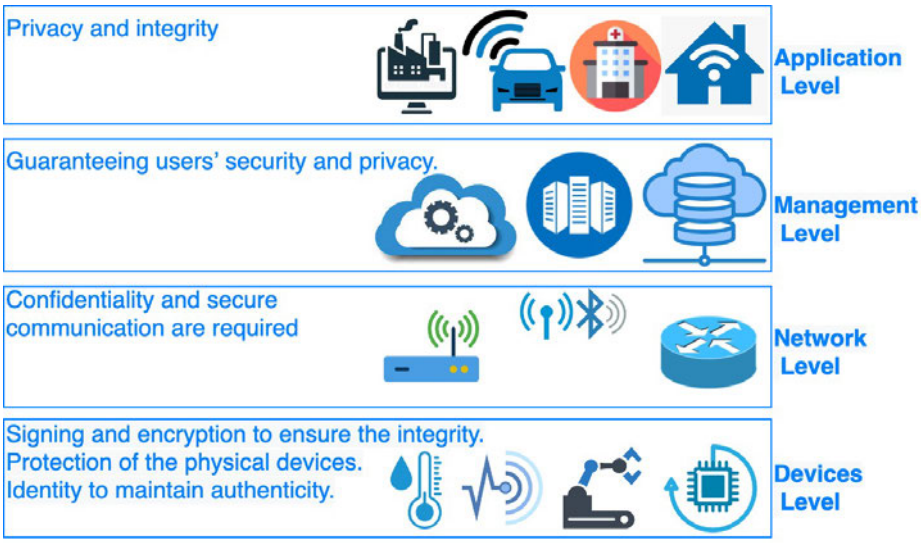

Figure 1.1: IoT Security Requirements

1.1.2.3 Application layer security

As more data is collected from various devices, guaranteeing integrity becomes a critical concern at the application level (Jaigirdar et al., 2019).

1.1.2.4 Management layer security

This layer is responsible for managing the IoT systems by managing users and devices access, applying rules and policies, coordinate the automation process across different devices, manage users and devices to identify and to provide an optimal access control based on the privilege assigned to them, and finally auditing & monitoring data. However, when user authentication data is stored on cloud servers, there is a risk that it may be shared or even sold to other organisations, infringing on users' privacy and security rights. As a result, there is scepticism about the level of privacy provided by the current centralised authentications and access control methods (Rasool et al., 2022).

1.2 Motivation

This section describes the main motivations behind dealing with privacy and security issues of current authentication and access control methods applied in IoT,

and why decentralisation matters for IoT. The following presents the fundamental challenges with the state-of-the-art and motivate the need for further research on this area.

1.2.1 IoT security challenges

- **Centralised data structure** Most state-of-the-art IoT infrastructures are heavily centralised, prone to a single point of failure. The entire network infrastructure risks being paralysed in the event of a failure in the centralised servers, which hinder scalability and wide adoption of the IoT. The centralised infrastructure has also raised severe privacy and security concerns because it relies on third party to maintain the trust. In addition, in the centralised model, users have limited control over their personal data. Users are required to trust such entities to handle their personal data, which is prone to the risk of being deleted or tampered with. Additionally, the centralised infrastructure leads to higher latency for end-to-end communications and lacks guaranteed accountability and traceability (Wójcicki et al., 2022).

- **Resource constraints** IoT devices are always resource-constrained, preventing them from implementing effective advanced security solutions. When encryption is used for authentication, for example, certain advanced encryption methods can result in issues such as decreased computing performance, increased hardware power consumption, and so on (Ullah et al., 2022).

- **Heterogeneity of IoT systems** Another challenge is the heterogeneity of the IoT system's components, as IoT systems might comprise devices from different vendors, each having its own platform and technology. Heterogeneity is seen in various IoT devices, communication protocols, and data formats. The heterogeneity is also the root of other challenges such as interoperability, privacy and security (Wójcicki et al., 2022).

- **lack of encryption** To protect sensitive data saved on a device, encryption should be employed at rest. Keeping API tokens or credentials in plain text on a device is a common security flaw. Even when data is encrypted, vulnerabilities may exist if the encryption is incomplete or incorrectly set. Another issue might develop when implementing cryptography in IoT-constrained devices, for example, when employing RSA with a 1024-bit key in a microcontroller with restricted RAM and storage (Ullah et al., 2022).

- **Complexity of networks** In the Internet of Things, a variety of communication and network protocols coexist. The data collected by IoT devices will be transmitted to server stations or other low-power devices. These devices require a software layer to enable access to hardware functionalities and protocols to connect these devices to other communication protocols (Ukwandu et al., 2022).

- **Poor interoperability** It refers to the ability of IoT systems (including hardware and software) to communicate, utilise, and collaborate on information. Due to the distributed nature of IoT systems and their heterogeneity, data interchange between different industrial sectors, strategic centres, and IoT systems is hard. As a result, IoT interoperability is difficult to achieve (Wójcicki et al., 2022).

1.2.2 The need for blockchain in IoT

Since the blockchain was first introduced in 2008 (Nakamoto, 2008), by Satoshi Nakamoto as an open-source project for secure financial transactions, it has played a prominent role in the cybersecurity field and opening the horizon for tackling problems in several other domains. The second generation of blockchain and distributed ledger technology is more general-purpose. Instead of recording financial transactions, it can record data for any other type of application. The decentralised structure of the blockchain offers a secure distributed ledger to store and

then validate transactions in a distributed manner. This will provide a secure verifications and authentication method where data will be trusted by all nodes in the blockchain network. In addition, blockchain can execute and deploy a script, called a smart contract (Buterin et al., 2014), such as on the Ethereum Blockchain platform, providing the ability to expand the usability of blockchain to include other emerging technologies that require a high level of secure verification, such as Internet of Things (IoT). Therefore, there is a significant opportunity for blockchain and decentralised approaches, which exclude the use of a third party to manage the trust. The decentralisation of trust is increasingly becoming a dominant direction, creating opportunities to manage authentication and authorisation in a decentralised and autonomous manner. Additionally, it is seen as a viable alternative to address privacy and security issues in current centralised identity management systems.

1.3 Thesis Objective

Given the motivations identified above in Section 1.2 and motivated by the promising advantages of blockchain and distributed ledger technologies, the main objectives of this thesis can be summarised as follows.

- This thesis aims to develop a blockchain-based decentralised authentication and access control mechanism for IoT by providing a reliable and privacy-preserving authentication framework that resolves security and privacy concerns in the current centralised paradigm and removes the need for third parties to maintain trust.

- The thesis further looks into the provision of the self-sovereign concept, which allows IoT clients to have full control over their own data. To achieve this, the thesis provided a blockchain-based decentralised identity management framework for secure and fair exchange, delegation, and revocation of credentials using the power of smart contracts.

- Due to the complexity and resource overhead in the legacy distributed consensus protocols, such as PoW, which hinder the development of lightweight and scalable blockchain-based IoT applications. The thesis further aims to develop a new hybrid blockchain consensus based on a reduced mining algorithm combined with the Practical Byzantine Fault Tolerance (PBFT) verification. Thus, preserving the security characteristics of the PoW consensus protocol while also improving the transaction's finality speed and reducing its energy consumption.

1.4 Main Contributions

The main contribution of this thesis is to adopt blockchain technology to develop a secure and privacy-preserving authentication and access-control framework satisfying security requirements in all aspects of the layered IoT structure, thereby overcoming security challenges outlined earlier in section 1.1. For users' authentication into the IoT applications layer, section 1.4.1 presents a blockchain-based two-factor authentication mechanism for web-based access to sensor data. Section 1.4.2 provides a lightweight authentication and authorisation mechanism for the Message Queuing Telemetry Transport (MQTT) protocol, a Machine-to-Machine connectivity and communication protocol. For authentication in the devices layer, section 1.4.3 provided a decentralised authentication for wearable medical devices along with a decentralised access control mechanism for medical data together with a decentralised identity model to manage users' identities. Section 1.4.4 presents the Proof-of-Notarized-Work (PoNW), a lightweight and scalable consensus mechanism that overcomes distributed consensus's resource overhead and complexity for IoT.

1.4.1 Blockchain-based User-centric Two-factor Authentication for IoT

Development of a blockchain-based two-factor authentication mechanism for web-based IoT applications. The proposed method is employed to authenticate and control users' access to data collected from IoT sensors via the web interface. The proposed approach addresses significant privacy concerns, eliminates the need for a third party to maintain trust, and mitigates the risk of using weak passwords for authentication into IoT applications. In addition, the proposed method presents a lightweight and user-centric authentication that makes use of blockchain and smart contracts technology. For this, we utilised a smart contract in the Ethereum blockchain to facilitate a secure and reliable two-factor authentication process. Blockchain in our systems is used to provide a secure independent channel for sharing and verification of the One Time Password (OTP). Our approach allows users to fully control their identities by utilising a blockchain-based decentralised identity model. Based on the evaluation results, our method has proven effective and can facilitate reliable authentication.

1.4.2 Decentralised identity and authentication mechanism for MQTT protocol

A lightweight authentication and authorisation scheme together with a decentralised identity system to manage the users' identities. This mechanism helps in facilitating the authentication for both subscribers and publishers by utilising a smart contract in Ethereum blockchain to guarantee trust, accountability and preserve user privacy. We provided a proof-of-concept implementation to prove our work, which involves a decentralised MQTT platform and dashboard using our approach. The usability of this approach was further analysed, particularly concerning CPU and memory utilisation. Our analysis proved that our approach satisfies IoT applications' requirements since it reduces the consumption of resources, and

the smart contract helps in the automation of the processes.

1.4.3 Decentralised authentication and access Control framework for wearable medical sensors data

The development of a blockchain-based access control framework for managing access to users' medical data. This solution leverages blockchain technology's inherent autonomy and immutability to solve the existing access control challenges, such as centralisation, heterogeneity, resource limitations, scalability, and privacy concerns that arose from relying on a third party to maintain trust and maintain customers' sensitive data, which makes it vulnerable to misuse and attacks. These challenges were presented in Section 3.3. This is facilitated by using a smart contract on the blockchain, which allows for delegated access control and secure user authentication. We have presented the solution in the form of a medical wearable sensor prototype and a mobile app that uses the Ethereum blockchain in a real data sharing control scenario. Based on the empirical results, the proposed solution has proven effective. It has the potential to facilitate reliable data exchange while also protecting sensitive health information against potential threats. As for security analysis and evaluation, the system exhibits performance improvements in data privacy levels, high security and lightweight access control design compared to the current centralised access control models.

1.4.4 Proof-of-Notarized-Work (PoNW) a secure and scalable Consensus

The development of a lightweight and secure blockchain consensus mechanism to reduce the number of nodes that need to achieve consensus, thereby reducing the overall energy consumption in the current PoW to meet the IoT requirements. To achieve this, we introduced a novel hybrid consensus algorithm that strikes a balance between the Byzantine Fault Tolerance (PBFT) and Proof of Work (PoW) consensus mechanisms. In doing so, our mechanism promises to provide a light-

weight and scalable consensus with an immediate block finality while also maintaining the security characteristics of the PoW consensus. Our contributions will involve proposing a ranking mechanism to resolve the chain fork based on the Pseudo-Random Process and a permutation function to arrange selected committee members into sequential order. In addition, we proposed a secure random model to select participants to perform PoW to stop an adversary from concentrating its presence in one committee and exceeding the byzantine-tolerance threshold. We presented the suggested mechanism and its implementation aspects, as well as security-related primitives and characteristics. Finally, we provided a security analysis of the model together with threat models.

1.5 Thesis structure

The dissertation is structured into eight chapters, which are briefly discussed below:

- **Chapter 2: Background.** This chapter provides background information on IoT, describing its architecture and its essential safety and security principles. In addition, this chapter will also provide an overview background on blockchain technology, illustrating its principal architecture and its unique characteristics.

- **Chapter 3: Literature Review.** The chapter commences through a literature review of initial findings, evaluation and hypothesis on proposals dealing with authentication, authorisation, access control, and identity management in IoT. The literature review summarises the most recent developments, challenges, and open research issues in authentication and access control for IoT. Then discusses blockchain technology and its application to secure authentication and access control in IoT, providing a thorough overview of current state-of-the-art efforts in this field.

- **Chapter 4: Blockchain-based User-centric Two-factor Authentication for IoT.** This chapter presents a decentralised framework for a blockchain-

based two-factor authentication mechanism for web-based access to sensor data. The proposed scheme will then be implemented, and the proof-of-concept model will be developed and evaluated to prove the system's effectiveness to facilitate reliable authentication.

- **Chapter 5: Blockchain-based identity and authentication scheme for MQTT protocol.** This chapter will introduce a blockchain-based approach to provide secure authentication and authorisation for Message Queuing Telemetry Transport (MQTT) protocol. This Machine-to-Machine communication protocol is being widely adopted in current IoT applications. It will further illustrate the proof-of-concept design and implementation of the proposed mechanism, which involves a decentralised MQTT platform and dashboard using our approach. The usability of this approach was further analysed, particularly concerning CPU and memory utilisation compared to the current centralised secure solutions.

- **Chapter 6: Decentralised Authentication and Access Control Mechanism for Medical Wearable Sensors Data.** This chapter introduces a blockchain-based access control framework and authentication mechanism for medical wearable sensors. The proposed solution will be discussed and presented in the form of a medical wearable sensors prototype that utilises the Ethereum blockchain in a real data sharing control scenario. Then, it provides evaluations of the empirical results of the proposed solution to prove its effectiveness in facilitating reliable data exchange while protecting sensitive health information against potential threats.

- **Chapter 7: PoNW: A Secure and Scalable Proof-of-Notarized-Work Based Consensus Mechanism.** This chapter presents a new hybrid consensus mechanism known as the Proof of Notarized Work (PoNW). The proposed consensus design, as well as its components, will be discussed.

Then provided security analysis in the proposed consensus and provides a threats model to insure an acceptable failure probability.

- **Chapter 8: Conclusions.** Finally, the conclusion chapter, which will discuss the main arguments and contributions. This chapter demonstrates a summary of the research contributions made in this thesis, and the research outcomes will be used as a base for further development and for the future research directions. Then, discusses the open research questions of integrating blockchain and IoT, and provides a summary of the future work that is intended to address those concerns with regard to each of the key chapters of this thesis.

Second Chapter

Background

Before we dive deeper into our methodologies of integrating IoT and Blockchain technology, there must be a clear understanding of IoT and blockchain technology and its suitability to satisfy the underlying security requirements of IoT. This is because both Blockchain and IoT are emerging technologies. To achieve this goal, in this chapter, we aim to provide a coherent and comprehensive overview of IoT and blockchain technology. Therefore, this chapter would serve as a background to provide the essential knowledge on blockchain and IoT technologies required for later sections of this thesis. The chapter starts by providing background information on IoT, describing its architecture and essential safety and security principles in section 2.1. Then section 2.2 looks into the IoT protocol stack, illustrating a variety of IoT communication and network protocols. In addition, section 2.3 provides an overview background on blockchain technology, illustrating its principal architecture and its unique characteristics. Then, the different blockchain architecture layers are presented and discussed in more detail in sections 2.3.1 to 2.3.5. Finally, in section 2.3.2, we present blockchain types and their characteristics.

2.1 IoT layered structure

The architecture of the IoT is a framework that outlines the physical components, the network's functional organization and configuration, operational procedures, and data formats to be used (Raj and Shetty, 2021). Although there is no globally

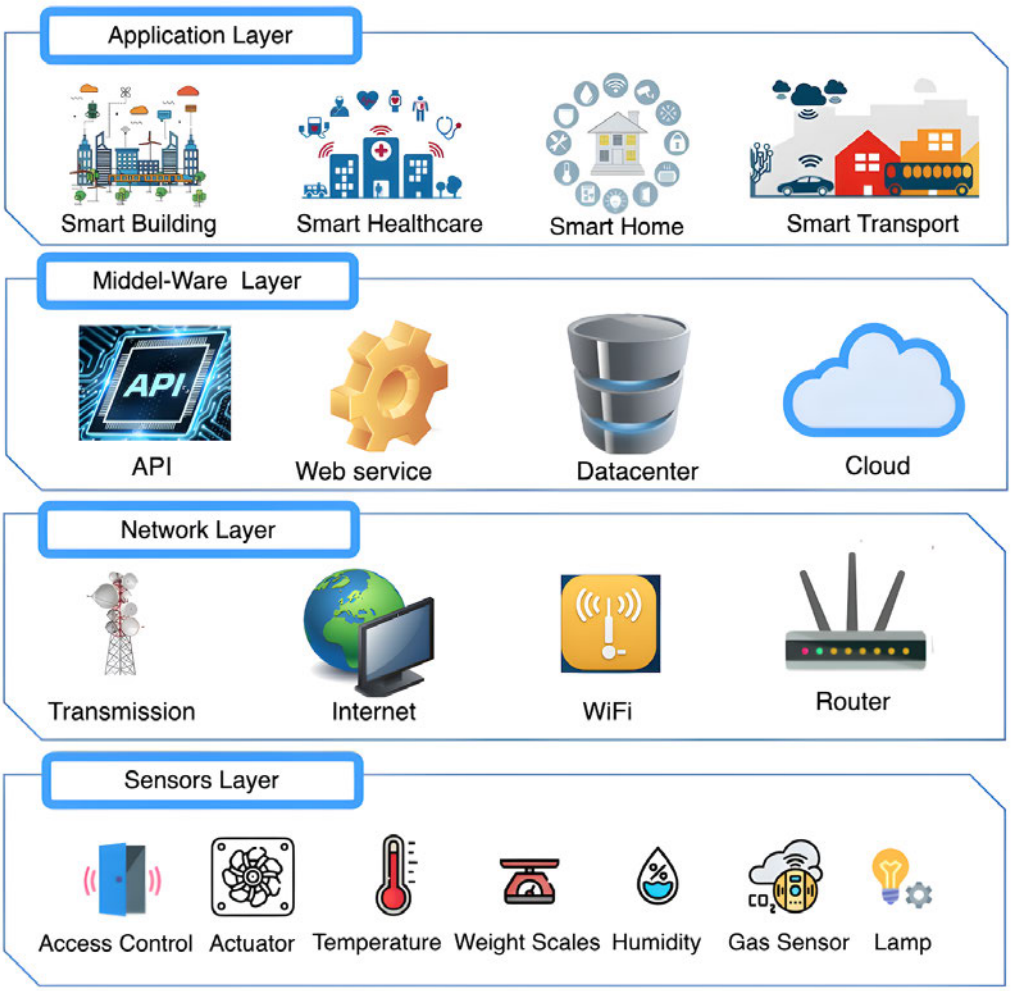

Figure 2.1: IoT layered structure

acknowledged IoT design, the most basic and frequently accepted model is a three-layer architecture, combining Perception, Network, and Application. This architecture, nevertheless, does not include new technologies like fog and cloud fog computing because it was created during the early stages of the IoT research. As a result, the infrastructure must be able to support the architecture of the IoT. The ITU-T (Darwish, 2015) proposes four levels that are critical to the overall feasibility of an IoT implementation as shown in Figure 2.1, which will be discussed below.

2.1.1 Application layer

All applications that leverage IoT technology or in which IoT has been deployed are defined by the application layer. Smart homes, smart cities, , animal tracking, and other IoT applications are possible. It is responsible for providing services to

the applications (Jabraeil Jamali et al., 2020). Because services are dependent on information collected by sensors, services may vary for each application. There are numerous challenges in the application layer, with security being the most important. The most common application layer security threats and issues are cross site scripting, malicious code attack, the ability of dealing with Mass Data (Gupta and Quamara, 2020a).

2.1.2 Application support layer

A new layer is proposed as a result of the limitation of the three-layer design in representing the newly introduced technologies. The addition of a fourth layer is motivated by security concerns, as information transferred directly to the network layer in a three-layer architecture increases the risk of threats. Information from a perception layer is delivered to a support layer in a four-layer architecture (Darwish, 2015). The support layer is responsible for two issues. Firstly, it ensures that information is sent by legitimate users and that it is safe from dangers. There are numerous methods for verifying users and information. Authentication is the most widely used mechanism. Pre-shared secrets, keys, and passwords are used to implement it. Secondly, it is used to send data to the network layer. Wireless and wired media can be used to send data from the support layer to the network layer. This layer is vulnerable to a variety of assaults, including Denial of Service (DoS) attacks, malicious insider attacks, illegal access, and so on (Gupta and Quamara, 2020a).

2.1.3 Network layer

Responsible for network connectivity-related operations such as mobility management, authentication, authorisation, and accountability, as well as IoT transport management data. The transmission layer is also called network layer. It serves as a link between the perception and application layers (Jabraeil Jamali et al., 2020). Through sensors, it carries and communicates the data collected from

physical objects. The transmission medium can be wireless, or wire based. It is also in charge of connecting smart items, network devices, and networks to one another. As a result, it is extremely vulnerable to attacks. It poses serious security vulnerabilities with the integrity and authentication of data being transmitted across the network. The most common network layer security threats are DoS Attack, Man-In-The-Middle (MITM) Attack, storage attack, exploit attacks (Gupta and Quamara, 2020a).

2.1.4 The Devices layer

The elements of processors, memory, firmware, sensors, and actuators, as well as their features, are represented by the devices and gateways. Device features include the capacity for devices to engage directly with the communication network; they can gather and deliver data without the use of a gateway. Support for numerous interfaces is one of the gateway's advantages, allowing IoT devices to communicate through a variety of wired and wireless protocols, such as ZigBee, Bluetooth, and Wi-Fi. However, attackers who want to use them to replace the sensor with their own are primarily after them (Jabraeil Jamali et al., 2020). As a result, the sensors layer is the source of most threats, such as eavesdropping, node capture, fake and malicious node, replay attack and timing attack.

2.2 IoT Protocol Stack

The Internet Engineering Task Force (IETF) has designed a possible procedure for transmission among IoT modules employing IP because IP is adaptable and dependable medium. The Internet Protocol for Smart Objects (IPSO) Association has reported much research representing possible procedures and mediums for the IP stack mediums and furthermore adding adaption layer, which is employed for transmission among smart objects (Dunkels and Vasseur, 2010). Figure 2.2 represents the Protocol stack of IoT. New protocols have been proposed to replace or flank the TCP/IP stack protocols due to a lack of computational resources

Table 2.1: IoT protocol stack

Web Stack	OSI	IoT Stack
Web Application, HTML, XML, JSON, HTTP, HTTPS, DHCP, DNS, TLS/SSL	Application Presentation Session	Data, CoAP, MQTT, AMQP, XMPP, JSON, CBOR
TCP, UDP	Transport	TCP, UDP, DTLS
IPv6, IPv4, IPsec	Network	RPL, IPv6, IP routing, 6LowPAN
Ethernet, DSL, ISDN, WLAN, Wi-Fi	Data Link	IEEE 802.15.4 MAC
Physical	Physical	IEEE 802.15.4 PHY radio

and various devices and traffic. Instead of the TCP/IP stack's application-level protocols, the Constrained Application Protocol (CoAP) (Shelby et al., 2014) is used, a lightweight version of the Hypertext Transfer Protocol (HTTP) that is ideal for interacting with low-resource devices and sensors. At the transport level, CoAP uses the User Datagram Protocol (UDP), which delivers fewer services but is substantially lighter than the Transmission Control Protocol (TCP) used by HTTP. Finally, an adaption layer is added in which IPv6 packet headers are encapsulated and compressed using the IPv6 over Low Resources Wireless Personal Area Network (6LoWPAN) protocol to manage them with devices minimum CPU power (Rayes and Salam, 2019). Figure 2.2 depicts IoT Network Protocol Stack and presents the new protocols that have been proposed to replace or flank the TCP/IP stack protocols.

2.2.1 MAC and Physical Layer

The IEEE 802.15.4 standard is introduced to provide transmission between densely packed and low-energy implanted mediums that require a longer battery life. It establishes standards and measures for the IP storage's MAC and physical media. It is in charge of a small amount of energy transmission and low-cost and low-area transmission. Because of the limited resources available, we want a portable structure with minimal transmit power and bandwidth. Communication uses a small

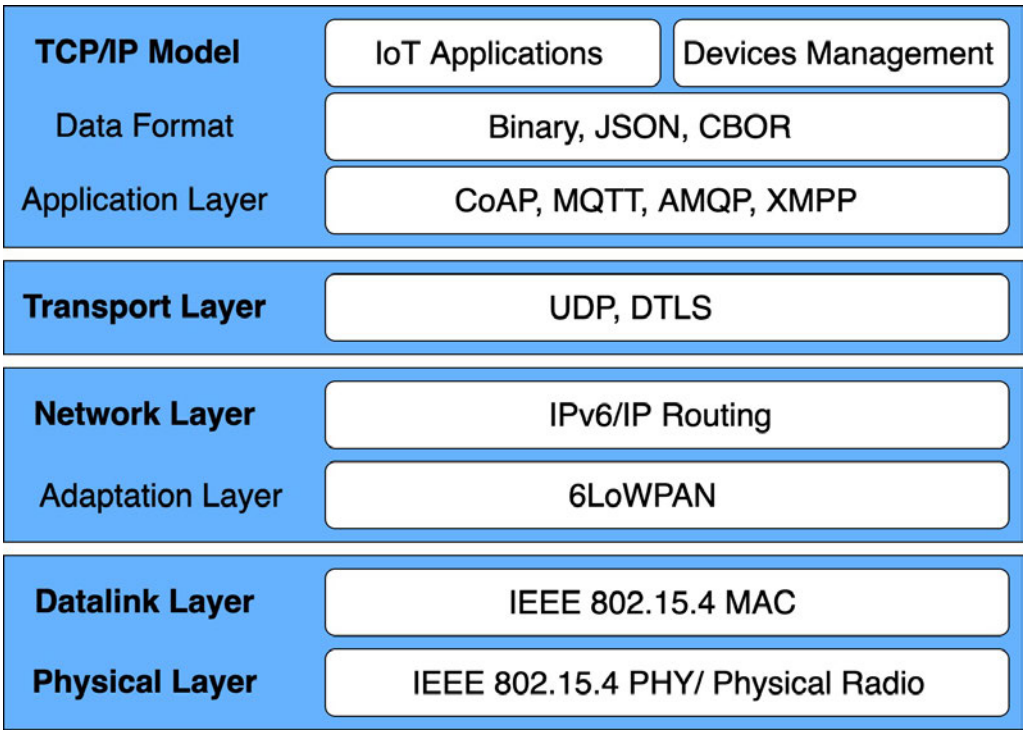

Figure 2.2: IoT's Security and Network Protocols Stack

amount of power, about a quarter of what is used in mobile or WiFi clusters (Rayes and Salam, 2019). The transmission medium is therefore maximized. Because there is a limited region, the modules must work together to enable multi-hop routing across large areas. This resulting in the packet medium is restricted to 127 bytes and the medium of transmission is restricted to 250kbps. The programming function in IEEE 802.15.4 has a built-in discharge, which possesses strong transmission, authorize us to identify losses and validate the recommunication of missing packets (Rayes and Salam, 2019).

2.2.2 Adaption Layer

Because of its adaptability and stability, IPv6 is regarded as the superior method for transmission in the IoT platform. Initially, mass IP standards like IEE 802.15.4 were not expected to be suitable for transmission in low-energy wireless frameworks. The 6LoWPAN is an excellent established medium for wireless transmission since it is a composition for IPv6 across low-energy wireless unique range clusters. It allows IPv6 transmission over the IEEE 802.15.4 l transport and link channel. It

may communicate with the system's various IP components. It was chosen because of the most extensive label range achievable in IPv6, and it is connected to the internet through a router. IPv4 is currently the most widely used Internet protocol; it also maintains standard maintenance for changes between IPv4 and IPv6. The headers in IPv6 are insufficient to fit within the 802.15.4 protocol's tiny 127-byte MTU. To reduce transmission overhead, the adaptive layer performs the following optimizations individually (Rayes and Salam, 2019).

2.2.3 Network Layer

For overthrow the obtained information from transport medium, the network layer plays an important role. The routing protocol (RPL) for Low Power and Lossy Networks has been developed by the IETF Routing Over Low Power and Lossy Networks (ROLL) functional organisation (Winter et al., 2011). RPL is an attainable gateway standard for these systems, installed on the different nodes. It specifies how to create a DODAG (Destination Oriented Directed Acyclic Graph) with the edges following the interchange distance vectors. A set of constraints and targeted action is used to construct the representation with the superior route. According to their specifications, the targeted function and constraints may differ. For example, restrictions could be used to avoid battery power mediums or to progress encrypt media. The target function aims to lower the latency or the anticipated number of packets that desire to post. Junctions are divided into storing and non-storing junctions based on their ability to stack gateway data to meet their memory requirement. When junctions are on non-storage mediums, and a descending route is created, the gateway data is attached to the next context and sent across the source as well. The source encounters the packet throughout and provides an information message to the target, leap by leap, with the routing packet (Winter et al., 2011).

2.2.4 Transport Layer

The transport layer's principal protocols are TCP and UDP. TCP, on the other hand, is not a better method for transmission in a low-energy environment because it has a large overhead because it is a link-based protocol. As a result, UDP is preferred because it is an unconnected standard with a short latency (Rayes and Salam, 2019).

2.2.5 Application layer

This platform oversees the configuration and presentation of information. The internet is generally established using HTTP in this media. On the other hand, HTTP is not fit in a constrained measure habitat because of its extremely expanding temperament, and this experience is in desperate need of a thorough examination. Many protocols such as MQTT (Message Queue Telemetry Transport) (Standard, 2014) and CoAP (Constrained Application Protocol) (Shelby et al., 2014) are being implemented.

2.2.5.1 Message Queue Telemetry Transport (MQTT)

It's a lightweight standard ideal for IoT applications. MQTT is a message-passing protocol that works over TCP. IBM was the first to introduce it as a client/server standard. The clients communicate with one other, while the server acts as a bridge/broker for users who connect to TCP. A topic could be subscribed to or published by users. This transmission connects the broker, whose job is to consider presentations while also validating the user's privacy. It cannot be used with all IoT implementation methods since it works with TCP. Furthermore, it uses context for research names, maximizing its overdue (Standard, 2014).

2.2.5.2 Constrained Application Protocol

It's a different concept from HTTP. It is frequently used in a variety of IoT setups. It incorporates expansion for confined implementation habitat, unlike HTTP. It

Table 2.2: IoT protocols stack and security protocols

IoT Layer	IoT Protocols	Security Protocol
Application	CoAP, HTTP	User-defined
Transport	UDP, TCP	TLS, DTLS
Network	IP	IPsec
Routing	RPL	RPL security
6LowPAN	6LowPAN	N/A
Data Link	IEEE 802.15.4	802.15.4 seecurity

uses the efficient XML Interchange information, which is a binary data format that saves a lot of space compared to plain text HTML/XML. Some of the different properties are resource identification, multicast message support, header compression, congestion control, auto-configuration, and asynchronous message exchange. Acknowledgement, reset, non-confirmable, and confirmable are the four message types in CoAP. Sustainable packets are used in UDP for successful communication. The acknowledgement may be returned to the sender. It also uses DTLS for security reasons (Datagram Transport Layer Security) (Shelby et al., 2014).

2.3 Blockchain technology

Blockchain is a decentralised, cryptographically secure, immutable, transparent, and traceable system that can only be restructured by most of the blockchain network's existing peers (Nakamoto, 2008). It is a peer-to-peer network that manage a distributed ledger of immutable records. Through consensus procedures, all nodes in the network maintain the blockchain's integrity and correctness. The blockchain's security is enhanced by placing trust in a network of nodes. Users connect to the blockchain and initiate a transaction using their private key as a signature. This transaction is delivered to a transaction pool, which will remain until a miner fetches it into a block. After aggregating transactions from the pool and determining the block's valid hash, the miner creates a new block. When a miner successfully generates a new block, it is broadcast to the P2P network's

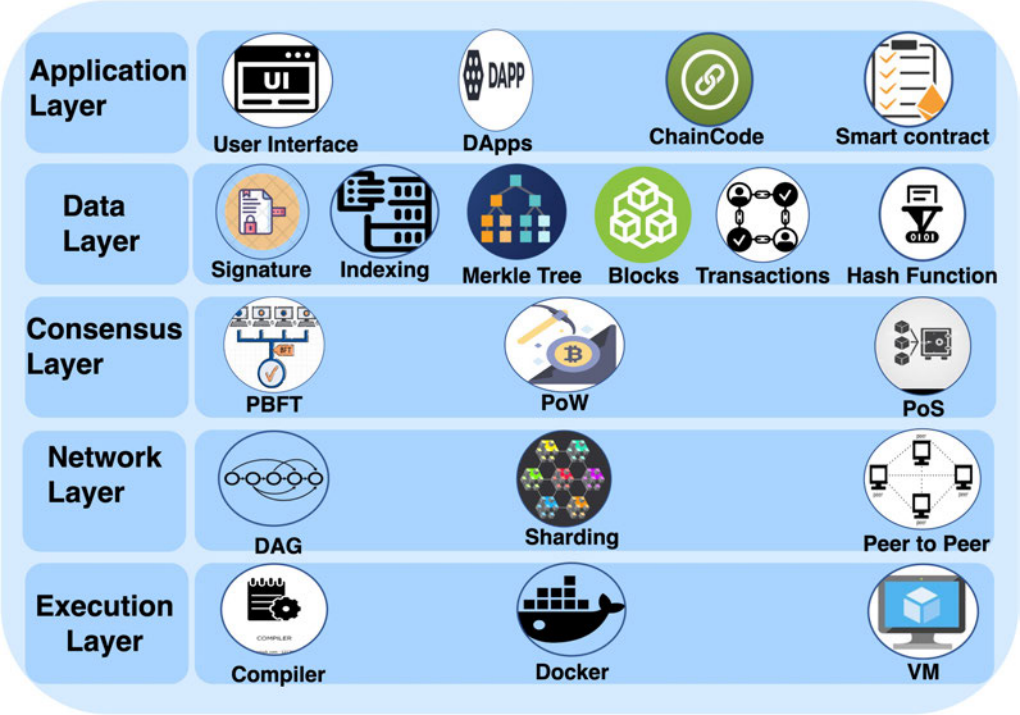

Figure 2.3: Blockchain Layers

nodes. The block is verified by all nodes in the network using a consensus process, and if successful, it is updated to their copy of the chain, and the transaction is complete. From the standpoint of an architecture, blockchain, linked-list information structure that establishes a linking through using the preceding block's hash. In the blockchain, every block consists of their hash, a set of transactions and the prior block's hash; the blockchain's connection to the previous hash makes it immutable (Nakamoto, 2008). This section will explain and illustrate blockchain technology and discuss various blockchain kinds and implementations.

2.3.1 Layered blockchain architecture

This section explains the layered architecture of blockchain technology, which consists of five layers. Although, there are many other representations possible that are presented in numerous other studies (Newell et al., 2021). However, the one we present provides a basic overview. We will delve more into the layered structure of blockchain illustrated in Figure 2.3 below.

2.3.1.1 The application layer

The application layer is in control of creating blockchain solutions that may be used in various applications and sectors. Smart contracts, chaincode, and decentralised applications (dApps) make up the application layer. The layer is divided into the following sub-layers: the application and the execution layers. Scripts, APIs, and the user interface are part of the application layer. These tools allow the application layer to communicate with the Blockchain network. Smart contracts, chain code, and underlying rules are part of the execution layer. The execution layer receives instructions from the application layer and executes them. For instance, directives are sent to Hyperledger Fabric chain code and Ethereum Virtual Machine smart contracts. The application layer's components are listed below.

- **Smart contract** Smart contracts which were introduced by Sbazo (Szabo, 1996), refer to as computerized transaction procedures which implement contract conditions on a blockchain. The main objective of smart contracts is to satisfy common contractual constraints, lessen unintentional and malicious exceptions, and eradicate the necessity for trusted intermediaries. A variety of blockchains now supports the smart contract. Ethereum initially introduced the smart contracts as a publicly accessible platform in 2014 (Buterin et al., 2014). A smart contract is a code that contains an enterprise's logic and is recognized by an exclusive address on the Ethereum Virtual Machine (EVM). When a transaction is performed against such functions, a smart contract has executed functions. A transaction can result in a change of state in the smart contract, depending on the smart contract's logic. Developers can design smart contracts in a high-level language like Solidity, compile the code into bytecode and deploy the bytecodes to the blockchain using a particular compiler like Remix. The bytecode is smaller and runs faster on EVM because it is compiled. The smart contract is given a unique address once it has been launched. Any user on the blockchain can

utilize that smart contract to make a transaction. The code that runs on EVM is entirely isolated and has no interface with the network or disk. However, there are some codes that connect smart contracts to the world, such as inter-blockchain, logic execution, and so on (Buterin et al., 2014). Oracles and dApps are the terms for this. There is no VM like EVM in Hyperledger Fabric (Ethereum). Smart contracts operate on a peer node controlled by a company and are primarily written in typical languages like Node.js, JavaScript, and Go. Chaincode (Freeman, 1961) is deployed on network nodes, and smart contracts are mainly written in standard languages like Java, Node.js, and Go. Each blockchain instance has access to a secure Docker container that runs chaincode. The peer nodes orchestrate the chaincodes, which operate as proxies, giving client applications via REST APIs or SDK. In the chapter 3, we will review numerous authentication and access management strategies that made use of a smart contract.

- **Chaincode (Hyperledger Fabric)** It is in this environment where business items' life cycles are controlled by smart contracts. Before the chaincode can be installed to the network of the blockchain, quite a few of related smart contracts gave to be packed together in the chaincode. For the channels to be initiated, the chaincodes have to be presented. An endorsement policy for a chaincode can be defined by the administrator for a specific channel. Through this, it guarantees that the chaincodes that are packed in all the smart contracts are available for that given channel. Depending on the endorsement policies of the configured channels, the chaincode might follow the endorsement policy which it was designed for the different channels. Depending on whether the smart contracts are on the same channel or a different one, they are able to interact with other smart contracts. In the Hyperledger Fabric, the chaincode oversees the packaging and deployment of the smart contracts

(Androulaki et al., 2018). Additionally, the ledger's data's schema, it's initiation, performance of ledger updates based on consensus, as well as responding to the enquiries of the ledger's data are all defined by the chaincode. Events are also emitted by the chaincode, which permits other applicants to subscribe to the events of the chaincode and undertake subsequent processes of downstream. In the Hyperledger fabric, unlike the EMV, standard languages like Go, Node.js and Java are written in the chaincode and deployed on peer nodes held by various organisations. The chaincode operates on a secure Docker container. Through SDK or REST APIs, client applications can access the chaincode. The administrator decides on the endorsement policy running on a channel, once the chaincodes are initiated for the channel.

- **Oracles** Only when a defined logic is met can it trigger a contract state change; otherwise, a smart contract usually operates on values. For smart contracts, the agent whose responsibility is to securely provide these values is called an Oracle. From third-party services, data feeds, which are like Oracles, supply values to the smart contracts (Al-Breiki et al., 2020).

- **dApp** This refers to a distributed application operating on top of distributed technologies such as blockchain, including Bitcoin, Hyperledger and Ethereum. However, it is a decentralized app which can communicate with the chain code or blockchain through smart contracts. Since it is decentralised, it is not controlled by a single entity like other apps, once it's deployed on the network of the chaincode. dApps are easy-to-use applications that allow business users to transact on a blockchain network. You can link to blockchains with smart contracts, but you can only connect to a smart contract or chaincode with dApps (Leiponen et al., 2022).

2.3.1.2 The data layer

Where transactions are ordered, the blockchain data structure can be expressed as a linked list of blocks. A linked list and pointers are two major components of the data structure of the blockchain. The variables that refer to the location of the other variables are the pointers. The linked list meanwhile, consists of a list of chained blocks, whereby each block has the pointers and data of the predecessor blocks. The blockchain, being a massively distributed ledger, is decentralized, whereby transactions are placed and arranged in a peer-to-peer network (Newell et al., 2021). The state of all the accounts is kept in such a manner. A private or public network consists of many nodes; thus, data cannot be altered, without a common consensus. Transactions, Blocks, Digital Signature, Hash function, Merkle tree, and other elements make up this layer. Significant components of this layer are discussed below:

- **Transactions** These refer to data structures that are kept in block forms, which form a chain to be a Blockchain since transactions are often connected to one another. Once a transaction is obtained by a miner, they save it from incorporating it in the following block to be mined. The transaction becomes public and immutable once this block is added to the chain. A public key system is used to sign transactions. The transaction's owner must sign transactions using the private key to establish ownership. It is critical to note the user's public key who will obtain the value to encrypt the transaction so that it can only be deciphered by the private key holder that matches the target public key. In this approach, the system can be open to the public while only the transaction owner can access it. Once a transaction message is received by miners, it is stored in a database of unmined transactions. Transactions are placed in a priority queue depending on arrival time and charge taxes until they are removed from the block to be included in the next one. Every miner has its transaction queue and can choose

which transactions to include in the new block. It will construct a Merkle tree and include the value of its root in the header after picking which transactions to include. Now it's missing the value of the nonce that will be included in the new block; this is the time-consuming phase of the process, demanding a lot of processing power from the miners and, as a result, a lot of energy. After each failure, the nonce is increased until a valid hash is discovered. The majority of transactions include a digital currency amount termed a transaction cost (Beck et al., 2016). This is referred to as a fee, and it will be paid to the person who mined the block that records transactions. This fee, or the transaction cost, will be paid to the miner as an incentive to make the blockchain system secure and consistent. The blockchain account or wallet calculates this charge or transaction cost autonomously.

- **Merkle tree** As discussed in the preceding section, transaction records are arranged in a Merkle tree. A Merkle tree is a binary tree structure that allows and summarizes big data set to be examined fast and firmly. If the transactions are not packaged into Merkle trees, each network node will have to retain a complete copy of every transaction made on the Blockchain (Sheth and Dattani, 2019). Figure 2.4 below illustrated the Merkle tree architecture.

  A Merkle tree sums up all transactions within a Block by providing a digital fingerprint of the entire collection of transactions, allowing a user to determine whether or not a transaction is included in the Block. The Merkle tree root is likewise altered when a single transaction is. The Merkle tree root formed while constructing the Block is one of the fields in the Block's header. Merkle trees are created by continuously hashing node pairs until only one hash remains, known as the root hash or Merkle tree root. Each non-leaf node has a hash of its previous hashes, and each leaf node has a hash of its prior hashes. A Merkle tree provides security,

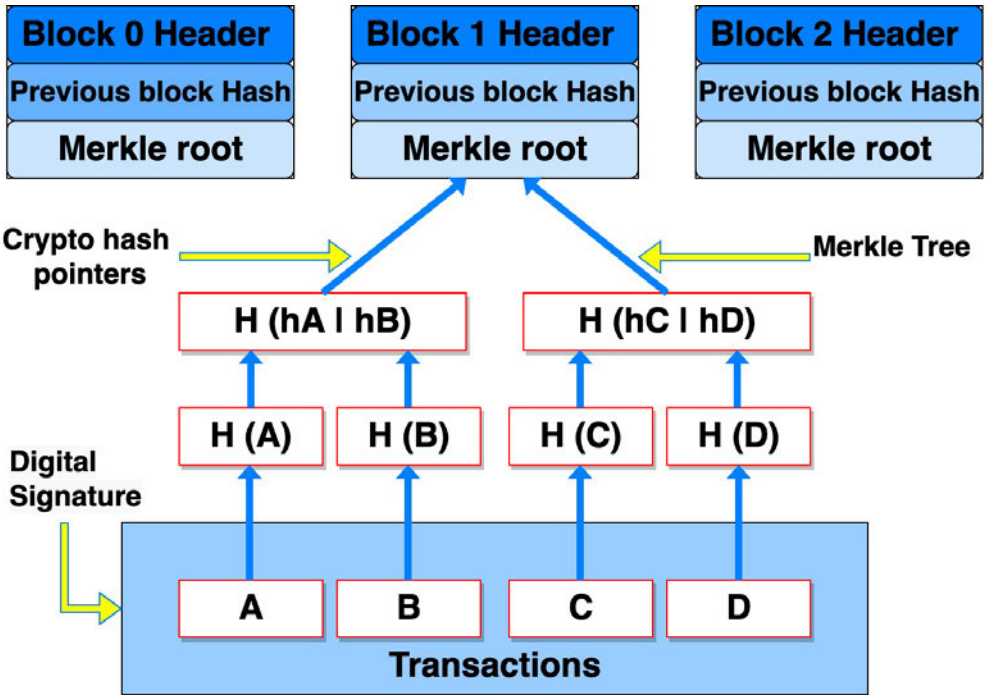

Figure 2.4: Merkle tree architecture

integrity, and irrefutability for blockchain technology. The blockchain system is built on Merkle trees, cryptography, and consensus algorithms (Sheth and Dattani, 2019). The Ethereum blockchain, for instance, stores data in a Patricia tree database. Patricia tree is a Merkle tree that functions similarly to a key-value store. Patricia trees, like Merkle trees, have a root hash. This root hash can refer to the entire tree. As a result, you can not change the tree's content without altering the root hash. Each Block comprises a list of transactions that have occurred since the previous Block, and the Patricia tree's root hash represents the new state after those transactions have been applied (state tree). Because it is the first Block in the chain, the genesis block (the first Block) does not contain the pointer (Buterin et al., 2014).

- **Data block** The public record of all transactions carried out in a blockchain system is the blockchain. After a block is formed, it is added to the blockchain as a permanent database from which new blocks can be constructed. Blockchain systems are made up of a chain of blocks that are sorted and timestamped, with each block containing the previous

block's hash as shown in Figure 2.5. The hashes are created through the SHA 256 hashing technique, and the headers of every block refer to its parent hash and link them back to the genesis block, which is the first block in the blockchain (Buterin et al., 2014). As a result, all validated Blocks in the chain can be traced back using cryptographic hash codes; no change or alteration of Block data is possible.

- **Block's Structure** The header and transactions are the two most important sections of a block. The data recorded in the block is known as transactions. The header, in turn, comprises various fields, the most essential of which are the hash of the preceding block, nonce, difficulty, and Merkle tree root for its operation. Apart from that, two metadata must be understood: header hash and block height used to categorize the block and its location in the chain (Buterin et al., 2014). These fields will be described in detail below. In general, the Block header contains the following information:

Height: In the chain, the blocks are added in chronological sequence, and each new block is assigned an order number. The height is the difference between the last block's number and the first block's number. This field is not always utilized to identify a block because there could be two or more blocks with the same height at any given time. A fork in the chain occurs in this situation.

Header hash: It's the block's primary identifier. The block header is used as input for a cryptographic digest process. It is not part of the data structure of the block and is not sent over the network. Upon receiving a new block, each full node computes it. They then store it as part of the block metadata in a separate database. Unlike the height, the header hash can identify a block.

Figure 2.5: Transactions' Chain (Nakamoto, 2008)

Hash of the previous block: This field is in the header to allow the block to connect to the one before it. Block 2 has the hash of block 1 in its header, as seen in Figure 4. The metadata of the block is stored in the full nodes. As a result, all nodes have the hash of block 1. When a complete node receives block 2, it checks this field and determines that block 2 is the child of 1.

Nonce: This is an integer used as a variable to change the output of the header hash. It is used in conjunction with the difficulty field to demonstrate that a miner has completed a task. If the difficulty requires that the header hash begins with a series of three zeros, the miner will iterate the nonce until it matches that condition. The full nodes will only calculate the header hash once they get a new block to determine if the nonce is legitimate.

- **Hash function** A hash is a mathematical function that turns an arbitrary-length input into a fixed-length encrypted output. As a result, its unique hash will always be the same size, regardless of the original quantity of data or file size involved. A cryptographic hash algorithm (such as the SHA 256 algorithm) can provide a data hash value with a specified length. These hashes aid in the easy identification of blocks and the

detection of any changes made to the blocks. Blockchain is simply a chain of hashes because each block has a previous block's hash. In a blockchain system, hashes are employed in various places. First, each block includes a hash of the previous block's block header, ensuring that nothing has been tampered with when new blocks are added (Nakamoto, 2008). As a result, the security of hash functions is critical to blockchain security. If the hash function is compromised, the entire system's security is compromised.

- **Digital Signature** Transactions are digitally signed to protect the security and integrity of the data recorded on the Blockchain. It uses asymmetric cryptography to secure information about the block, transactions, and transacting parties, among other things. A private key is used to sign transactions, and anyone with the public key may verify the signer. The digital signature is tamper-proof. Because the encrypted data is also signed, digital signatures ensure data integrity. As a result, any manipulation will render the signature invalid. The data cannot be identified because it is already encrypted. It cannot be tampered with even if it is detected. A digital signature also protects the sender's (owner's) identity. Owners (users) are linked to private keys: As a result, signatures are legally connected to the owner and cannot be revoked. The electronic signature associated with the original material is verified by the receiver's signature algorithm in two steps: 1) creating the message's hash or digest 2) using the sender's public key to decrypt the attached digital signature (Cachin et al., 2016a). The data has not been modified if both digests are identical. Otherwise, the message or signature has been tampered with, or the digest has not been encrypted using the associated public key's private key.

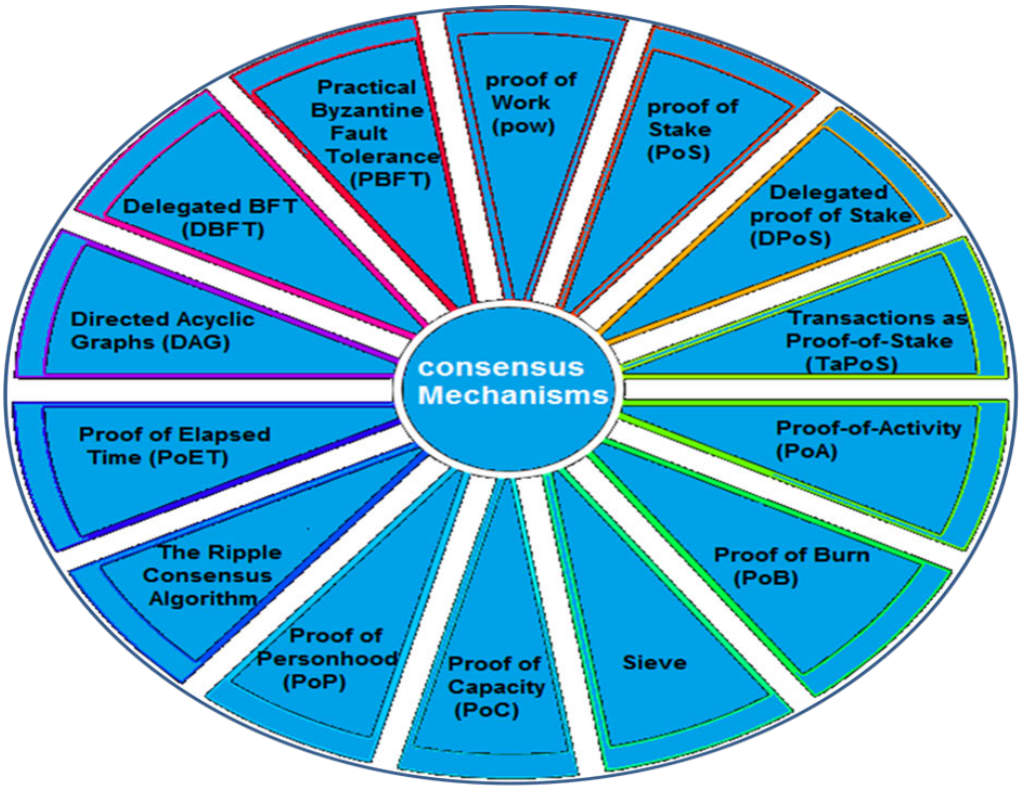

Figure 2.6: Blockchain consensus mechanisms

2.3.1.3 The consensus layer

The consensus protocol is essential for blockchain platforms to exist. A consensus algorithm is a means of obtaining an agreement between several insecure nodes on a particular data block in the Blockchain setting (Cachin et al., 2016a). This layer is responsible for enforcing network rules that specify what nodes should do to establish consensus on broadcasted transactions. It also has to do with block generation and verification. Different varieties of Blockchain have other consensus processes (Ferdous et al., 2020). Consensus, for example, is known as probabilistic consensus when it is obeyed by a permissionless blockchain network like Ethereum, Bitcoin, and so on. Although there is a chance that different participants have different perspectives of the blocks, such a consensus ensures the ledger's consistency.

As a result, they are still vulnerable to ledger forks [36]. The longest chain is consistently chosen as the one to which the new block is always appended. Deterministic algorithms are used in permissioned blockchains like Hyperledger

Fabric. Ordering nodes are specialized nodes in blockchain networks, and blocks validated by these ordering nodes are regarded as final and authentic. As a result, there is no chance of a fork (Pahlajani et al., 2019). Figure 2.6 presents various consensus mechanisms, such as Proof of Work (PoW), Proof of Stake (PoS), Byzantine Fault Tolerance (BFT), Proof of Authority (PoA), and Proof of Elapsed Time (PoE) (PoET).

2.3.1.4 The Network layer

The network layer also referred to as the P2P layer, is in charge of inter-node communication. It handles block propagation, transactions, and discovery. Propagation layer is another name for this layer. The P2P network ensures that all nodes can discover and connect to one another, allowing blocks to be propagated throughout the network and the blockchain's valid, current state to be synchronized. A peer-to-peer (P2P) network is a computer network in which computers (nodes) are distributed and share the workload of the network to achieve a common purpose (Nakamoto, 2008).

- **Nodes** Nodes are an integral part of the blockchain system which represent clients or computers that connected to the blockchain network. Nodes have been used to achieve various tasks such as mining, routing and serving as a wallet by storing a copy of the blockchain data. Also, nodes are responsible of discovering the directly connected peers to the blockchain network. Moreover, nodes are also responsible of stablishing and maintaining connection with their discovered peers. All nodes involved in the verification process and propagation of transactions (Nakamoto, 2008). In addition, nodes work as distributed ledger by saving a copy of the blockchain which contain information about all the transactions that have been maintained in the blockchain system, thereby replaces the uses of centralised server to store the transactions details with decentralised and distributed ledger. Finally, nodes can

also work as miners and can be awarded cryptocurrencies for verifying and validating the transactions that have been made by all users in the blockchain system. Full nodes and light nodes are the two types of nodes. Full nodes are responsible for transaction verification and validation, mining, and enforcing consensus rules. They are in charge of maintaining network trust. Light nodes merely keep the blockchain's header (keys) and can send transactions (Gao et al., 2019).

- **DAG** A directed acyclic graph (DAG) is a graph that is directed and has no cycles linking the other edges in computer science and mathematics. They are made up of edges and vertices (the spheres and the lines connecting them). They are guided since they all go in the same direction. They are acyclic because the vertices do not loop back on themselves, which means you can not go back to the same spot if you start at one point and follow the graph (Benčić and Žarko, 2018). This approach led to the creation of a new type of database structure that connects disparate types of data. Vertices and edges make up directed acyclic graphs. Unlike a blockchain, transactions are recorded as vertices that are stacked on top of each other. Transactions are added to the DAG by nodes, which function similarly to nodes on a blockchain. A node must finish a Proof-of-Work task before submitting a transaction. Every new transaction in a DAG, like blocks on a blockchain, must reference prior transactions in order to be accepted onto the network. A transaction is confirmed when it is referenced by another transaction. That transaction must be referenced by another transaction in order to be confirmed, and so on. The tip on which a new transaction will be built is determined by an algorithm. Tips with more confirmations have a higher chance of being chosen. DAGs are said to have a variety of advantages. Transaction speeds, for example, are rapid because processing is not constrained by block formation. Because there are no miners, there are no transaction

fees. There are, however, some substantial disadvantages. The fact that, unlike blockchains, they are not entirely decentralized is maybe the most crucial (Benčić and Žarko, 2018).

- **Sharding** Sharding is a distributed database-inspired approach that has become one of the most often used blockchain consensus scaling options. Sharding divides the state of the whole blockchain network into separate databases known as "shards", making it easier to administer than having all nodes maintain the entire network. The network processes these network shards in parallel, allowing for sequential work on multiple transactions at the same time (Chow et al., 2018). Furthermore, instead of retaining a full copy of the blockchain, each network node is assigned to a specific shard. Individual shards offer proofs to the mainchain and communicate with one another using cross-shard communication protocols to share addresses, balances, and general statuses (Dang et al., 2019). Along with Zilliqa (Team and Barrett, 2018), Tezos (Goodman, 2014), and Qtum (Dai et al., 2017), Ethereum 2.0 (Buterin et al., 2020) is one high-profile blockchain system that is researching the use of shards.

2.3.1.5 The infrastructure layer

In this section we discuss the Blockchain infrastructure layer for two enterprise BCs: Ethereum and Hyper-ledger Fabric.

- **Ethereum** By running client software such as Geth, Parity, or Pantheon on a user's PC, they can participate in the Ethereum Blockchain. There are two types of nodes in Ethereum: light nodes and full nodes. The light node executes the client software and keeps track of the Ethereum state. Additionally, the light node verifies transaction execution, whereas full nodes download the entire ledger to their local storage, participate in full consensus enforcement, validate signatures, transactions, and

Block formats, and check for double-spending. The Ethereum Virtual Machine (EVM), which is similar to Java Virtual Machines (JVMs) that run byte code, is operated by the Ethereum nodes. EVM, which operate as sandboxes, provides a smart contract execution environment. EVM is a Turing complete software, a stake-based virtual machine that manages smart contract internal state and processing. Virtualization is included in this layer (creation of virtual resources such as storage, network, servers etc.) (Buterin et al., 2014). Nodes are the most important part of this layer. A node is a device that connects to a blockchain network and is referred to as such. These nodes are decentralized and dispersed over a blockchain network.

- **Hyperledger** There are three types of nodes in the Hyperledger Fabric: 1) endorsers, 2) orderers, and 3) peer nodes. As mentioned in section 2.4.1.1, peer nodes host ledgers and chain code, commonly known as smart contracts. Fabric Software Development Kit (SDK) APIs allow users' applications and administrators to always communicate with peer nodes in order to access the chain code or distributed ledger. The Hyperledger Fabric manages a number of channels, each of which refers to a separate private sub-network with a number of peers (member). Each channel has its own ledger, which is kept by each peer on the network. Channels allow a limited number of applications and peers to communicate with Hyperledger. In the Hyperledger Fabric, transactions are processed in three stages (Androulaki et al., 2018). Because peer nodes are in charge of hosting the ledger and chaincode, applications and administrators must communicate with them. In Hyperledger Fabric, a node can host many ledgers. A peer node can sometimes only host a ledger and not the chaincode (it is rare, but possible). To update or access the node's ledger, most nodes have at least one chaincode installed. Multiple chaincode and ledgers can be hosted on a single node,

all of which are driven by channels. Applications and administrators (through admin applications) will always connect with peers via Fabric software development kit (SDK) APIs to access the chaincode or ledger. These APIs enable apps to carry out transactions on the blockchain network and get events relating to the process's confirmation. The query and update transactions are the two types of transactions. Consensus is not required for query transactions because the peer will return the result instantly from its local copy of the ledger. For update transactions, however, no one peer may update the ledger because other peers must agree before the ledger can be updated. Consensus is the process of achieving an agreement to update the ledger. Because a channel is a partition (a communication pathway) between a given application and a peer, it can communicate with that set of apps and peers (Androulaki et al., 2018). Hyperledger Fabric is a private-permissioned (consortium) and private-permissionless blockchain platform for businesses. There is no VM like EVM in Hyperledger Fabric (Ethereum). Smart contracts run on a peer node controlled by an organization and are mostly written in standard languages like Java, Node.js, and Go. Chaincode is deployed on network nodes, and smart contracts are mostly written in standard languages like Java, Node.js, and Go. Each blockchain instance has access to a secure Docker container that runs Chaincode. The peer nodes orchestrate the chaincodes, which operate as proxies, giving access to client applications via REST APIs or SDK (Cachin et al., 2016b).

2.3.2 Blockchain types

Depending on whether the membership is permissioned or not, blockchains can be classed as public, private, or consortium (Guegan, 2017). Any user can become a member of a public blockchain. There are no limitations on membership, and they are just pseudo-anonymous. Public blockchains

such as Bitcoin and Ethereum are suitable examples. On the other hand, private blockchains are those that are owned and managed by a company or organisation. In this type of blockchain, users who wish to join the network are required to obtain permission beforehand. The underlying mining model is a fundamental difference between these two categories that can be examined. Permissionless blockchains adopt the PoW model, in which the power of hashing is donated to establish trust. To reach a consensus, permissioned Blockchains do not require computational energy-based mining. Because all players are known, they use consensus algorithms like Practical Byzantine Fault Tolerance (PBFT) to obtain consensus without utilizing PoW mining, resulting in a block processing time that is substantially faster than blockchain's time permissionless and can be termed real-time. Compared to public blockchains, private and consortium blockchains process transactions faster. Hyper Ledger and Corda are two examples of permissioned blockchains. Besides, a group of organizations or a private community owns and operates consortium blockchains (Guegan, 2017).

2.4 Conclusion

Both Blockchain and IoT are emerging technologies that have recently attracted a great deal of attention from academic and business communities. Therefore, a thorough grasp of IoT and blockchain technology and their underlying components are required to demonstrate their viability for meeting the IoT's underlying security requirements. This chapter provides an overview of IoT and blockchain technology in order to accomplish this objective. The chapter provides a comprehensive overview of the Internet of Things, detailing its architecture and fundamental safety and security principles. In addition, the chapter examines IoT networks and protocol stacks to illustrate the diversity of IoT networks and communication protocols. In addition, an

overview of blockchain and distributed ledger technologies, including their architecture, primary characteristics, and technical working principles, is provided. The chapter also discussed the various blockchain types, including permissioned, permission-less, and consortium. Therefore, this chapter provides the necessary baseline knowledge of blockchain and IoT technologies for further chapters of this thesis.

Third Chapter

Literature review

3.1 Introduction

This chapter presents a literature review on blockchain and the decentralised authentication and access control in IoT providing a coherent and comprehensive picture of the current state-of-the-art efforts in this direction. The goal is to provide an evaluation methodology for acquiring a better knowledge of blockchain and distributed ledger technologies and their applicability for providing secure authentication and access management in IoT.

3.2 Access Control in IoT

Authentication and access control are critical in the IoT to ensure that users and devices can be trusted to be who they claim to be. Access control allows controlling access to data and resources within the IoT systems. Thus, access control can regulate which resources may be accessed and how they will be used and in which context, hence reducing the risk of unauthorised access. It ensures confidentiality in such a way as to ensure that information is only accessible to those authorised as well as the integrity of the data. Despite the fact that authentication and authorisation issues have been widely discussed in the literature, they are still at an early stage for IoT (Gupta et al., 2022a). Most of recent proposals have addressed the problem of access control using centralised approaches where a central entity

is responsible for managing the authorisation mechanisms, allowing or denying requests from external entities. However, traditional centralised access control models do not meet the requirements imposed by IoT scenarios, introducing lack of flexibility, scalability and usability in environments with billions of devices. Various access control mechanisms exist (Alnefaie et al., 2021). The following are the most well-known methods:

3.2.1 Access control List (ACL)

System resources are assigned permissions by use of the access-control list. An ACL describes which users or system processes are authorised access to objects, as well as what operations are permitted on such objects. Access control mechanisms are applied in the cloud in ACL, which allows for easier administration and tracking of actions, but is limited by a centralised infrastructure (Xiong et al., 2020) (YOSHII et al., 2020). As the number of IoT devices grows, the complexity of access regulations results in chaotic duty issues. ACL lacks granularity and scalability, and its centraliced architecture makes it vulnerable to a single point of failure (Qiu et al., 2020).

3.2.2 Discretionary access control (DAC)

This type of security access control known as discretionary access control, or DAC, gives or restricts object access based on an access policy that is set by an object's owner group or subjects (Al-Shaboti et al., 2018). This technique takes into account object administration based on the owner. In other words, the object's owner will set the access restrictions and policies. Access Control Lists (ACLs) or access control matrixes can be used to implement DAC (Langmead, 2022).

3.2.3 Mandatory access control (MAC)

This type of access control limits the ability of system users to get access to or conduct any operations on a particular object or target by implementing mandatory

access control (MAC) (Heydari et al., 2019). The categorization of objects and subjects is the foundation of this concept. It signifies that only subjects with a higher level than the item has access to it. The access decision will be decided by a central authority rather than the owner in this technique. MAC can be effective in circumstances where strong access control restrictions are required (Aftab et al., 2022).

3.2.4 Role-based access control (RBAC)

This method controls subject access based on their role in the system and the rules that define what kinds of access are permitted for subjects in specific roles (Bisma et al., 2020) (Jaikla et al., 2019). Because of the nature of this access control architecture, a small number of roles can represent a large number of users, making it easy to audit which users have which permissions and what permissions have been granted to each user (Gupta et al., 2022b).

3.2.5 Attribute-based access control (ABAC)

This approach is a logical access control model that restricts access to things by comparing the attributes of the subject, object, actions, and environment relevant to a request or a combination of these attributes to some established control rule or policy. ABAC is useful for access control with finer granularity (Vijayalakshmi and Jayalakshmi, 2022). Subject characteristics are associated with identifiers that specify the subject requesting access to an information asset, such as user roles, user IDs, group memberships, management level, and certifications, in the ABAC approach (Ameer et al., 2022). Object attributes distinguish the resources that the subject wishes to access, such as files, folders, and applications. The action attributes define the subject's action on the object. These actions include but are not limited to reading, writing, executing, and viewing. Environment attributes provide the circumstances in which access is requested, such as the time and location from which access is requested, the type of communication channel used, and so on.

However, in a widely distributed network environment, these solutions do not fully match the needs of access control mechanisms and inter-device communication. Traditional access control models have flaws that make them inappropriate for usage in an IoT environment, necessitating the implementation of Capability-Based Access Control (CapAC) systems. The ABAC paradigm directly associates attributes with subjects to address the role explosion problem in RBAC. The user attribute certificate is used to give access rights. The ABAC model becomes more complex as the number of IoT devices grows and policy administration becomes a significant concern (Vijayalakshmi and Jayalakshmi, 2022).

3.2.6 The Capability Access Control (CapAC) model

For large-scale IoT-based systems, a capability model is implemented. The capability list is associated with each subject in this model, and it determines the subject's access privileges to the target objects (Xu et al., 2018). Despite its widespread success, CapAC poses several issues regarding the propagation and revocation of access permissions (Sivaselvan et al., 2020).

3.3 Access Control Challenges in IoT

Current access control methods are centralised, requiring a centralised trust for them to work successfully. And different service providers maintain and manage the authentication data. The following are the key issues of implementing existing access control systems in the IoT environment:

3.3.1 Centralisation

One of the concerns is that centralised security solutions, such as Public Key Infrastructure (PKI), may cause scalability issues due to the hundreds of nodes connected. Furthermore, using a central authority comes with threats such as single-point failure (Dramé-Maigné et al., 2021).

Table 3.1: Access control methods for IoT

Study	AC methods	Objectives	Advantages	Limitations
(Xiong et al., 2020) (YOSHII et al., 2020)	ACL	Secure access in the cloud and fog nodes	Storage efficiency	Latency
(Al-Shaboti et al., 2018)	DAC	SDN-based framework for manufacturers and smart home IoT security	static and dynamic access control	Bandwidth, ARP response time
(Heydari et al., 2019)	MAC	Framework for indeterminacy-tolerant access control in IoT	indeterminacy handling	Theoretical no implementation
(Bisma et al., 2020) (Jaikla et al., 2019)	RBAC	Techniques for Ensuring Role-Based Access Control in IoT Devices	protection from masquerade attacks	Role Explosion, Security Risk Tolerance, confidentiality of data is not considered
(Ameer et al., 2022)	ABAC	Secured Smart-Home and IoT Access Control	two use-case scenarios, PoC, hypered model (HABAC and EGRBAC)	complex requirements
(Xu et al., 2018)	CapAC	A federated capability-based access control mechanism for IoT	scalable, lightweight and fine-grained access control solution	requires registration, dissemination, and revocation of the entry

3.3.2 Third-party

Relying on a third party to offer data backups is a common solution. Using a third party to gather and analyse such data increases the risk of being exposed (Al-Turjman et al., 2022).

3.3.3 Privacy

The difficulty with relying on third parties to manage trust is that the cloud server will gain personal information, resulting in major data leakage concerns.

Furthermore, individuals have little to no control over the personal data collected from them (Al-Turjman et al., 2022).

3.3.4 Resource limitation

It has been demonstrated that these devices will always be resource constrained devices with low resources, preventing them from implementing advanced security solutions. When encryption is used for authentication, for example, some complicated encryption algorithms might cause issues such as low computing performance, increased hardware power consumption, and so on (Thakor et al., 2021).

3.3.5 Heterogeneity

IoT infrastructure is distributed and consists of several heterogeneous, networked devices that use various underlying technologies and come from various disciplines. These systems have separate underlying authentication and authorisation procedures for each domain, making heterogeneity a major impediment to establishing a scalable, robust, and secure IoT environment (Al-Turjman et al., 2022).

3.3.6 Scalability

The number of linked devices is rapidly expanding, which increases their management effort. A decentralised and distributed access control mechanism must enable scalability to accommodate the ever-growing number of homogenous and heterogeneous IoT devices (Gupta and Quamara, 2020b).

3.4 Authentication in IoT

IoT authentication is a methodology for establishing trust in IoT technologies and systems' identities to safeguard data and manage access as it travels over an insecure network. Authentication is used in the IoT to identify users, devices, and applications, as well as to restrict access to only authorised users and non-manipulated

devices or services (Liyanage et al., 2020a). Additionally, authentication helps prevent attackers from posing as IoT devices to gain access to data on servers, such as recorded sensor reading, medical health data, and other potentially sensitive data. There are numerous approaches for achieving robust authentication in IoT. However, when it comes to deploying authentication and access control mechanisms in an IoT environment, many aspects need to be considered. The common needs for all IoT applications are high flexibility, scalability, collaboration with multiple stakeholders, and the necessity for lightweight security procedures (Mehta and Patel, 2020). The following are four primary authentication methods.

3.4.1 Knowledge-based authentication

This is an authentication method that attempts to prove the identity of someone using IoT apps. As the name implies, these elements are based on the user's knowledge, such as their ID and password. It needs the individual to know private information in order to establish that the person supplying this information is the identity owner. There are two different kinds of knowledge-based authentication, namely static and dynamic. Static knowledge-based authentication is based on a set of agreed-upon held secrets. In contrast, dynamic knowledge-based authentication is based on questions made up of an extensive pool of personal information (Kim and Park, 2019).

3.4.2 Possession-based authentication

This element is based on the user's ownership of credentials, RFID, or other identifiers available to the principal user (Sharma and Agrawal, 2021).

3.4.3 Inherence-based authentication

These are biometric traits such as fingerprints, iris data, etc (Sulaiman et al., 2021).

Table 3.2: IoT Authentication Mechanisms

Study	Auth method	Objectives	Advantages	Limitations
(Kim and Park, 2019)	Knowledge-based authentication	An authentication model for intelligent closed circuit television in mobile personal computing	safer CCTV surveillance environment in the future implementation of smart cities	High overhead due consuming more resources
(Sharma and Agrawal, 2021)	Possession-based authentication	Personal authentication based on vascular pattern using finger vein biometric	user-friendly, easy to enrol, contactless, highly hygienic and has universal acceptance	require a high database and storage to store enrolled template
(Sulaiman et al., 2021)	Inherence-based authentication	Online Voting System using Face Recognition for Campus Election	remotely reachable, convenient, and reduces the voting time	require additional devices and technical complexities
(Alsahlani and Popa, 2021)	Multi factor authentication	multi-factor authentication and authorization scheme for real-time data access in IoT cloud-based environment	can prevent several attacks, including sensor and user impersonation as well as man-in-the-middle, reply, and traceability attacks.	centralisation, fuzzy extractors cannot be reused multiple times for the same biometric due to their low-key entropy

3.4.4 Multi factor authentication

The previous procedures are combined in this method. Authentication using MFA necessitates the submission of two or more verification factors by an individual in order to obtain access to a particular resource (Alsahlani and Popa, 2021).

3.5 Encryption and authentication services

Two of the most critical security services provided in any network system are authentication and encryption. Generally, public-key cryptography, one of the most well-known security frameworks, can be used to provide these services. The

entities must have private and public information to use public-key cryptography techniques. They will need a system in place to generate, revoke, manage, distribute, use, and store the generated keys or information (Al-Naji and Zagrouba, 2020). The public key cryptography and its applicability in today's applications are first explored in this section. In the following section, an overview of public key management approaches and the issues associated with them is presented. After that, there is a discussion and comparison of the ways in which blockchains can be utilised to address these issues, and an overview of several management techniques that are based on blockchain technology.

3.5.1 Public Key Cryptography and its Services

Asymmetric cryptography, often known as key cryptography, is a cryptographic technique that employs a pair of keys: public keys that are circulated across the system and private keys that are kept secret. Diffie and Hellman first proposed it in 1975 (Whitfield and Hellman, 1976), and it is still commonly used today. The essential concept is to employ one of the keys to perform one task (encryption or signature) and the other to perform the opposite duty (decryption or validation). In this approach, every entity can use the public key of a specific user to verify a message sent by that user. Before sending the reply message, it can also be encrypted. Only that unique user may sign or decrypt the communication with its private key. Many security services, such as entity authentication and secrecy, can be provided through public-key cryptography. The entity authentication service can be given through the signature/verification technique. Everyone can verify/authenticate an entity by confirming the signature with the entity's public key after it sends a message signed with its private key. Because the private key is kept private, no one other than the entity or someone with access to the private key can sign the communication. The public keys, on the other hand, are used for verification. As a result, anyone with access to the user's public information can verify and authenticate the user. On the other hand, encryption/decryption, a comparable

operation, can be used to provide confidentiality services. The sender encrypts the message using the receiver's public key. The receiver, using his private key, decrypts the message. The data can only be decrypted and understood by the receiver or someone who possesses the receiver's private key. As a result, confidentiality is assured (Hanaoka et al., 2022).

3.5.2 Cryptography and its importance for the current applications

Entity authentication and message confidentiality are vital in practically all current network applications. A smart healthcare environment is an excellent example of how important these services are. The system must encrypt sent data to protect patients' privacy from outsiders. Furthermore, it is critical to identify the correct doctor, hospital, and pharmacy and secure their data access. Many techniques have been proposed to produce the system's private/public keys, such as RSA (Rivest et al., 1978), elliptic curve (Miller, 1985). The scope of this study does not allow for a discussion of these algorithms. However, these are complicated and require infrastructure to generate and manage public/private keys. The certificate authorities (CA), the web of trust (WoT), and the entity-based cryptosystem were introduced to create, manage, utilise, store, and distribute keys (Khalaf and Kadi, 2017). The following section discusses traditional and blockchain-based key management systems, such as CA and the web of trust. We will address the entity-based cryptosystem in a later paragraph, a contemporary development that extends the CA methods to better use public-key cryptography.

3.5.3 Key Management by the Public Key Infrastructure (PKI)

One technique to provide key management for public-key cryptography is through the public key infrastructure (PKI). PKI has traditionally been achieved in two ways: centralized by a Certificate Authority (CA) or decentralized WoT. The most widely used technique is the CA-based PKI, specified in the X.509 standard (Albogami

et al., 2021). The CA is a third-party entity that all members of the system trust in this technique. The CA provides "certificates," which connect each user to a public key and authenticate them. A signed certificate that connects a user to their public key verifies that it belongs to that user. Web of Trust (WoT), introduced by Phil Zimmerman in 1992 (Zimmermann, n.d.), is the other traditional technique. This method employs a decentralised approach, in which the keys are generated locally and confirmed by at least one other trustworthy person in the system (Chenchev et al., 2021). Traditional PKI Systems have several challenges as both standard procedures face several difficulties, which will be explored in this section. Three significant issues confront the CA-based PKI: a trusted third party, a single point of failure, and expense. Users of the systems must trust the CA to generate and manage their public keys, posing significant security risks if the CA is hacked. Because the entire system collapses if the CA fails, this architecture has a single point of failure. Moreover, the administration of public keys by a single centraliced CA can be costly and inefficient, especially in today's massively dispersed applications involving many users (ITU, n.d.). On the other side, Trustworthiness must be established by IoT-based PKI signers. Users can only join the network if they have the trust of a "trusted" party. In other words, new members must first establish trust with existing members before joining the network. This makes it challenging for new members to join the network (ITU, n.d.). Furthermore, neither the CA-based nor the IoT-based PKI can enable identity retention. A user can spoof the identity or public key of another user who has already registered. Some solutions have been proposed to this problem; however, they are generally log-based, which could be quite complex, especially given the global dispersal of users.

3.6 Identity management in IoT

3.6.1 Digital identity

In the Internet era, the digital identity remains the keystone of online services and upon which security mechanisms (i.e., authentication, authorization, secure exchanges) and protocols are built. As defined by the International Telecommunication Union (ITU), an identity refers to a set of information used for uniquely identifying an entity in a given context (ITU, n.d.) whereas an Identity Management System (IdMS) refers to the management of identity information through a set of operations, including registering, updating, revoking and looking-up digital identities. However, existing identity management systems in the context of the Internet could not be directly transplanted to IoT environments due to some native IoT characteristics like scalability, interoperability, mobility, limited computational and storage resources. Traditional centralized identity management systems, relying on the so-called trusted third parties, raise many privacy concerns (Rey, 2021). The proliferation of online identity providers also leads to fragmented identities scattered all over the Internet, which makes us be overwhelmed by multiple accounts and expose personal information retained by identity providers to vulnerabilities and data breaches.

3.6.2 Identity Management System (IdMS)

IdMS are in charge of handling user identification data, which includes identifiers (UserID, Email, URL), credentials (Certificates, Tokens, Biometrics, etc.), and characteristics (Roles, Positions, Privileges, ...) (ITU, n.d.). IdMS has long been recognized as the cornerstone for accessing Internet services and resources since the dawn of information technology. IdMS have progressed from isolated to centralized, and finally to federated models during the previous three decades. Identity providers have played a key role as dependent parties (service providers) in the Isolated IdM paradigm, providing subjects (users) with access to Internet

services and resources managed by a single security domain. When subjects intend to use Internet services, they must first register with service providers and receive digital identities based on their security domain credentials (Nur and Wang, 2021). Nonetheless, identity bloating results from the rapid proliferation of internet services in numerous security fields. Following the isolated IdM model makes managing many digital identities (e.g., memorizing their matching passwords) impractical for humans. To address this issue, the centralised IdM approach tries to decouple identity management from a service offering, allowing multiple service providers to use the same identity provider (Nur and Wang, 2021). Users still need access to distributed services governed by different centralised IdM systems and security domains, even if the centralised IdM architecture decreases the number of user IDs.

3.6.3 Federated IdM

The federated identity management paradigm establishes trust connections amongst identity providers so that consumers in one security domain can access services from another. The federated identity allows information about users to be shared between security domains within the federation. This means that services supplied by another domain in the same federation can be accessed using credentials provided by its domain, regardless of whose identity is validated in one domain. However, phishing attempts could spread due to the access of many unauthenticated third-party service providers to the detached identity providers. The emergence of centralized and federated identity management systems has reduced the complexity of managing multiple identities from various security domains. However, when the number of apps per domain grows, all agreements, protocols, standards, and processes (such as authentication and authorization) across these domains become highly complicated, compromising identity usability (Aldosary and Alqahtani, 2021). Furthermore, while centralized and federated IdM systems are created with service providers in mind, they are inflexible due to a lack of user

concern. To improve user experiences while also ensuring security and privacy, user-centric identity management approaches have been developed.

3.6.4 User-centric identity management

An efficient user-centric consent management system was introduced in the study that was presented in (Marillonnet et al., 2021). Users of this system would be able to access online services offered not only by the Territorial Collectivities and Public Administration (TCPA), but also by third parties that have been authorised by the user. OpenID (Recordon and Reed, 2006) is a decentralised, user-centric identity system for web applications. To authenticate users, it introduces ID token (JSON Web Token) based on the OAuth 2.0 authorisation protocol. Thanks to the decentralised framework, the identity providers are more resilient to DDoS (Distributed Denial of Service) assaults. However, identity providers who use the OpenID standard may see all linked web login information, making cross-site tracking easier. Furthermore, OpenID's URL-based identifiers frequently endanger users' privacy. Although user-centric identity management systems provide better solutions for managing subjects' and service providers' identities, the trust assumption that users must place their whole trust in third-party identity providers persists. Users must continue to rely on "trusted third parties" Identity providers to access services across domains, although these identity providers have access to all transactions between users and service providers. Current identity management solutions need users to consider all IoT entities and coordinate different application domains to join the IdMS, limiting scalability and making it more difficult to establish an interoperable system in such a diverse environment. Although some initiatives, such as OpenID, are expandable to some level because of their decentralized nature, the IoT IdMS still need a robust, extensible system to manage all entities in the IoT context (Pöhn and Hommel, 2020). Furthermore, in the context of omnipresent IoT devices or services, the mobile IdMS is crucial. The mobile IdMS should ensure that user identities are accessible no matter where they are or how they move.

Finally, the majority of efforts take security and privacy into account. However, as previously stated, they are based on the idea that all users, including subjects and dependent parties, should trust their IdPs because the corresponding IdPs are invariably involved in every transaction, compromising user privacy. Even though many systems use a user-centric identity management paradigm, they can still not meet the IoT identity management needs (Pöhn and Hommel, 2020).

3.6.5 Challenges of IdMS in IoT

Despite multiple promising principles and methodologies presented during the Internet era, important difficulties such as access controls, privacy, trust, and performance remain unaddressed when it comes to developing successful IdMS for IoT.

3.6.5.1 Access controls centralisation

The purpose of building identity systems in the IoT is to facilitate communications while also ensuring that the authorisation procedure for devices and resources is appropriately regulated. Due to the rapid growth of roles and policies, many classic accesses control models, such as Access Control Lists (ACLs) and Role-based Access Control (RAC) models (Houhamdi and Athamena, 2020), that were built for centralised systems, have become obsolete with the onset of the IoT era. In addition, more and more elements and parameters, such as time and location, should be considered when creating access control solutions. Despite the fact that the Attribute-based Access Control (ABAC) model tries to solve this problem, the presence of centralised identity providers in the ABAC model still poses a scaling challenge. Existing solutions have a similar disadvantage: they rely on centralised administrative parties (such as administrators or identity providers) to give access rights, responsibilities, and attributes, making them unsuitable for scalable decentralised IoT systems. Because of its versatility, the Capability-based Access Control (CAC) approach has gotten a lot of attention (Houhamdi and

Athamena, 2020). However, the basis remains the same, customers who seek services must rely on third-party authentication, such as that provided by identity providers or certificate authorities. This appears to be problematic in trustless IoT contexts because each topic could form users without the approval of other third parties (Dramé-Maigné et al., 2021).

3.6.5.2 Privacy

The term "privacy preservation" refers to safeguarding users' sensitive data, such as their identity, location, mobility traces, and habits, from third parties. In terms of users' privacy, there are two elements to consider. Firstly, the protection of personal information from identity providers, secondly, securing of sensitive application data from service providers (Wang and Meng, 2021). Many academic articles and IT industry experts have advocated that sensitive application data be preserved rather than identification information kept in identity providers. In most cases, identity providers and service providers are linked, and to verify users; they need some personal information. Users can, for example, disable the location service to protect their location information from map service providers, but they overlook the risk of their personal identification information being leaked by identity providers with security flaws. Even though these recommended methods in partially alleviate the privacy challenges, their data is still available to identity providers. Due to the existence of centralised identity providers, privacy protection before blockchains was insufficient. Service accessors and owners must have complete faith in their identity suppliers. Put another way, centralised identity providers keep track of service accessors and owners and can track all transactions between them (Thilakarathne, 2020). Blockchain technology unites all user identities that are spread across several identity providers and are under the control of the users. Users determine to who their sensitive personal information can be revealed (from the user's perspective), rather than trusting identity providers to manage their personal information, which affects the design of identity solutions. Furthermore, including zero-knowledge proofs in blockchains allows for the selective publication

of sensitive personal data (Patil et al., 2021).

3.6.5.3 Third party

Users and service providers trust and rely on the same identity provider within the same security domain, acknowledging that their personal information will not be compromised or misused by the identity provider or third parties. In many circumstances, identity providers are vulnerable to attacks that allow for the theft of personal information repositories by malicious attackers. The implicit faith in identity providers is called into question, rendering the centralised identity paradigm outmoded in terms of privacy preservation and long-term viability in the IoT era, which aims to link everything. Although firms (such as Google or Facebook) strive to create universal identity providers for all cyber-users and service providers, the growing number of online service providers and identity providers isolates and fragments digital identities across the Internet. Furthermore, the cost of communication and mutual authentications between different security domains is quickly increasing. Different identity providers must also negotiate to create trust relationships with federated identities across security domains (Thilakarathne, 2020). Without a doubt, blockchain-based identity management solutions remove unwanted information exposure to third parties and offer several beneficial properties such as immutability, neutrality, and secure timestamping that can be utilized to establish trust relationships (Patil et al., 2021). For instance, a blockchain-based privacy-preserving and rewarding private data sharing system (BPRPDS) for the Internet of Things (IoT) has been proposed in (Li et al., 2022a). The work presented addresses the prevention of behaviour profile construction and non-frameability of BPRPDS using the deniable ring signature and Monero. Then, they use smart contracts to limit access to multi-sharing by using licencing technologies. However, these decentralised or distributed approaches confront numerous challenges in developing a reputation system or feedback mechanism for aggregating trust relationships amongst all parties, including subjects and service providers. The applications should be redesigned in a decentralised manner and

self-contained (Dapp).

3.7 The need for blockchain in IoT authentication

The lack of privacy and security of sensitive data that is sent when linked devices speak with each other or with the cloud, according to (Dramé-Maigné et al., 2021), is a fundamental barrier in the centralised environment of IoT. Due to a lack of computational power and energy to run encryption methods. The introduction of Blockchain has opened the door for further investigation into its possibility of meeting some or all of the security requirements for IoT systems outlined in section 3.7.1. Decentralisation of trust is quickly becoming a dominating trend, opening up possibilities for decentralized and autonomous authentication and authorization management. Each device can be considered a blockchain user when using blockchain technology to safeguard IoT devices. As a result of implementing a consensus mechanism, IoT items do not need to trust each other, allowing nodes connected to the Blockchain to operate securely. To conduct transactions, blockchain users use a pseudonym (address). Because blockchain is an immutable transaction log, it may be used to track millions of IoT devices and offer highly secure communications and coordination amongst them. Furthermore, attaining consensus among peers is a fundamental requirement in a distributed setting like blockchain, detecting assaults and preventing further harm. DDoS attacks, device spoofing, impersonation, inserting malicious code, side-channel attacks, and other threats would be eliminated if Blockchain was implemented in the IoT. Another reason is that when we utilise blockchain to assure trust in IoT data, we get much-needed security services like confidentiality, accountability, integrity, and availability. Authentication of M2M communication between IoT devices is possible. Malicious nodes can be recognised and segregated (Dai et al., 2019).

3.7.1 Blockchain characteristics

Several features and advantages of blockchain technology which can benefit IoT are discussed below.

3.7.1.1 Integrity

Certainly, the data has not been tampered with, except by those who have the authority to do so. In the context of the Blockchain, integrity ensures that transactions are unchangeable. Cryptographic procedures are frequently employed to verify integrity (Wang and Zhang, 2019).

3.7.1.2 Availability

It assures that users of a system may access it whenever they need it. In other words, the service is always available when a legitimate user requests it, which necessitates the communication infrastructure and database. Even if some nodes go offline, blockchain systems cannot jeopardize the network's availability or security. On the other hand, traditional databases rely on one or more servers and are more vulnerable to cyber-attacks and technological failure. Furthermore, Blockchain's peer-to-peer architecture gives all network participants fair validation rights to evaluate the accuracy of IoT data and ensure immutability (Abdelmaboud et al., 2022).

3.7.1.3 Confidentiality

It ensures that unauthorized individuals will not obtain information. Only those with the appropriate rights and privileges will have access to the data, whether in processing or transit. The Blockchain employs pseudo-anonymization mechanisms, like hash functions, which obscure users' identities to ensure this concept (Fan et al., 2022).

3.7.1.4 Authentication, authorisation, and auditing

This seeks to verify the identity of who performs a specific function in a system, check what rights that user owns, and store usage information for that user. The structure of the Blockchain ensures these three functions since only users who have the private keys can perform transactions, and all transactions are public and auditable (Li et al., 2022b).

3.7.1.5 Immutability

On the Blockchain, transaction data is immutable over time. Technically, after being validated by the Blockchain network, transactions are timestamped and then inserted into a Block that is cryptographically protected by a hashing process. Hash mechanisms connect blocks and create a sequential chain. The hash value of metadata from the preceding Block is always stored in one field of a new block's header, making the chain highly immutable. After it has been validated and stored in the Blockchain, the block data cannot be modified, edited, or erased in this manner (Al-Naji and Zagrouba, 2020). The cryptographic link between successive blocks thwarts any attempts to edit or modify transactions. Even if a transaction changes, it will be immediately identifiable.

3.7.1.6 Nonrepudiation

It ensures that a person cannot deny a system's operation. Nonrepudiation establishes that a user took a specified action, such as giving money, authorising a transaction, or sending a message. Because all transactions are signed, a user can't say he didn't do them (Al-Naji and Zagrouba, 2020).

3.7.1.7 Decentralisation

With its decentralised character, Blockchain is a viable solution for effectively tackling bottleneck and one-point failure concerns in IoT networks by removing the requirement for a trusted third party. The functionality of the BCIoT network is

unaffected by the failure of a Blockchain node. The data on a blockchain is often stored in numerous nodes on a peer-to-peer network, and the system is highly resistant to technological failures and vicious attacks (Wang and Zhang, 2019).

3.7.1.8 Smart contract

Blockchain in conjunction with smart contract technology reduces the need for central servers to ensure that transaction parties are treated fairly. Every linked entity on the blockchain network will have a copy, giving them equal power over all contract processes. Furthermore, Blockchain-based IoT can provide users with trustworthy access control using Blockchain-enabled smart contracts, automatically authorizing all IoT device functions. Furthermore, smart contract services provide users with data provenance. This gives data owners control over how their data is exchanged on the Blockchain. Users can specify access rules for self-executing smart contracts on the Blockchain, ensuring personal data protection and ownership. With smart contract-based authorization, malicious access may be confirmed and disabled (Lone and Naaz, 2021).

3.7.1.9 Enhanced Security

In numerous ways, Blockchain is more dependable and secure than traditional record-keeping methods. Prior to being documented by the network participants, transactions must be agreed upon. When a transaction is approved, it is encrypted and connected to the initial transaction. Furthermore, rather than storing information on a single server, information is distributed throughout a network of computers, preventing hackers from gaining access to transaction data. The use of PKI (private/public key infrastructure) is the most important security aspect in Blockchains. Blockchain systems employ asymmetrical cryptography to safeguard transactions between participants. These keys are produced using random numbers and strings, making it impossible to calculate the private key from the public key. This prevents future assaults on Blockchain documents, minimizes data leakage issues, and improves the security of a Blockchain network. Furthermore, Blockchain

has the potential to change how personal information is shared to avoid fraud and criminal actions in any industry where sensitive data from several applications is crucial, such as financial services, government, and healthcare (Miraz and Ali, 2020).

3.7.1.10 Transparency

Access control utilising Blockchain may effectively solve data leaks and give traceability by achieving transparency. Because all network users have access to the transaction histories in Blockchain, they are more transparent. In contrast to individual copies in a traditional network, Blockchain is a distributed network where all members share the same documents. This shared document can only be changed by agreement, implying that everyone must agree. Put another way, the identical copy of Blockchain data is distributed throughout a large network for public verification. As a result, all Blockchain users have equal access to the network, allowing them to link, verify, and trace transaction activities. To change a single transaction record, all subsequent records would have to be changed, necessitating network-wide collusion. As a result, data on a Blockchain network is more accurate, reliable, and transparent than data on a traditional network. By decreasing the risk of illegal data changes, such transparency also helps to safeguard the credibility of Blockchain-based systems (Hellani et al., 2021).

3.8 The rise of blockchain-based identity solutions

With cyber security concerns becoming more prevalent in IoT, blockchain technology is gaining traction as a viable option for developing IoT security solutions in decentralised and trustless environments. Blockchain-based IdM systems are garnering much attention in academic research to provide new solutions for digital identities (Liu et al., 2020). For instance, the work presented in (Al-Bassam, 2017) provided a blockchain-based public key infrastructure (PKI) and implements it using Ethereum smart contracts. It outlines several identity-related actions in his

work, including adding characteristics, signing attributes, and revoking signatures. More crucially, they evaluate the cost of various Ethereum platform processes. Another interesting work was presented in (Liu et al., 2017) uses Ethereum smart contracts to create an identity management system that binds the public key and the user's entity information. They also redefine the token to meet their suggested reputation model, such that it reflects the reputation of users and the identity management aspect. The authors in (Axon, 2015) examined privacy needs and proposes a blockchain-based PKI with privacy awareness when creating decentralized PKI systems. They introduce the notion of a neighbourhood group to improve the performance of privacy preservation in addition to a set of activities such as registration, revocation, and recovery.

Maintaining identity privacy and providing identity anonymity solutions are presented in (Augot et al., 2017), which alter the Bitcoin stack to create an identity management system and provide a zero-knowledge proof dubbed the Brands selective disclosure method to ensure identity anonymity while doing so. In a permissioned blockchain system, the work in (Hardjono and Pentland, 2019) proposes ChainAnchor, a blockchain-based privacy-preserving identification solution that uses zero-knowledge proof. Verified nodes in ChainAnchor can write and process transactions, while others can only read and verify them. To provide privacy protection services to consumers, all confirmed nodes are constructed on tamper-resistant hardware and form the privacy preservation layer. Another work is presented in (Halpin, 2017) which proposed NEXTLEAP, a federated identification system that uses blind signatures to maintain anonymity. Furthermore, they construct a more secure messaging application using authentication services supplied by their identity solution. Similarly, the authors of (Azouvi et al., 2017) also suggest blind signatures as a privacy-preserving identity solution. They create a threat model, do a security study, and then deploy their solution in Ethereum.

In addition, various startups and IT players, such as Uport (Naik and Jenkins, 2020), Showcard (ShoCard, 2020), Bitnation (Bitnation, 2018), Civic (McCabe and Kennedy, 2014), Jolocom (Decentralized, 2021), Sovrin (Khovratovich and

Law, 2017), and ID2020 (Alliance, 2018), are concentrating on creating identity systems. Uport, a key component of the Consensys Ethereum ecosystem, intends to tackle the digital identity problem by developing decentralised applications. It primarily employs smart contracts to create digital identity models, and it provides identity dependability and usability through a set of actions (i.e., keys revocation and identities recovery) (Azouvi et al., 2017). Sovrin proposes a different approach, offering a full-stack solution for managing identities, from the distributed ledger to devices. It acts as a global public utility that provides an identity layer to every entity on the Internet and serves as a permanent, private, and trustworthy identity provider (Khovratovich and Law, 2017). Sovrin creates a public permissioned blockchain in a peer-to-peer network with nodes divided into authenticated validator nodes and observer nodes to ensure high performance and scalability. More crucially, the sovrin token is incorporated into their system to generate incentives for their transactions. In general, a blockchain-based identity is a self-sovereign identity, which refers to a method of shifting access control rights and identity management from traditional identity providers to the edge, where identity owners have power. In other words, only the owners of identities have the authority to dispose of them, preventing attacks from malevolent third-party identity suppliers (Khovratovich and Law, 2017). Although a self-sovereign blockchain identity provider could eliminate superfluous third parties, the most difficult element is generating trust in a trustless IoT context rather than eliminating trust requirements on these third-party authorities. To put it another way, even if people could verify one other's true identities due to privacy issues, they still don't trust (or know) each other.

3.8.1 Blockchain Distributed PKI

Traditional PKI's certificate authority (CA) is particularly vulnerable and prone to compromise and operational failure because to its central position in the system. Keeping track of CAs and revocation lists can be time-consuming, especially in big,

disjointed systems. Although log-based PKIs have been presented as a solution, their effectiveness has not been proven. A generic concept and solution for decentralised and dynamic PKI built on blockchains and webs of trust was offered by the authors of (Toorani and Gehrmann, 2021), which does away with traditional CAs and digital certificates entirely. Everything is recorded on the blockchain in their model. As part of a consensus-based system, the registration, revocation, and updating of public keys are done by a group of entities that are already members of the system. Auditing and revocation procedures can be initiated by any system node, as long as the node is part of it. No longer are revocation lists necessary because any node can easily validate the public keys through witnesses. Similarly, the work presented in (Shi et al., 2022) suggested a novel distributed authentication model as a secure approach for supporting public-key cryptography. The proposed model creates a decentralised public key infrastructure by combining blockchain smart contracts and optimised zero-knowledge proof-verifiable presentations via the DID project, which enables the management of public-key certificates via blockchain and ensures the authenticity and availability of public keys in decentralised infrastructure. The proposed approach fundamentally resolves existing schemes' security and feasibility difficulties while also providing a more scalable alternative for verifying data sources. Despite the fact that the paper did not include any implementation-related aspects, it did introduce the possibility of blockchain-based PKI, which was later implemented in many other works, as will be discussed next.

3.8.2 Self-sovereign Identity (SSI)

Rather than using servers like those found in traditional centralised identity management systems (IdM), users in SSI utilise Dapps to access their wallets and regulate who has access to their sensitive data. As a result, system users are empowered and can govern their identity and credentials (Shi et al., 2022). Organisation resources are restricted to authorised individuals in traditional identity management systems. Traditional identity management methods include Open

Authentication (OAuth) and OpenID Connect. There would be an identity supplier, a service provider, and users in an effective identity system. Users and service providers can rely on identity providers for authentication, registration, and other identity-related services. An identity provider can be a service not affiliated with the service providers. The identity provider, or validation and authentication of a user's claimed identity, is usually requested by the service provider (Preukschat and Reed, 2021). An example of the blockchain-based identity management solutions that manifest a digital identity without relying on a centralised server are Sovrin (Khovratovich and Law, 2017), uPort (Naik and Jenkins, 2020). The storing of identification information is handled by peer nodes rather than a central server, a major aspect of blockchain-based identity management systems. They should also ensure that authentication, trust, and privacy are maintained. In addition to an SSI system, several suggested blockchain-based identification systems keep users anonymous and rely on an attribute reputation model. The identity provider, or validation and authentication of a user's claimed identity, is usually requested by the service provider. The storing of identification information is handled by peer nodes rather than a central server, a major aspect of blockchain-based identity management systems. They should also ensure that authentication, trust, and privacy are maintained (Khovratovich and Law, 2017). In addition to an SSI system, several suggested blockchain-based identification systems keep users anonymous and rely on an attribute reputation model. A lost or forgotten password in a password-based system can be quickly reset. Losing the private key in blockchain-based SSI systems, on the other hand, results in asset loss (Laatikainen et al., 2021).

3.9 The need for Blockchain in IoT authentication and access control

Using blockchain for authentication in IoT will provide a viable solution. Because blockchain is a distributed ledger with a list of related data records or blocks, each block contains a collection of new data records or transactions and the preceding block's hash value and a timestamp that verifies the transactions at the moment of the block's creation, which makes record change impossible because they rely on previous records. The fact that data can't be changed because it's replicated and stored in a distributed and reliant manner characterises blockchain (Ourad et al., 2018a). In addition, blockchain is a highly transparent access control solution that offers end-to-end decentralised security while lowering the chance of human mistakes. Thus, it provides strong protection against hacker attacks and is critical for access control systems. Blockchain can provide a decentralised solution that uses a consensus process to ensure the integrity of authentication data. To provide secure distributed transactions in a trustless environment, no third-party intermediary is required, but trust is built through a public ledger recorded in a decentralised manner. Due to its decentralised infrastructure and anonymity maintenance, the blockchain protocol can bring some evolutionary modifications to established IT security methods (Ourad et al., 2018a). Traditional security and access control systems necessitate using a centralised trusted entity, which degrades end-to-end security features. Applying a centralised access control mechanism to an ever-increasing number of IoT devices might make trust management more difficult and limit the system's scalability. The construction of a fair authorisation framework to meet the IoT as mentioned above access control concerns can be aided by blockchain.

3.9.1 Blockchain-based authentication and authorisation

Based on our findings, the existing research on blockchain-based authentication and access control systems can be summaraised as follow. Authentication methods can be classified based on the type of authentication and the application environment. Knowledge-based, possession-based, biometric-based, and multi-factor authentication is the main authentication types used in various contexts such as cellular networks and telecommunication, IoT devices and smart cities, healthcare and medical data records, cloud computing, and resource sharing as previously mentioned. Some techniques are general-purpose methods that can be used in any environment, regardless of the application (Esposito et al., 2021). There are three categories of blockchain access control mechanisms based on the technology employed, the application, and how the blockchain network is used. The three fundamental categories of access control techniques are referred to as ABAC, RBAC, and ACL-based approaches such as DAC. Blockchain technology is used in two different ways in these strategies. Although some employed blockchain as a secure, immutable, and distributed database for access rules and policies, others used blockchain and smart contracts to control the entire access management process (Alilwit, 2020). Such solutions, like authentication mechanisms, can be general-purpose or used in specific domain (i.e., cellular network and telecommunication, IoT devices and smart cities, healthcare and medical data records, cloud computing, and resource sharing).

3.9.2 Blockchain-based authentication methods

In this section, we will describe and talk about the many existing authentication mechanisms that make use of blockchain or smart contracts. In summary, table 3.4, shows the taxonomy-based classification of the approaches. The authors of (Zhang et al., 2017) developed a general-purpose architecture that keeps a user's identity in the blockchain and uses a smart contract to manage different permissions for different websites based on the user's relevant data. This method has four primary

actors (i.e., users, websites, blockchain, and off-chain storage). A user's identity is stored on the blockchain, but his encrypted personal data is stored off-chain. A smart contract will be tied to the user's identity in the blockchain to prepare different web pages with different and related user data. When a user submits a login request to a website, the service provider verifies the user's identification and extracts the user's data from off-chain storage using the smart contract's rules. In addition, the work presented in (Deep et al., 2019) suggested a cloud-centric database authentication technique for application in cloud and healthcare environments. Both insiders and outsiders can use this strategy. It first verifies the user's credentials and validates the blockchain node specifications. If the cloud database does not have the user's credentials, they can either try again or create a new account. For storing credentials, the proposed solution uses blockchain as a distributed database. SAMS is another authentication method introduced in (Kim and Jeong, 2018) in the cloud context. This technique employs a master node as a coordinator, which is in charge of the system's security. The master node first creates and saves its block on the blockchain for user authentication. When a new client node wishes to connect, he produces a new block and sends his data and the newly formed block to the master node. The master node produces a block with the information received from the client and verifies its authenticity. If the client block and the server block are the same, the connection will be made. Blockchain is employed as an immutable database for credentials in this manner. Another work was presented in (Huh and Seo, 2019), which introduced a fingerprint authentication and verification method for mobile phones utilizing blockchain to create an automatic door locking system. A person authenticates himself via fingerprint recognition on a mobile device. The hash value of a user's fingerprint will be recorded to the blockchain to prevent forgery, tampering, or leakage. This solution requires the cell phone to use a PoW consensus mechanism, which would be extremely resourced intensive for these devices.

Additionally, the authors of (Widick et al., 2019) provided another blockchain-based authentication and authorization system for controlling user access to IoT

device resources. Two smart contracts make up this strategy. One is responsible for digital certificates and operations, while the other is in charge of access control. These contracts are both managed by an agent node. The Ethereum blockchain is used in this system to offer a tamper-evident, auditable log of all steps and to decentralize specific functions (e.g., evidence review). For the IoT environment, the authors of (Hammi et al., 2018) developed a decentralized blockchain-based authentication mechanism dubbed bubbles of trust based on the user's ID and token. This paper's key issues are data integrity and availability. This technique uses Ethereum's security benefits to construct safe virtual zones (bubbles) where items may identify and trust one another. Bubbles of trust confirm transactions in roughly 14 seconds, which is a lengthy time for real-time applications, and it also uses public blockchain, which requires fees to be paid for each transaction. Another noteworthy application conducted by the authors of (Lin et al., 2018) for Industry 4.0. This form of mutual authentication consists of four tangible levels that unite vertically inter-organizational value networks, manufacturing factories, and the engineering value chain. This conceptual framework enables the effective implementation of a smart factory that is adaptable and reconfigurable. This approach employed a one-time public/private key combination for each request for mutual authentication. This pair can be used to encrypt and decrypt messages and calculate message authentication codes. FairAccess in (Ouaddah et al., 2016) proposed an AAC system for the Internet of Things. It generates tokens for users based on their credentials in the authentication section. The next section of the paper will go through this strategy in further detail. FairAccess only supports token-based authorization and does not have a mechanism for renewing expired tokens. It takes longer for a token to become available and useful (at least two blocks must be mined). Similarly, the work presented in (Niu et al., 2017) presented an authentication mechanism for Wi-Fi hotspot access. The service provider, hotspot APs, users, and the blockchain are all part of this method. All users' credentials are recorded in the blockchain, and when a user requests to connect to the network, the service provider and Wi-Fi hotspot connect to the blockchain

to obtain valid credentials and establish the connection. This strategy can enable both accountability and anonymity at the same time. The Colored Coins as well as CoinsShuffle were two factors that had an impact on the design of this scheme. And in (Sanda and Inaba, 2016) the authors proposed adopting Bitcoin 2.0 as another authentication technique in a telecommunication setting. The user installs "Auth-Wallet" in this manner, which allows him to obtain authorization by exchanging "Auth-Coins" instead of user information. The goal of this strategy is to improve user privacy (Mohsin et al., 2019). The two basic protocols for implementing the desired authentication solution are registration and authentication. When a user first logs in, the Registration Protocol is used to submit user information to the Auth-Wallet server. The procedure of connecting to the internet is done via the Authentication Protocol. The user connects to the access point by entering its unique ID. The access point initiates a transaction that sends the user Auth-Coin. The mail is verified and signed by the user. If the access point's verification procedure succeeds, the token will be published to the blockchain, and the access point will allow users to connect to the internet. Furthermore, the authors of (Lee, 2017) offered an authentication management solution for telecommunications and the Internet of Things. This method created a blockchain-based ID for each user, then registered on the blockchain. This system is only used to register users as a distributed database. The mutual authentication is the most noteworthy security technique that can be found in this study.

By combining blockchain, smart contracts, and the cloud, the authors of (Manzoor et al., 2019) created a hybrid architecture for IoT data sharing. Cloud storage was used to overcome the Blockchain storage challenge. Only the owner and individuals named in the smart contracts had access to the data; hence the proxy-encryption technique was utilized as the security mechanism. A testbed was set up to determine the viability of a platform in terms of scalability and performance. For ensuring hierarchical IoT access restrictions, the authors of (Ma et al., 2019a) proposed a Blockchain-based distributed key management architecture that integrates Fog and Cloud computing. The Fog network, which includes a security access management

(SAM), is split into zones. To conserve storage for IoT applications, they split the blockchain into many side blockchains. Each SAM is in charge of a domain-specific side Blockchain. The cloud collects all side Blocks from each SAM and hosts multi-Blockchains to support cross-domain interactions. The suggested scheme was tested in OMNet++ to see how secure it was and how long it took to execute transactions. On behalf of IoT devices, the authors of (Almadhoun et al., 2018) introduced Edge servers to execute authentication using a smart contract on the blockchain. To relieve IoT devices of the effort of conducting an authentication procedure, the Fog nodes have an interface with the Ethereum Blockchain's smart contract. IoT devices can be accessed by Blockchain-enabled Fog servers linked to Ethereum smart contracts. Smart contracts were introduced in (Nguyen et al., 2018) to ensure that authorized users may access data without the involvement of other parties. The authors also proposed a Blockchain-based firmware update system for IoT devices to prevent fraud and data tampering. Furthermore, the authors of (Bao et al., 2018) presented an IoT Chain that is made up of three layers, namely, authentication, Blockchain, and the application layer. The architecture supports several services, including identity verification, access control, and storage integrity, without imposing significant overheads or delays. They claimed that the architecture had a lightweight characteristic and DoS attack fault tolerance. However, most of the presented approaches were just conceptual proposals without providing proof of concept or evaluation. There are providing a brief general overview of system design. Rather than that, we intend to provide a proof of concept to demonstrate our approach.

Table 3.3: blockchain-based identity solutions

Study	Purpose	Platform	Advantages	Limitations
(Al-Bassam, 2017)	Authentication for intelligent closed circuit television	Ethereum	Transparency, secure against rogue certificates	Privacy and transaction costs
(Liu et al., 2017)	Blockchain-based identity management system	Ethereum	Enhanced security and privacy. Tokens for entity reputation	Theoretical model only
(Axon, 2015)	blockchain-based PKI with privacy awareness	proved by Theorem	Network security, registration and revocation of keys	Scalability issue
(Augot et al., 2017)	Identity management system and ZKP	Bitcoin	Privacy-preserving	Credential revocation is not supported
(Hardjono and Pentland, 2019)	Blockchain-based identity solution based on ZKP	ChainAnchor	ZKP, privacy	High computational cost
(Halpin, 2017)	Identity and blind signatures to maintain anonymity	NEXTLEAP	Blind signatures for privacy	Scalability drawback
(Azouvi et al., 2017)	Blind signatures identity solution	Ethereum	Blind registration	Privacy of the public ledger. Performance limitations
(Naik and Jenkins, 2020)	A decentralised identity system	Ethereum	Secure data exchanges. Identity management	Issues of private key recovery
(ShoCard, 2020)	A decentralised identity system	Bitcoin	Eliminate the username/password authentication	Use of four-digit passcode
(Decentralized, 2021)	A decentralised identity system	Ethereum	Interoperability and support for W3C standards	JWT used only for authentication
(Khovratovich and Law, 2017)	A full-stack solution for identity management	Hyperledger Indy	Permissioned blockchains and anonymous credentials	Require knowledge of governmental records
(Alliance, 2018)	A decentralised identity system	open-source identity system	Design and implementation of ID solutions	Need cooperation from legislators
(Toorani and Gehrmann, 2021)	PKI based on blockchains and webs of trust	proved by Theorem	User-centric identity	chain forks prevent identities finalisation.

Table 3.4: Blockchain-based authentication and authorisation

Study	Purpose	Platform	Advantages	Limitations
(Kim and Jeong, 2018)	An authentication method in the cloud context	Mobile resource	Immutable credentials	In-chain storage
(Huh and Seo, 2019)	Blockchain-based fingerprint authentication for mobile	mobile resource	Prevent forgery	Performance due to the use of PoW
(Widick et al., 2019)	Authentication and authorisation for IoT device	Ethereum	Key management	Performance
(Lin et al., 2018)	Blockchain for smart factory	JUICE, Ethereum	multi-receivers encryption	Performance
(Ouaddah et al., 2016)	Blockchain-based access control framework for IoT	bitcoin (regtest)	Reliability	communication and processing overheads.
(Niu et al., 2017)	authentication scheme for Wi-Fi hotspot access	Bitcoin	Reliability	High cost
(Sanda and Inaba, 2016)	authentication technique in a telecommunication setting	Bitcoin	Reliability	Performance
(Mohsin et al., 2019)	Blockchain authentication of network applications	Bitcoin	Reliability	Authentication only
(Lee, 2017)	Authentication solution for telecom and IoT	BIDaaS	Identity management	Authentication only
(Manzoor et al., 2019)	Blockchain based solution for Secure IoT Data Sharing	Ethereum	Proxy Re-Encryption	Limited experimentation
(Ma et al., 2019a)	Blockchain-based distributed key management architecture	simulation	integrates Fog and Cloud computing	Multi blockchain complexity
(Almadhoun et al., 2018)	Authentication for IoT using Blockchain in Fog Nodes	Ethereum	Reliability	Communication overhead

3.10 Blockchain-based Access control

Access control systems for the Internet of Things are also designed using blockchain technology. In restricted IoT devices, blockchain-based access controls reduce the requirement for a centralised authority to create access control policies. They use blockchain transaction data models to execute access control actions in IoT contexts, such as granting or denying access requests. Another advantage is auditing access control settings, which can produce immutable blockchain transactions (Riabi et al., 2019).

3.10.1 Blockchain Based IoT Access Control Methods

IoT research has exploded due to the explosion of expanding items in communications and networking technology. Data sharing, ease of access, and remote monitoring are just a few of the benefits of connecting diverse smart devices over the internet (Riabi et al., 2019). One of the most serious problems with IoT is its centralised structure based on the client-server architecture. Because a lack of trust between different participating devices might lead to network failure, a reliable solution is required to avoid this problem. Several techniques have been presented in recent years, with blockchain gaining traction due to its decentralized nature, security, and immutability. This section will explain and discuss existing authentication methods that employ blockchain or smart contracts. In summary table 3.5, show the taxonomy-based classification of the approaches.

3.10.1.1 Attribute Based Access Control (ABAC)

The Attribute-Based Access Control (ABAC) architecture has been developed to make access control in the Internet of Things (IoT) easier (Ding et al., 2019). Blockchain technology is used in this project to append and retain the distribution of attributes, such as user attributes, resource attributes, and object attributes, based on users' needs. ABAC converts uniqueness or representations into a set of

attributes that the attribute authority publishes. Each collection of characteristics is represented by a series of Boolean equations, each of which has its own set of access regulations. Valid and allowed access is granted using these access policies. It takes care of distributing roles and creating an access control list for all system devices. According to the performance analysis presented in this work, the ABAC scheme provides a high level of confidentiality, robustness, flexibility, and scalability. Authors suggested an Attribute-based Access Control technique in (Zhang et al., 2020), which consists of five primary components: Consortium Blockchain Network, Authority Nodes (AN), IoT Devices, Chaincode, and Public Ledger, and Access Tree. The Consortium Blockchain Network's authority nodes handle all interactions with the Blockchain Network on behalf of IoT Devices. When a requester sends an access request to a target, it forwards it to the AN. The AN check the legitimacy of the requesters' uniqueness and the target's access rule by querying Chaincode and retrieving registered access credentials. After that, AN creates an access tree to do authorization. The final access information is recorded on the blockchain, together with the authorization outcome, after which AN sends the results to the requester. Attribute-based Access Control is performed by keeping three key-value databases: Device database, Attribute database, an Access database, all of which are closely related. After the attribute is registered, it is given a name, and individual owner lists are kept for each attribute when assigned to a device. Due to reduced storage and computation overhead, performance analysis reveals that the proposed system is lightweight and efficient.

3.10.1.2 Fair Access

In (Ouaddah et al., 2016) a completely pseudonymous technique with no central governance is presented to allow users to own their data. To achieve pseudonymity, all interacting entities are identified using bitcoin-like addresses, and access control policies are set in the smart contract and then preserved in the blockchain. Permission tokens are used as unique identification and to demonstrate the connection permission for access to a certain resource, are also distributed by blockchain.

Transaction integrity checks and a double-spending detection mechanism are in place to identify forgeries and token reuse. The suggested approach alleviates the burden of managing a large amount of admittance control data on restricted IoT devices.

3.10.1.3 Blockchain-based access control hub/Distributed Access Control

Oscar Novo (Novo, 2018) suggested a new method for decentralized access control in sensor networks that are geographically scattered. Wireless Sensor Networks, Manager Nodes, Agent Nodes, Smart Contracts, Blockchain Networks, and Management Hubs are all part of it. The authors used blockchain to store and distribute access control information in this technique. To represent all of the actions certified in the admittance regulatory system, a single exclusive and non-destroyable smart contract is used. The managers contact smart contracts to specify the structure's admittance rules. The key benefit of this strategy is enhanced scalability, as different systems can be linked to the blockchain setup at the same time via special nodes known as management hubs. In (Hwang et al., 2018), Hwang, D. et al. described a mechanism for exchanging data between geographically dispersed IoT devices. Rather than sending information requests straight to the device, they are forwarded to the management center, verifying access permissions stored in the blockchain. If the request is approved, the management hub retrieves data from the device and delivers it to the requesting device. This method is appropriate for devices located far apart and cannot communicate directly with one another. In addition, dynamic policy generation is proposed for devices that do not have access control rules registered. As a result of this method, increased scalability has been attained.

3.10.1.4 Blockchain-based Distributed Key Management

The authors of (Ma et al., 2019b) proposes a distributed key management approach based on blockchain for privacy-oriented IoT applications. Different side blockchains are established in the fog layer based on the deployment fields to speed

up the verification and conserve storage space. Fog computing is used to reduce wait times, and communal blockchains are used in the cloud layer to provide cross-domain access. Extensibility is higher, coupled with higher communication and processing costs compared to hierarchical methods.

3.10.1.5 Token Based Access Control

The authors of (Fotiou et al., 2019) describes designing a large event-based Internet of Things (IoT) control system utilizing tokens and smart contracts. Smart contracts provide a mapping between device operations and functionality. A smart contract generates a blockchain event whenever a client calls any function. The events are received by the appropriate IoT gateways, which eventually result in an action in the appropriate IoT equipment access. This technique has concerns with fluctuating monetary costs and transaction delays. Direct interaction between the client and the IoT gateways can help. A smart contract-based blockchain system based on Ethereum is deployed (Ourad et al., 2018b). The access control and authentication mechanism comprise a smart contract that verifies the client's identity using their Ethereum wallet address. The sender's access token and Ethereum address made public via the smart contract are only genuine if the client is genuine. The client and IoT devices receive this published information. The client creates a combination that includes the ethereum public key, the user's IP address, the access token, and the duration of access. The ethereum private key endorses this combination, followed by the transmission of the associated public key. To safeguard the integrity of the combination, it must be endorsed. After verifying the information, the IoT equipment assigns the admittance to the client against the sender's IP address for the specified interval. When any verification checks fail, the appeal or investigation is terminated. The evaluation of the proposed system revealed improved availability and scalability. In addition, the work presented in (Al Breiki et al., 2019) provide a permission authorization mechanism for IoT data using trusted oracles and blockchain. Oracles are gateways, serving as a connection point between the blockchain, service providers, and remote clients. The interface and permission to

utilize IoT data are governed by multiple smart contracts. They also keep track of reputation and enlist new oracles. Users can request access to IoT data by sending a request to a smart contract. After evaluating the access control policies, smart contracts validate the right of access to IoT data. Smart contracts authenticate the right to access and provide Access tokens to oracles and end-users after verification. Oracle-based access control allows heterogeneous storage and provides distributed access control with dynamic policy administration.

3.10.1.6 Access authorisations/Control Chain

For IoT access authorizations, the Control chain (Pinno et al., 2017) architecture, a fusion of four separate blockchains, is presented. All people's public recommendations and dealings are recorded in the relationship blockchain. Context blockchain is used to record environmental data such as refined data, physical inputs, and sensor reports, which can be utilized to make permission decisions. The confirmation of admittance acceptance or denial is maintained in the accountable blockchain. Endorsement regulations defined by device owners or devices are kept on a blockchain dedicated to rules. This architecture is more scalable, user-friendly, and well-suited to a wide range of IoT admission control representations.

3.10.1.7 Blockchain-based Attribute updates

Because of its contradiction with revocation of ABE or attribute updates, the blockchain's immutable nature is the main barrier to adopting Attribute-Based Encryption (ABE) in fine-grained access control. Chameleon Hash algorithm, (Yu et al., 2020) proposes a novel multilayer blockchain-based IoT system to simplify attribute updates in fine-grained access control. The technique can prohibit revoked members or miners from accessing impending as well as previous data without jeopardizing the blockchain's integrity.

3.10.1.8 Cloud-based access control

The authors in (Bera et al., 2020) designed an access control method with blockchain implementation to address confidentiality and security problems on the Internet of Drones (IoD) network. It allows access control between two neighboring drones in similar flying region as well as between the drone and its Ground Station Server (GSS). Ground Station Server gathers instantaneous data from drones and organizes it into blocks containing transactions. After that, the blocks are sent to the cloud server. Using the Ripple Protocol Consensus Algorithm, the cloud server that serves as the leader among other cloud servers will verify the block and add it to the blockchain (RPCA). The proposed system is safe against both "replay" and "man in the middle" assaults, according to simulation reports.

3.10.1.9 Access control based on multiple smart contract

The authors of (Sultana et al., 2020) uses the Ethereum blockchain to achieve permission-based service sharing, and authorized access control. To provide efficient access control administration, Access Control Contract (ACC), Register Contract (RC), and Judge Contract (JC) are three different forms of smart contracts. Setting different permission levels to enable permissioned access privileges for IoT users allows for secure service sharing. According to the results, the method is more cost-effective and less complex for access management and data sharing across IoT devices. Tables 3.5 shows a summary of Blockchain-based IoT access control solutions and a study of their performance.

3.11 Conclusion to this chapter

To sum up, this chapter reviewed the literature on blockchain and the decentralised authentication and access control in IoT, providing a coherent and comprehensive picture of the current state-of-the-art efforts in this direction. Thus, it provided an evaluation methodology for acquiring a better knowledge of blockchain and

distributed ledger technologies and their applicability to improving the privacy and security of the existing authentication and access management solutions for IoT. We first discussed the fundamental concepts of the current access control in IoT and summarised various access control mechanisms, such as ACL, DAC, MAC, RBAC, ABAC and CapAC. Then, the key issues of implementing existing access control systems in the IoT environment are discussed. After that, a summary of the key authentication factors is presented, including, Knowledge-based authentication, Possession-based authentication, Inherence-based authentication or Biometric-based authentication, and Multi-factor authentication. Then, it discusses the challenges of the existing authentication for IoT. In addition, we provided a discussion and comparison of the ways in which blockchains can be utilised to address these issues, including the need for blockchain in IoT and the blockchain characteristics. Finally, this chapter went further and provided an overveiw and classification of several techniques that employ blockchain or smart contracts technologies for authentication, authorisation, identity solutions and access control.

Fourth Chapter

Blockchain-based Lightweight and User-centric Two-factor Authentication for IoT

4.1 Introduction

Two-factor authentication is commonly used in Internet of Things (IoT) authentication to provide multi-layer protection. Tokens, often known as One-Time Passwords (OTP), are used to offer additional information. While this technique provides flexible verification and an additional layer of security, it still has a number of security issues. This is because it relies on third-party services to handle tokens or OTPs, which leads to serious information leakage issues. Additionally, relying on a third party to provide authentication tokens significantly increases the risk of exposure and attacks, such as man-in-the-middle (MITM) attacks, as many services after the first authentication, use a session token saved on the user's local system (Cekerevac et al., 2017). In trying to rectify this issue, this chapter proposes and develops a blockchain-based two-factor authentication method for web-based access to sensor data. The proposed method provides a lightweight and user-centric authentication that makes use of the Ethereum blockchain and smart contracts technologies. Then the performance of the proposed system will be investigated, and the system's

security will be analysed and discussed. Based on the evaluation results, the proposed method has proven to be effective and has the ability to facilitate reliable authentication. The remainder of this chapter is organised as follows. Section 4.2 provided an overview of the IoT and its data security, summarising different authentication methods, such as single and multiple-factor authentication. Section 4.3 looks at the problem statement and section 4.4 reviews the related work. Then, this chapter elaborated on our proposed solution in section 4.5 and illustrated the system's design in section 4.6. Subsequently, described the technologies used and the implementation process of the developed system in section 4.7. Then, section 4.8 presented the evaluation of the proposed approach, including performance and security analysis. Finally, concluded the chapter in Section 4.9.

4.2 IoT applications security

IoT and its applications were initially introduced as a concept encompassing cooperating and interacting of multiple components, such as wireless sensor networks and radio frequency identification (RFID). However, IoT is hampered by the gap that exists between the physical and information worlds, limiting the proper processing of data gained from the interaction of people with electronic equipment (Peña-López et al., 2005). Thus, virtual and physical accessibility must be available for these devices, and content must be accessible from any location. Thanks to developments in hardware technology and embedded devices, IoT devices can now consume web services directly by contacting REST APIs to interface with physical devices via local bridges or direct wireless communications. Using sensors, you may collect data such as humidity or temperature, which can then be recorded in a server-side database or shown in a user-friendly application interface. The common needs for all IoT applications are high flexibility, scalability, collaboration with multiple stakeholders, and the necessity for lightweight security procedures (Nižetić et al., 2020). Digital security concerns exist at every level of the IoT journey, and hackers stand ready to take advantage of any system weaknesses. Without proper security

measures, attackers will have the opportunity to gain control of users' credentials. As a result, secure IoT requires authentication and access control solutions.

4.2.1 IoT applications and users authentication

Authentication is used in the IoT to identify users, devices, and applications, as well as to restrict access to just authorised individuals and non-manipulated objects or services (Liyanage et al., 2020b). One-factor authentication, often known as password-based authentication, is a type of authentication that is commonly encountered in web applications (Duncan, 2001). However, one disadvantage of employing password-based authentication is that it is prone to brute force attacks, making the users' data open to theft. To save time, users frequently reuse passwords across platforms. This strategy is vulnerable. Hackers who gain user passwords from one platform have the ability to steal data from other platforms. As a result, password-based authentication cannot assure the security of account data. In this context, access control mechanisms such as single sign-on (SSO), Multi-Factor Authentication (MFA), Open Authentication (OAuth), open ID connect, and other forms of authentication are key alternatives. For instance, SSO allows an entity to be authenticated using a single set of login credentials and granted access rights to multiple applications and services in a cloud platform, eliminating the need for additional prompts when the user switches applications or services during the same session (Koundinya and Baliga, 2020). However, several businesses have chosen to implement multi-factor authentication to authenticate a user's identity, necessitating the usage of multiple identification and access management credentials (Duncan, 2001). As a result, MFA might be viewed as a feasible method of enhancing security. However, these models necessitate security evaluation and scalable security management systems.It is called two-factor authentication when the first-factor authentication and the second-factor authentication are both used in conjunction. Two-factor authentication is most commonly accomplished through the use of an email address or a username and password combination. When using

two-factor authentication, the user must provide additional information in order to be authenticated. In order to offer additional information, tokens or one-time passwords (OTPs) can be utilised. As long as a third party deliver two-factor authentication tokens, they are vulnerable to attack because a MITM can steal tokens and find that all generated tokens are equal, as revealed by the attacker (Mail and Box, 2017).

4.3 Problem statement

Increased privacy and security concerns have arisen as a result of the Internet of Things' extensive use and implementation (Sezer, 2018). Due to the lack of human interaction, wireless communication makes IoT more open to assaults such as message manipulation, message eavesdropping, and identity theft. As many devices have limited resources, they may not be able to implement more advanced security measures. In addition, centralised architecture is increasingly adopted in current IoT solutions, which connect to cloud servers via the Internet. This strategy provides amazing elastic computation and data management capabilities for IoT systems as they become more complex, but it still faces a number of security and privacy concerns. For instance, the cloud-based design could lead to high expenses and latency, as well as a single point of failure due to its centralisation (Sezer, 2018). Scalability is another concern with the centralised IoT data paradigm, given that hundreds of nodes are involved. In addition, current authentication and access control standards are predicated on trust, compromising user transparency and privacy by introducing a centralised trusted authority. We can deduce that a more user-driven access control paradigm provides users complete control over their identities with their own distinctive granularity is essential in light of the previously discussed concerns and the sensitivity of the data obtained by IoT devices. A significant amount of potential exists, on the other hand, for blockchain and other decentralised alternatives that do not rely on a third party to manage the trust. Increasingly, decentralisation of trust is becoming the dominant trend, opening the

door to decentralised and autonomous access control and authentication in the Internet of Things applications. The benefits of blockchain and distributed ledger technologies motivated this chapter to propose that blockchain and smart contract technologies be used to create a secure and reliable two-factor authentication solution. In order to accomplish this, a smart contract is implemented on the Ethereum blockchain, which serves as an independent, secure channel for the transmission and verification of OTP. Our approach gives consumers complete control over their identities by allowing them to utilise the uPort identity mobile app to store their private keys in a secure location.

4.4 Related work

Blockchain technology has been employed in a variety of contexts recently for two-factor authentication systems. For instance, the authors of (Amrutiya et al., 2019) presented a two-factor authentication system based on blockchain technology. The work proposed a use case for adding an extra layer of protection to the OpenSSH server, a commonly used application for the Secure Shell (SSH) protocol. However, their system's token does not meet the standard for a token, which is to be a one-time use password. The proposed framework in (Alharbi and Alghazzawi, 2019) is also utilising Blockchain technology to complete the 2FA process. The proposed framework sends an encrypted OTP generated by a smart contract and its hash value to the application website. However, this system continues to deliver the token through SMS using OTP-SMS. If the phone is lost and the attackers manage to take control of it, they can obtain the token and gain access to the system. Similarly, the work proposed in (Danish et al., 2019) provided a blockchain-based two-factor authentication solution for LoRaWAN join procedures. However, the initial joint request experiences a significant delay owing to the mining process carried out in the blockchain network. In addition, it is imperative to make alterations to the firmware of end devices to enable the successful implementation of the two-factor authentication, as these devices are required to process particular data. This

modification of firmware enhances the interdependence between the LoRaWAN and blockchain networks (Ribeiro et al., 2020).

The proposed blockchain-based framework adds an additional layer of security to the LoRaWAN join operation and fosters trust among LoRaWAN network components. Compared to earlier proposed alternatives, the proposed approach in this chapter utilises blockchain as an independent layer for generating, sharing, and verifying access tokens. The proposed technique uses a smart contract to generate random challenges that must be signed using the users' mobile wallet private keys. Thus, represents users as real people who can fully express themselves using their mobile.

4.5 The proposed solution

This section presents the proposed solution, which uses the blockchain and its smart contract functionality to provide a secure authentication and access control mechanism in a secure and decentralised manner. The proposed approach addresses significant privacy concerns, eliminate the need for a third party to maintain trust, and mitigate the risk of using weak passwords for authentication into IoT applications. The proposed model aims to develop a Lightweight Two-factor Authentication scheme that takes advantage of the decentralised trust given by smart contracts and blockchain technology. This will satisfy the National Institute of Standards and Technology's (NIST) security criteria for two-factor authentication (Burr et al., 2006). According to the NIST guidelines, to avoid the issue of storing, protecting, and exchanging a secret between prover and verifier, it is critical to have two channels for the transmission of second-factor authentication (disposable) secrets that are as independent as possible. Consequently, in our approach, the blockchain will be employed as a second independent channel to exchange the second factor. We will rely on the Ethereum blockchain (Buterin et al., 2014), which will allow us to use smart contracts to accomplish our goals. The smart contract will be used to perform on-chain access control decisions and any process

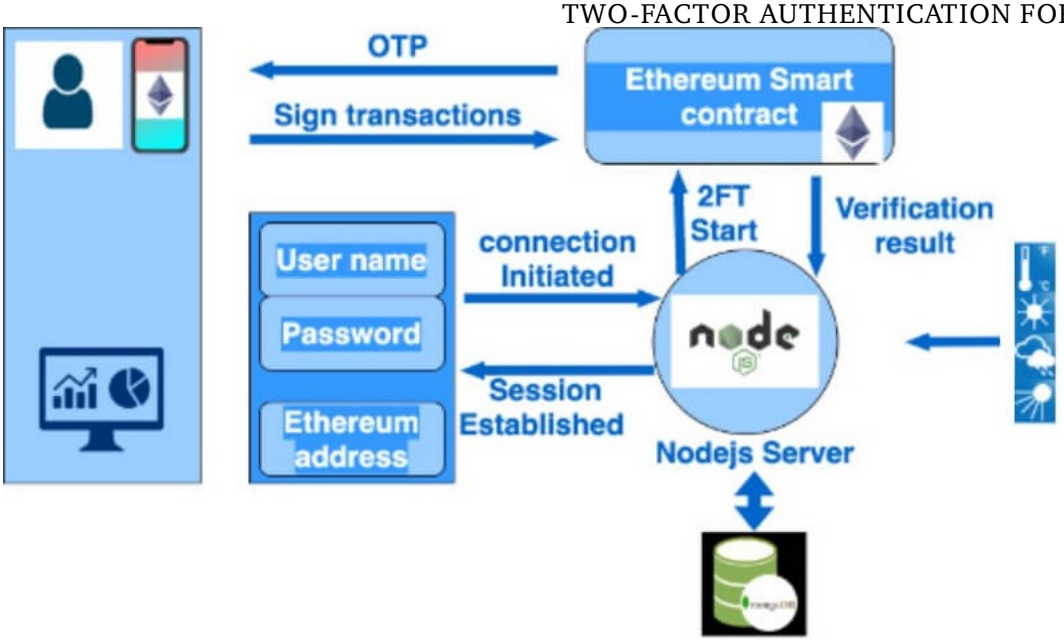

Figure 4.1: The architecture of the proposed 2FA system

related to authentication. The smart contract will also generate tokens that are utilised by the resource owner to acquire data from the IoT sensors layer. The access token will function as an OTP that users must sign in order to prove their identity. We no longer require a centralised trustworthy authority with our method. Our scheme allows users to save their identities directly on their mobile devices, which can then be accessed throughout the validation process. The decentralised web application will request users' credentials by requesting signing transactions from their mobile apps.

4.6 System design

This section goes through the primary components of the proposed model as well as their functions. Our system consists of five main components, as detailed below. Figure 4.1 illustrates the authentication scheme and its accompanying processes.

4.6.1 Users identity

We leverage Blockchain-based identification characteristics in our system, enabling more secure management and storage of digital identities. Users of our system

are Ethereum clients, each of whom has a public and private key pair to have access to the system. Users can sign transactions with their private key, whose hash is used to represent the user's address and is linked to their access token. Therefore, users will be able to maintain and store their own identities rather than rely on a third party to do so. The primary difference between our proposed system and the existing method utilised by Ethereum clients is that we employ Ethereum addresses or secp256k1 publicKeys as fully self-managed Decentralised Identifiers (ID) and wrap them in a DID (Decentralised IDentifier) Document in our proposed system. The distributed ledger can then be used to store the DID. In addition, we use a DID resolver, which takes the Ethereum address, examines contract events, and generates a DID document based on the ERC1056 Events associated with the address. This can function as a piece of information that serves as a pointer to a specific identity. This makes DIDs globally unique, highly available, and cryptographically verifiable. Therefore, decentralised identifiers can be associated with various entities, such as individuals, organisations, and government agencies. This is the concept known as Self-sovereign identity, which employs a similar combination of blockchain and cryptographic technologies.

In addition, signing with a mobile wallet allows smart contracts to validate users' credentials. Users can save their credentials directly on their devices, eliminating the need for a centralised authority to handle the data. We will use uPort (Naik and Jenkins, 2020) decentralised identity function to securely share and communicate information between the user's mobile application and our application. With uPort, the crucial issue of storing private keys can be addressed via blockchain. To accomplish this, uPort employs the blockchain as an identity certification authority, where a smart contract represents a user's digital identity and permits the revocation and replacement of that user's keys. Users have full authority over their identities and private information thanks to uPort's mobile self-sovereign identity wallet. Users on the Ethereum blockchain can make a new identity with uPort. Therefore, unlike other systems that treat users as hex-encoded addresses that communicate with the smart contract, our system treats users as real people who can fully express

themselves using their mobile.

4.6.2 Smart contract

In our system, we employ the public Ethereum-based blockchain because it facilitates the development of smart contracts. The proposed architecture makes use of smart contracts to provide two-factor authentication for data access from IoT devices. Our system's smart contract will be used to authenticate users and interact with data stored on the blockchain. Additionally, they provide resiliency through the execution of smart contract code across all blockchain nodes. The smart contract will be used to implement regulations such as decision-making regarding on-chain access restriction. Smart contracts will assist in the issuance of tokens that system users will use to authenticate to the IoT application. The smart contract issues the access tokens that are needed to authenticate and access the system. The access token will serve as a one-time password (OTP). This token indicates a challenge value that a user must sign to successfully authenticate to the system. In this scenario, our smart contract will help to manage, store, deliver and securely generate this challenge value. Since smart contracts can interface with data stored in the blockchain, so will utilise some random values that are produced during the block generation process, such as the block's timestamp and the block difficulty to perform as a source of entropy to facilitate the generation of the random challenge (Token).

By utilising smart contracts, users are relieved of the need to keep their tokens locally, as they assist in the generation of access tokens and management of their authentication process. Therefore, this will reduce risk associated with generating, storing, and sharing randomness between different parties. In addition, the access token integrity and authenticity, which serves as a one-time password, can be verified via a simple lookup in the distributed ledger. The web3 JS library, which communicates with the smart contract via RPC, will be used to access all of the smart contract's services.

4.6.3 IoT end devices

This component consists of a variety of resource-constrained devices, including microcontrollers capable of communication and data storage, as well as processors equipped with temperature and humidity sensors. These sensors assist in converting sensor readings to a readable format and syncing them with a user account.

4.6.4 Data base

Because blockchain technology is not designed to handle huge transaction data payloads, alternative data storage technologies are required to handle massive amounts of IoT data. Our system employs a different software solution based on the Mango database to support big file storage and minimise duplication across the entire blockchain filesystem.

4.6.5 Decentralised web interface

4.6.5.1 Registration

First, users need to visit our decentralised web interface and create an account by registering an e-mail address and password. Before it is accepted and a user account is created, the e-mail address and password are verified. E-mails are verified for proper formatting, and passwords must be at least eight characters in length and contain uppercase letters, symbols, and numbers. If e-mail and password being registered do not satisfy these requirements, an error notice will be presented. If the user's e-mail address is already registered, they should register using a different e-mail address. Additionally, the user must supply a public Ethereum address that will be utilised for second-factor authentication.

4.6.5.2 User login

For the first factor verification, user required to enter email and password that they registered on the registration menu. An error notice will show if the email

and password entered do not match. The blockchain will be used as a second independent channel to exchange the second factor (token) if the first factor and prerequisites are met. As part of this, the decentralised web app (web server) will call TokenGeneration() function in the smart contract and give the user public Ethereum address associated to this user account in the registration menu. This function will receive the user's decentralised identity (public Ethereum address) and then generate a random challenge, which will be associated to the user Ethereum address. The smart contract makes use of some random values generated by the blockchain network, such as the block's timestamp and difficulty, which are not known until the block is mined. These variables will serve as a source of entropy, facilitating the random challenge generation. The smart contract will then emit event that containing the produced random challenge, which need to be signed by the user's private key in order to prove their identity. Then the user will be directed to the two-factor authentication page which will present a signing request in form of a QR code that need to be scanned by the user's mobile wallet in order to sign the received token back to the smart contract and authenticate to the system. Following that, the web server will call a function that contain the user Ethereum address parameter in the smart contract. The function will return the value of the token in the array with the index corresponding to the user's Ethereum address in the session struct. The web server will next verify that the token obtained via the smart contract function call matches the token stored in the session struct, and if it does, the user will be granted access rights. In the event that the token value is not identical, an error message will be displayed. Our two-factor authentication system that does not rely on third parties to produce and distribute tokens. Additionally, users authenticating in the IoT system do not need to enter tokens because all checks are performed automatically by the dApp. Additionally, users will retain complete control over their authentication information.

Algorithm 1 User login process

Signup
username: = *read*username();
password: = *read*password();
Ethereumadd: = *read*Ethadd();
If username and password = Exist
return: user already registered
If username and password = invalid
return: invalid
else
register user1 and password1 and usr1$_E$*thertumaddress*

login
username: = *read*username();
password: = *read*password();
If username = user1 password = password1
return: user1Ethadd

2*FA via Smartcontract*
Call function in smart contract: TokenGeneration(user1Ethadd);
prove: msg.sender;
If approved
Token: unit256(unit256(keccak256(abi.encodePaked(Block.timestamp, Block.difficulty)));
return: user1token=Token;
request signing from user1
require web3 require abiEth
Contract=contractaddress &API
Tx=transaction.sign(user1token);
call.proveToken(user1token);
If user1token = Token)
return: True
else
return: False

4.7 Implementation

This section explains the technologies used and the implementation process of our system. The prototype is made up of two main components, the smart contract and the decentralised web application.

4.7.1 Smart contract implementation

We used the Ethereum blockchain to create our proof-of-concept prototype. Our key motivation for selecting the Ethereum blockchain is the amount of support

available due to its popularity and the ability to implement smart contracts on its network. Our smart contract was written in Solidity, a contract-oriented high-level programming language that is used in Ethereum blockchain systems to construct smart contracts. We used the Ethereum web browser-based IDE Remix (*Remix* n.d.) to build, test, and deploy the smart contract on the Ethereum network. Additionally, we measured gas consumption using the debugging tools included with the Remix IDE. The IDE also includes a compiler for testing smart contract functionality. Our contract uses a token-indexed mapping data structure to map the client's public key to its access token. The smart contract's unique address (a 40-characters hex string) can be used to access it. The blockchain API allows our software to communicate with the smart contract. The smart contract was accessed via a REST API endpoint on an Ethereum node run by Infura (*The world's most powerful blockchain development suite* n.d.) rather than our own Ethereum node. We used the web3.js Ethereum JavaScript Application Programming Interface to interface with the Ethereum blockchain and our application (API). The APIs and developer tools provided by Infura enabled rapid and scalable access to the Ethereum networks. The web3 JS library, which communicates with the smart contract via RPC, will be used to access all of the smart contract's services.

4.7.2 Decentralised web app

Our decentralised web app will present data obtained from sensors into the web interface. The decentralised web app allows for account creation, login, and communication with Ethereum smart contracts. There is a backend and a frontend, both of which use node.js servers, and they are hosted on an Apache server. Multiple interfaces are created using web front-end technologies, including HTML5, CSS, Bootstrap, jQuery and JavaScript, to facilitate user-to-blockchain interaction. All the blockchain functionality and services are available as representational state transfer application programming interfaces (REST APIs) that web clients or IoT devices may access. The client will communicate with the REST server, which

allows the end-user to execute appropriate APIs and submit transactions using HTTP GET or POST requests. The backend is implemented using the node js framework, which includes a server that reads data from a digital temperature and humidity sensor. We'll need a way to dependably store the flows that each user produces because container file systems aren't persistent. We took advantage of Node-RED, which provides an API that allows us to specify how and where things are shown. We wrapped the collections in MongoDB and used the passport-local-mongoose plugin to handle the password hashing. Finally, to develop the sensors prototype, a DHT22 sensor and an ESP8266 NodeMCU were used to read temperature and humidity measurements and then communicate them to a remote Nodejs application.

4.8 Evaluation

This section presents the evaluation findings of the performance of the proposed IoT blockchain platform. The evaluation system runs Ubuntu Linux 20.04 LTS on an Intel Core i5, 3.00GHz processor, and 8 GB RAM. Several experimental tests were conducted utilising various performance measures in order to present a comprehensive picture.

4.8.1 End to end delay

The service execution time comprised the time it takes to send a transaction request as well as the time it takes for the web client to receive confirmation. We calculated the overall duration of establishing a secure connection using our system's internal timing function. We note that Two major transactions must be performed in order for a user to complete the authentication process, invoking features in the smart contract that assist with authentication. The TokenGeneration() function is used to produce the client's access token. Second, we have the SetToken function (). Other smart contract operations, such as RetrieveTokeen(), which our application uses to retrieve users' signed tokens, are defined as view functions, which have

no CPU overhead, delay, or cost because they just read the state of the blockchain without altering it. Our testing revealed that the authentication process takes up to 22 seconds to finish, which is a significant amount of time. This is a well-known risk associated with the open Ethereum blockchain. This is because transactions take an average of 15 seconds to complete. However, when this is compared with realâworld applications that depend on a third party for implementing two-factor authentication, it has also shown a considerable delay. For instance, Reports showed that SMS can fail to get to the end-user about 15% to 25% of the time (BrankoviÄ, 2021). SMS centres can become obstructed as a result of the huge amount of data sent over SMS obsolete channels, including government organisations, which frequently causes disruptions and delays in regular data transmissions (Thatha, 2012). It has also shown a considerable delay in email-based 2FA, which experiences significant delays on congested networks.

As a result, this would only be an issue for programmes that require argent access to the public ledger. Additionally, by applying our approach in a private permissioned blockchain, we may considerably improve its effectiveness. Along with the private chain, switching from the PoW consensus method to a less computational mechanism such as Proof of Stake (PoS) (Siim, 2017) or Proof of Authority (PoA) (Wood, 2015) greatly reduces the time required to mine the block and also speeds up transactions.

4.8.2 Transaction's cost

The gas cost of each event in the system will then be calculated. We tested our proposed technique on the Ropsten Ethereum test network. The most significant cost, according to our findings, is the cost of deploying the smart contract. This action, however, will only be performed once when the system is first configured. The cost of deploying our smart contract was around 0.000101ETH Ethers, which equated to $0.03732657 on October 17, 2020, average Ether price of $369.57 (Coin Market Cap, n.d.). In comparison, the cost of calling a smart contract function

was around 0.000041ETH, or $0.01515237 per transaction. However, due to the financial considerations involved, our version is not dependent on the main Public Ethereum network. Instead, we constructed our implementation on the Ropsten test net, which offers a faucet to request free Ethers to this testing network. Our approach showed up to 0.000041 ETH (41000 gas units) per transaction compared to GnosisWallet (ConsenSys, 2019), which requires 275k gas units, TrezorMultisig2of3 (Capital, 2019) requires up to 95k gas units, and SmartOTP (Homoliak et al., 2020), which requires up to 150k gas units per transaction.

4.8.3 Security of our system

This section will highlight our proposed approach's security against security attacks. We'll start by assessing our model's security margin against various threats.

4.8.3.1 Man in the Middle attack

The proposed architecture will make use of the blockchain's cryptographic signature to assist prevent MITM attacks. The user will be required to sign the challenge using their private key in order to gain access to the system. The system will stay secure against such attacks due to the fact that authentication is performed via a smart contract on the Ethereum blockchain. The main difference between our work and the other existing 2FA that are using encryption is that some of the approaches, such as the work presented in (Amrutiya et al., 2019), which proposed a use case for adding an extra layer of protection to the OpenSSH server, a commonly used application for the Secure Shell (SSH) protocol. However, their system's token does not meet the standard for a token, which is to be a one-time usable password. Another similar work was presented in (Alharbi and Alghazzawi, 2019), which proposed a framework that utilises Blockchain technology to complete the 2FA process. The proposed framework sends an encrypted OTP generated by a smart contract and its hash value to the application website. However, this system continues to deliver the token through SMS using OTP-SMS. If the phone is lost

and the attackers manage to take control of it, they can obtain the token and gain access to the system.

In contrast to previously suggested alternatives, our methodology incorporates the use of blockchain as a separate channel for facilitating the dissemination of the second factor (Token). The methodology employed in our study involves the utilisation of a smart contract to generate, distribute, and authenticate the access token. The smart contract will generate challenges of a random nature, which necessitate the signing process through the utilisation of the private keys associated with the users' mobile wallets. Therefore, users are portrayed as real persons who can completely express themselves through their mobile devices.

4.8.3.2 Cryptographic attack

Our system is reliant on the Ethereum test network, which is open to the public. The cryptographic design of Ethereum is broadly acknowledged among cryptocurrency systems that use the KECCAK-256 hash algorithm, which is used by Ethereum. As a result, our approach, like other Ethereum keypairs, is immune to brute-force attacks.

4.8.3.3 Attacker on the network

Because our system is based on the Ethereum public test network, transactions will be exposed to all nodes participating in the blockchain system and to any users performing a simple blockchain lookup. As a result, further care must be taken to prevent attackers from compromising the system by authenticating to the web server using an existing token. Our system is immune to such attacks since our smart contract generates and maps a unique challenge to a particular user. Additionally, we demand that the user sign this challenge using his or her private key in order to complete the process and allow system access.

4.9 Conclusion to this chapter

This chapter provided a proof-of-concept design and implementation of a blockchain-based two-factor authentication system for web-based access to sensor data. The proposed method provides a user-centric and lightweight authentication solution. The proposed approach makes advantage of the Ethereum blockchain's smart contracts scripting capability. The proposed solution excludes the use of a third party to maintain the two-factor authentication process or OTP generation and validation. We employed the use of blockchain decentralised identity features to allow users to have full control of their authentication information rather than have it managed by a third party. In addition, our approach will enable users to store their identities straight in their mobile wallet app. This will allow users to represent them-self as real people. This chapter goes further and provides performance and security analysis to prove the feasibility of the proposed solution. Based on the evaluation results, our method has proven to be effective and has the ability to facilitate reliable authentication.

fifth Chapter

Decentralised identity and authentication mechanism for MQTT protocol

5.1 Introduction

The publish and subscribe messaging model has proven itself as a dominant messaging paradigm for IoT systems. An example of such is the commonly used Message Queuing Telemetry Transport (MQTT) protocol. However, the security concerns with this protocol have presented vital security challenges in most IoT applications. For example, the MQTT protocol does not have secure authentication mechanisms implemented and leaves that task to the developer as all the included native security services are fragile. To improve the security of MQTT, this chapter proposes a decentralised solution involving a lightweight authentication and authorisation scheme together with a decentralised identity system to manage the users' identities. The proposed mechanism helps in facilitating the authentication for both subscribers and publishers by utilizing a smart contract in Ethereum blockchain to guarantee trust, accountability and preserve user privacy. We provided a proof-of-concept implementation to prove our work, which involves a decentralized MQTT platform and dashboard using our approach. The usability of this approach was further analysed, particularly concerning CPU and memory utilization. Our analysis proved that our approach satisfies IoT applications' requirements since it

reduces the consumption of resources and that smart contracts help in the automation of data management processes. The remainder of this chapter is organised as follows. Section 5.2 provided an overview background on the MQTT protocol. Section 5.3 identify the problem with current MQTT protocol, and the related work is presented in section 5.4. The proposed approach is presented in section 5.4.1. The whole authentication process and system design are described in section 5.5. The decentralised identity model is discussed in section 5.6. In section 5.7, we described the implementation of our mechanism. To evaluate our work, we provided an intensive performance evaluation in section 5.8. and security analysis in section 5.9. Finally, we concluded the chapter and provided a suggestion for future work in section 5.10.

5.2 Overview of the MQTT messaging protocol

The HTTP is currently used in a request-reply messaging model to facilitate web data exchange and handle data acquisition from the hardware. However, it has proved unsuitable for use in resource constrained IoT systems (Yokotani and Sasaki, 2016). Since IoT is mainly used in resource-constrained devices, its protocols mandate that it uses low energy, gives back a real-time response and less bandwidth. The use of CoAP (Shelby et al., 2014) provides a temporary fix as a lightweight request-reply protocol that is ideal for resource-constrained IoT systems due to its low level of consuming resources. However, it lacks the portability and scalability required by most functions. Over the past several years, various messaging models have emerged known for their resource efficiency and low communication overhead (Jaikar and Iyer, 2018). This includes NesC (Gay et al., 2003), LooCI (Hughes et al., 2009) and MQTT (Standard, 2014). The MQTT messaging protocol is a lightweight publish-subscribe messaging protocol popular for key advantages over others, such as low energy consumption. It can also be used from smartphones, a functionality that other models such as the COAP fail at.

Compared to the HTTP protocol, MQTT has superior features when used on

android devices by slowing down energy consumption. Most IoT devices will often depend on the MQTT protocol to exchange data. MQTT, which is now registered under the standard ISO/IEC 20922:2016 is specifically created for IoT and requires low energy, offers real-time response and less bandwidth (Standard, 2014). MQTT relies on TCP/IP, Bluetooth or UDP, thereby a lightweight protocol that requires constant connection and minimizes messages. The use of MQTT offers various benefits such as low energy use and can be used across all smartphones. The MQTT session is separated into four parts: connection, authentication, communication, and termination (Standard, 2014). Each of these steps is described in more detail further into this chapter.

5.2.1 The Publish-Subscribe Messaging Model

A publish-subscribe communication model is made up of three main components. These are the subscriber, the publisher and the broker, as shown in Figure 5.1 below. The broker receives messages from the publishers, which are in the form of topics. The topic in this case is the metadata that has the information about the data in string format. Under the MQTT protocol, the process starts by the creation of the TCP/IP connection by the client to the broker. The client then moves to the authentication phase.

5.2.2 Users' identities

User identities are determined using identity management methods. Most current systems have their user's identities controlled and maintained by third-party applications or protocols such as single-point services or identity providers. However, the main issue with this approach is that identity is owned by the mentioned providers rather than the rightful owners. As blockchain technology grows the use of self-sovereign identities (SSI) (Tobin and Reed, 2016) has increased, which utilised the benefits associated with decentralisation. The use of SSI allows users to control and own their personal identities and other benefits such as decentralised

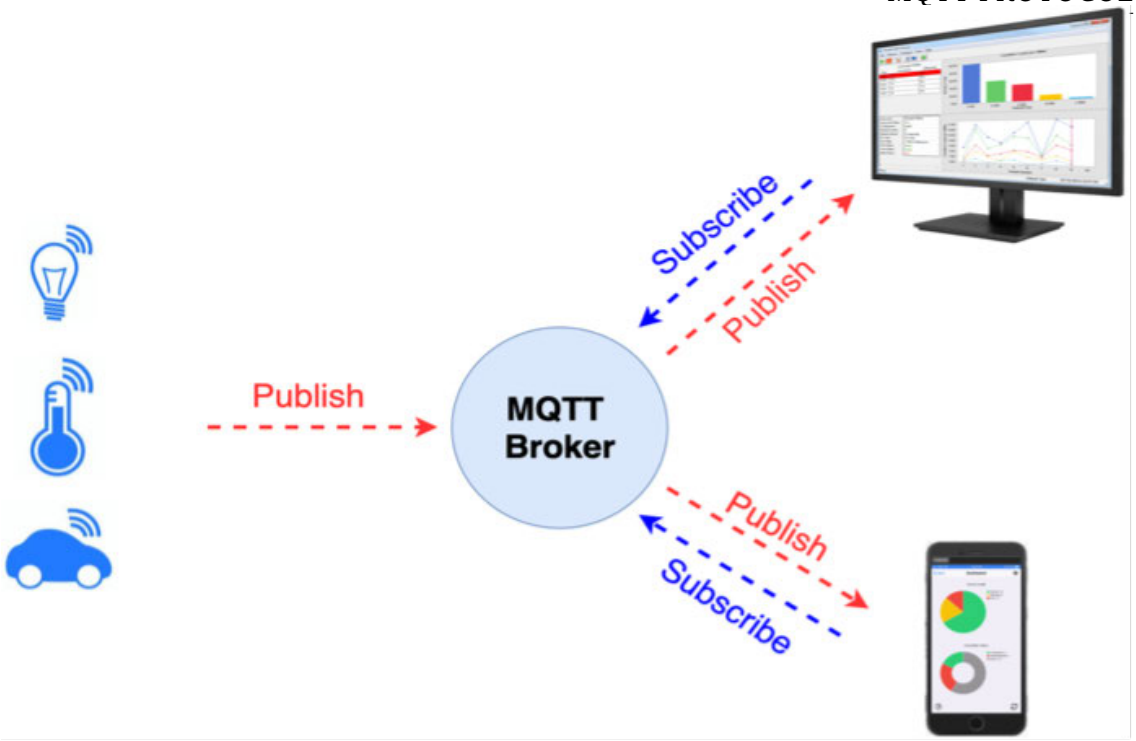

Figure 5.1: MQTT messaging protocol

control and privacy. For SSI to be realized, there is a need for the implementation of two key standards, such as Verifiable Credentials (VCs) (Longley et al., 2019) and Decentralized Identifiers (DIDs) (Reed et al., 2020). VCs are instrumental in facilitating the authenticated attribute disclosure and the privacy-aware while DIDs focus on cryptographic identification.

5.3 Problem definition

Currently, the large-scale deployment and adoption of IoT have increased privacy and security challenges. IoT remains vulnerable to attacks because: a) being that communication is wireless, the system faces increased risks of attacks such as message tampering, message eavesdropping and identity spoofing. b) most devices will often have access to limited resources that will prevent them from effecting advanced security solutions (Samaila et al., 2018). These resources include processing capacity, memory and energy. Another key challenge faced by IoT is the centralisation of efficient security solutions such as Public Key Infrastructure (PKI),

which leads to scalability issue owing that there are thousands of nodes connected in such an environment. IoT is also plagued by integrating new scenarios or services since each component relies on a different type of architecture, deployment and security approach. In addition, IoT related issues such as privacy and security remain the most critical issues that plagued the use of IoT systems. For instance, there are a number of issues and vulnerabilities that arise from different scenarios in both the devices and protocols. An example of a potential source of vulnerability to IoT systems is the use of the Message Queueing Telemetry Transport (MQTT), which only comes with a weak security mechanism. The authentication system uses a connect message to transmit the username and password in plain text. The choice of the security approach is left to the application designer. The Transport Layer Security (TLS) (Dierks and Rescorla, 2008) is not mandatory while most real-life applications will rely on it, which increases the likelihood of bad or incomplete implementations of security approaches and protocols. This creates the need for new approaches to enhance the security of MQTT.

In Turing to rectify the previously issues, the main contributions of this chapter can be summarised as follows. Firstly, it introduces a decentralised and light-weight weight approach, which embodies a methodology to achieve authentication of remote IoT devices without relying on any third party. Secondly, proposed a blockchain-based authentication and authorisation mechanism for the MQTT protocol. Thirdly, developed a decentralised identity system to manage the users' identity. In addition, the development of a smart contract in the Ethereum block-chain to facilitate the publisher and subscriber access to the MQTT broker without the need for a centralised trust. Finally, provided a proof-of-concept implementation to verify the feasibility of our solution.

5.4 Related work

The security of the Internet of Things has received significant interest from the scientific society. A significant of current researches are focused on the authen-

tication and authorisation in the IoT messaging and communication protocols. An example of such researches, the works presented in (Patel and Doshi, 2020) (Pahlevi et al., 2019) (Lohachab et al., 2019) (Bali et al., 2019) (Cruz-Piris et al., 2018) which provided authentication and authorisation solutions for the MQTT messaging protocol. However, the main issue with these approaches is centralisation. The problem with the centralised authentication approaches is that it requires to store authentication data on a centralised local server, which is prone to a single point of failure. Similar to our approach, the work presented in (Buccafurri and Romolo, 2019) has a similar focus as it is presenting a blockchain-based OTP authentication approach for resource-constrained IoT devices. The implementation is based on MQTT protocol. The Ethereum blockchain is being used to provide an independent channel to manage the second-factor authentication through the use of a smart contract. Similarly, the authors in (Buccafurri et al., 2020) proposed a blockchain-based OTP authentication scheme for the MQTT messaging protocol. The proposed approach utilises Ethereum blockchain to provide an out of band channel for implementing the second-factor authentication. Our approach advantage in comparison with other approaches is that it provides an integrated solution. In addition to authentication, our smart contract is also responsible for managing the user's policies and the authorisation. Therefore, our mechanism works as access control to prevent unauthorised access to the IoT systems. Besides, storing and sharing a secret between entities remains a challenging task as it requires a secure channel before it can be proven via blockchain. Therefore, the random challenge generation process in our approach is also based on the smart contract. The smart contract uses the blockchain-based random values to act as the source of entropy, facilitating the random challenge generation. This will help in reducing the risk associated with storing, protecting and sharing a secret between a verifier and a prover.

5.4.1 The proposed solution

To resolve significant privacy concerns that arose from relying on a third party to maintain trust and maintain customers' sensitive data, which makes it vulnerable to misuse and attacks, and remove the need for remembering passwords and prevent weak passwords from being used for authentication, we proposed using blockchain technology to facilitate decentralised authentication and authorisation in the MQTT messaging protocol. We will rely on Ethereum blockchain, which allows us to use smart contracts. We rely on smart contracts to implement on-chain access control decisions and other policies. We rely on smart contracts to implement on-chain access control decisions, set the users' policies and register the client's remote devices. We will also use smart contracts in our approach to storing a trusted mapping between the authorised access's public key and its access token. The smart contract is responsible for any operations involved with the authorisation and authentication process such as the whitelisting of all addresses that authorised the access token, generating the users' tokens, and retrieving access tokens issued for the authorised users. The smart contract will also be responsible for authenticating users by receiving signed challenges from users.

The smart contract issues the tokens used by the subscribers and publishers to connect to the MQTT broker. The users can interact with the smart contract by issuing transactions signed by their private key. The hash of the used key is taken to be the user's address and will be used to associate the user to their access token. Our proposed approach will reduce the entropy required and storage requirements. As the smart contract will generate access tokens and manage the authentication process; therefore, there is no need for users to store their tokens locally. The access token will serve as a one-time password (OTP). A simple lookup in the distributed ledger will help verify the authenticity and integrity of all the tokens. Our solution further looks into the provision of decentralised identity management services such as secure and fair exchange, revocation and verifiable credentials by using the power of smart contracts. During the programs preliminary phase, users are

prompted to register their remote devices, assign users to specific topics, set policy for the users and get back a verifiable claim. The smart contract helps in assigning the users' policy, after which we implement a decentralised identity model, which will be tasked with managing the users' identities. The users will have the ability to control their identities. This will be made possible through the use of self-sovereign identity (SSI). For this, we will utilise Uport decentralised identity functionality, which will help us to securely share and communicate information between the user's mobile application to our application. With our approach, we no longer need a central trusted authority. The application allows user to store their information straight on their mobile devices, which can be accessed during the validation phase, which makes use of our decentralised web application to retrieves users' credentials and request signing from the user.

5.5 The System designs

Before we can proceed to the authentication procedure and formalise the required concepts, it is important that we extract our solution's model. Our system established a link between the blockchain system and the MQTT components (broker, publisher and subscriber). Our system has three main entities: the Blockchain, MQTT system, and the decentralised identity system.

5.5.1 The blockchain system

Blockchain in our model is used to store information in a distributed manner while maintaining consistency. For this, we used Ethereum blockchain, which allows smart contracts to be deployed in their blockchain system. The smart contract is used in our approach to interfacing with data stored on the blockchain. We developed our smart contract to set the users' policies and register the client's remote devices. We will also use smart contracts in our approach to storing a trusted mapping between the authorised access's public key and its access token. The smart contract is responsible for any operations involved with the authorisation and

authentication process such as the whitelisting of all addresses that authorised the access token, generating the users' tokens, and retrieving access tokens issued for the authorised users. The smart contract will also be responsible for authenticating users by receiving signed challenges from users. It works similar to the PKI, as the client identity denotes the public part of the asymmetric key pair. This will guarantee that anyone has access to the blockchain can verify the authenticity of the message signed by the owner of the keypair.

5.5.2 Subscriber/Publisher

Our system's subscribers and publishers will be the Ethereum clients, all who have a public and private key that facilitates transactions with the Ethereum blockchain. When looking to distinguish subscribers and publisher authentication in MQTT, our system's subscribers represent clients who need to connect to the broker and subscribe to a specific topic to receive data published from the publishers on that topic. In contrast, Publishers are the resource-constrained IoT devices, mainly the sensors which need to authenticate to connect to the MQTT broker to publish their sensor reading in a specific topic. Anytime a publisher or subscriber initiates a connection with the broker, they will receive a challenge back from the smart contract. This will work as the first step towards security as the only authorised addresses can receive a challenge to sign it. The subscribers and publishers rely on the private key to sign a transaction and send the challenge back to the smart contract, enabling them to interact with the smart contract and cryptographically prove their identity. The challenge will serve as a one-time password to access to the broker.

5.5.3 The MQTT broker

The MQTT broker is similar to publishers and subscribers as all are on the Ethereum blockchain and have a public and private key. When the MQTT broker receives a connect request from a client, it will first extract the client ID, which is the public

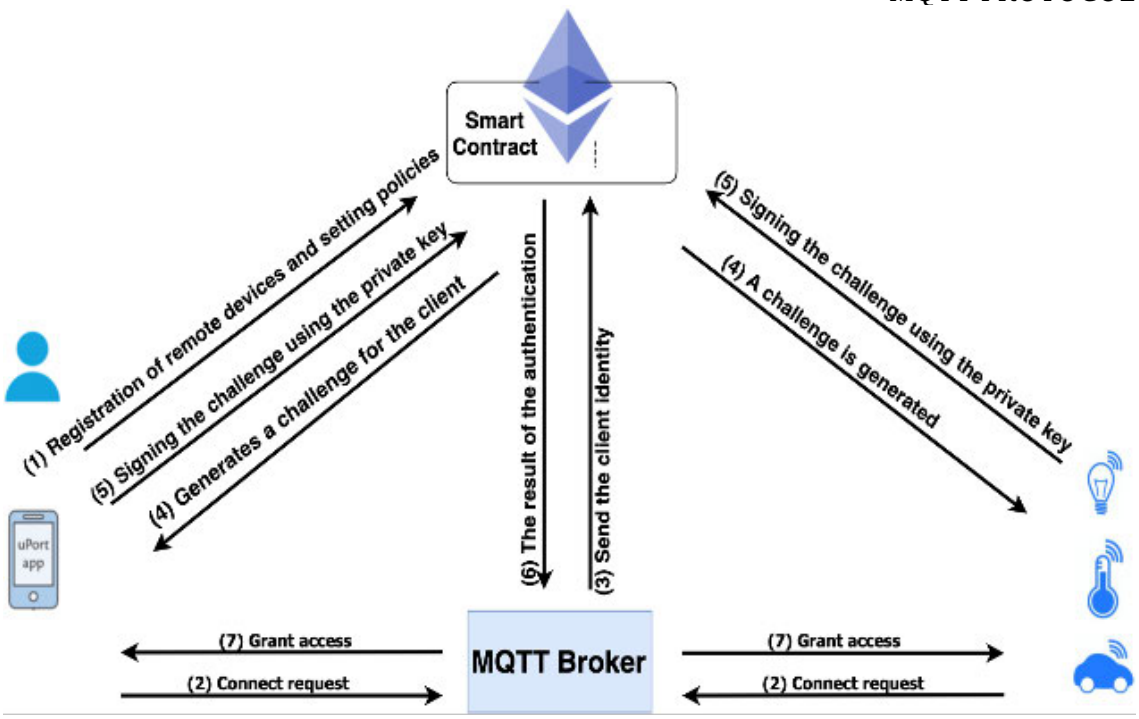

Figure 5.2: The proposed solution for authentication

part of the client key pair. Then the broker will send the client's public key to the smart contract. The smart contract will verify the user's permissions in the blockchain. If it is allowed access, the smart contract will generate a challenge and assign it to the client. Then the client signs the challenge using the private key and authenticates to the smart contract. This challenge will serve as a one-time password to access to the broker. The broker will authenticate the client through the smart contract and verify the procedure's correctness. The broker is no longer needed to reside in a specific physical location when using our MQTT-based architecture. That is, the broker can reside on the cloud or run on a specific host. A general overview of our proposed approach is illustrated in Figure 5.2 below.

1) The user first registers remote devices and set users' policies. 2) Publisher or Subscriber send a connect request to our MQTT broker using their public keys as a client ID. 3) The MQTT broker signs a transaction and sends the client ID to the smart contract to be authenticated. 4) The smart contract verifies the user's permissions and generates a challenge to be mapped to the client public key. 5) Publisher/Subscriber signs the challenge using their private key. 6) The broker verifies the procedure's correctness and connect the client.

5.6 The decentralised identity model

This section presents our decentralised identity model. For this, we proposed a self-sovereign identity model to preserve the privacy of our users. Therefore, giving them more control over their personal data rather than have it managed and stored by a third party. We utilised Uport identity (Lundkvist et al., 2017) to help with storing the identity data on the user's mobile wallet, which allows for the enhancing of MQTTs security. Our system recognises users as real people with the flexibility to express themselves fully when interacting with the smart contract compared to having them as abstract hex-encoded addresses that interact with each other.

5.6.1 Registration

A new user needs to download the Uport identity mobile app. The users will then have public/private keys, which will be stored on their mobile devices. The user's private key will be stored securely in the client's mobile device and used to sign transactions from the client's account. First, the owner is supposed to associate his/her user's identity with the Ethereum public key. For this to happen, the resource owner has to access our decentralised web application. The users need to register to our decentralised web application by clicking "Register Using Uport" on our website. The application will then redirect the user to a new QR code page. The user needs to scan the QR code using his/har Uport mobile app in order to allow the web application to receive the user's credentials from the user's mobile wallet. Then the user will be redirected to a form that needs to be completed by the resource owner. The user requires an IoT device that can communicate with the broker and the blockchain to help authenticate the owned remote IoT device. The user needs to submit the required information, such as the remote device ID (public address), the user's roles and topic name. The web application will proceed and calls functions on the smart contract that set the users' policies and add the remote IoT devices to the user's web of trust. After that, the user will receive a

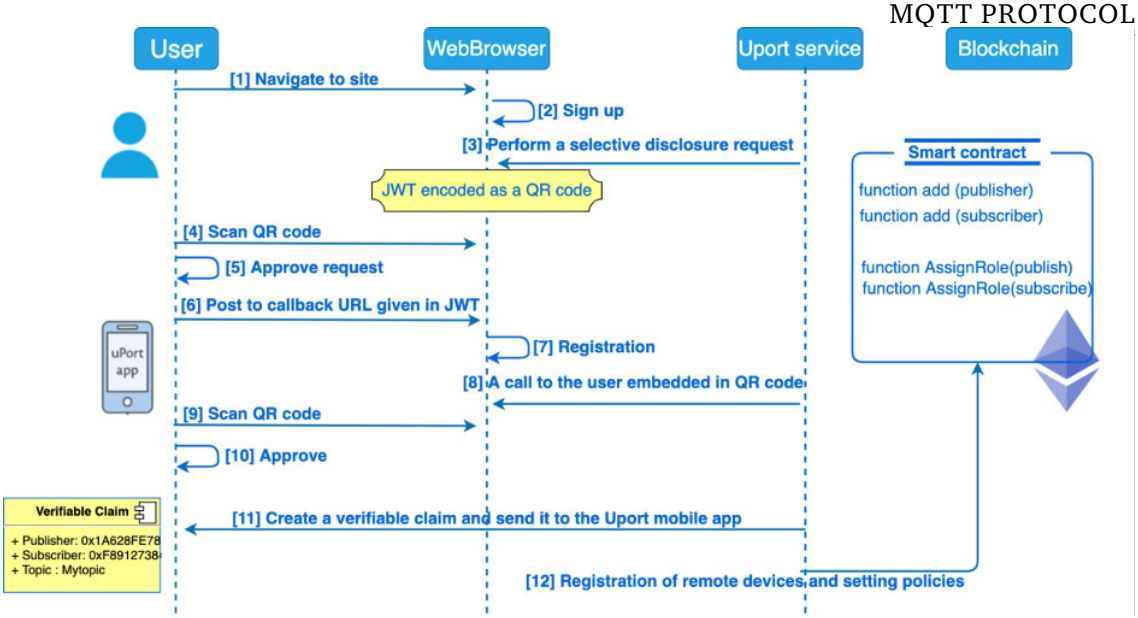

Figure 5.3: The registration model

verifiable claim in their Uport mobile app. Each claim has to have a corresponding security token, which helps with requesting the proof of the claim when a client needs to login to the web application. The registration process is shown in figure 5.3.

5.6.2 Login to the user's dashboard

The subscriber needs to login to the user's dashboard and gets authenticated through the smart contract to access the broker. When a subscriber requests access to the broker from the web interface, it first needs to click "Login with Uport" button on our website. The web application will begin by requesting a verifiable claim from the client before it requests access to the MQTT broker. The verifiable claim will be requested via a QR code, which needs to be scanned by the user's Uport mobile app. The web application will extract the client ID, topic name and the action requested (publish or subscribe) from the verifiable claim. The application will then send a connect request to the broker to request access and pass this information to the broker. The broker will then start the first step toward authentication by querying the smart contract to verify the user's permissions. This done by sending the client ID, a topic and the requested action (subscribe/publish)

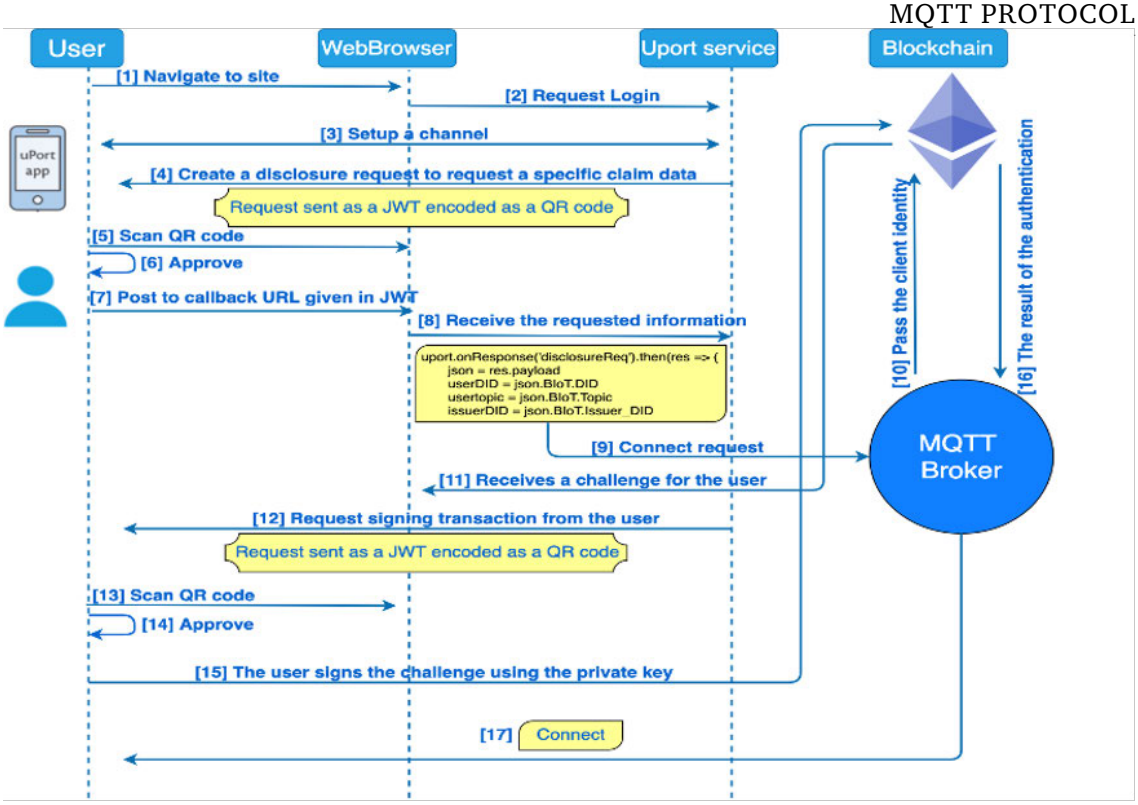

Figure 5.4: The user's login and verification

to the smart contract. The smart contract will verify the user's policies. If it is permitted, the smart contract will generate a challenge. The users can then use the Uport app on their mobile to sign the challenge using their private key. The user will get notified of the request on their Uport app and will be asked to either deny or approve. The approval needs to be confirmed using the user's fingerprint or pin code of their mobile, allowing users to present themself as real people. The final step in the authentication involves retrieving the verification result by the broker from the smart contract to establish an authorised connection with the client. The user's login and verification process is presented in Figure 5.4 below.

5.7 Implementation

This section describes the implementation of our system and explaining the technologies that were used. For the purposes of the implementation of our framework, we will rely on MQTT with the Ethereum blockchain framework. Our prototype contains three main components, the decentralised web application, a smart con-

tract for interfacing with data stored on the blockchain, and the MQTT application, which contains the broker, subscriber and publisher. We relied on Ethereum blockchain, the most prominent cryptocurrency measured by the market capitalisation. The primary impulse behind our choice of Ethereum blockchain is the amount of support that is provided due to its popularity and the ability to deploy smart contracts to their network. We implemented our smart contract in Solidity, the Turing complete language that designed to develop smart contracts in the Ethereum blockchain system. To write, evaluate and deploy the smart contract in Ethereum network, we utilised the Ethereum web browser-based IDE Remix (*Remix* n.d.). The role of the contracts in our implementation is similar to the ones of self-signed certificates. It implements any function that carries out operations in the authorisation and authentication process and generates access tokens for authorised users. To implement the MQTT broker, we adapted the open-source Mosca MQTT broker (Collina, 2013). The MQTT broker helps with data exchanging between the publishers and subscribers, or between devices and the web and mobile applications. We also built a publish/subscribe client application in JavaScript, which uses MQTT library called PAHO (*paho* n.d.). We relied on the JavaScript API to bridge the block chain's framework to the MQTT application. We then considered the need for a user dashboard to allow the subscribers to connect to the broker from the web interface. It additionally helps subscribers with reading the data detailed regarding specific topics. Our web application consists of various types of resources. These include the HTML templates, JavaScript files, CSS files and server-side implementation code. CSS and images files are used as static resources and particularly influence the display of our application. We implemented our web-server to host the website on ubuntu using apache server. The web-server is built to demonstrate how our decentralised web-based solution can allow users to communicate with blockchain to set users' policies and managing their identity to subscribe to the broker securely. The user interface communicated to the MQTT broker through a Web Socket. Therefore, the exchange of data will occur in real-time. Running the MQTT over a WebSocket to allow implementing MQTT in

the user interface. To allow the communication between our application and the Ethereum blockchain, we have relied on the web3.js Ethereum JavaScript API, to interact with an Ethereum node run on Infura. We will then use Ethereum-based Uport identity mobile application as an Ethereum user's wallet. We have provided an open-source code implementation of our project in GitHub (BC-of-Every-Thing, 2021). The project's root directory contains three main components. The first folder is the web app folder, which stores the source code for the front end and the back-end implementation of the web application, including the registration web page and the login web page. Secondly, the smart contract folder, which hosts the solidity programming code for our implemented smart contract. Finally, the MQTT folder, which contains the Broker and publishers' JavaScript implementation, which are based on the open-source Mosca MQTT broker.

5.8 Evaluation

5.8.1 Performance analysis

To evaluate our system's performance, we have implemented the MQTT broker and clients based on our proposed scheme. We compare our approach with the build-in authentication of the MQTT and the authentication over a secure TLS channel. Compared with TLS, our approach's memory utilisation is around 200 MB less than the TLS, as shown in Figure 5.5. This is a significant factor for efficient IoT applications. From our evaluation, we observed that the current TLS, which is being used widely to secure authentication on the MQTT, is consumed a higher RAM than our approach because it requires to allocate additional buffers. TLS will also cause more overhead for each MQTT message sent. However, the overhead is varied at runtime depending on the cypher suite used for the TLS connection.

Moreover, we observed that the computational overhead of the TLS scales up to 81% of the CPU, while the CPU overhead of our approach is around 24%. Our approach is around 57% less in CPU overhead compared with the TLS. Our analyses

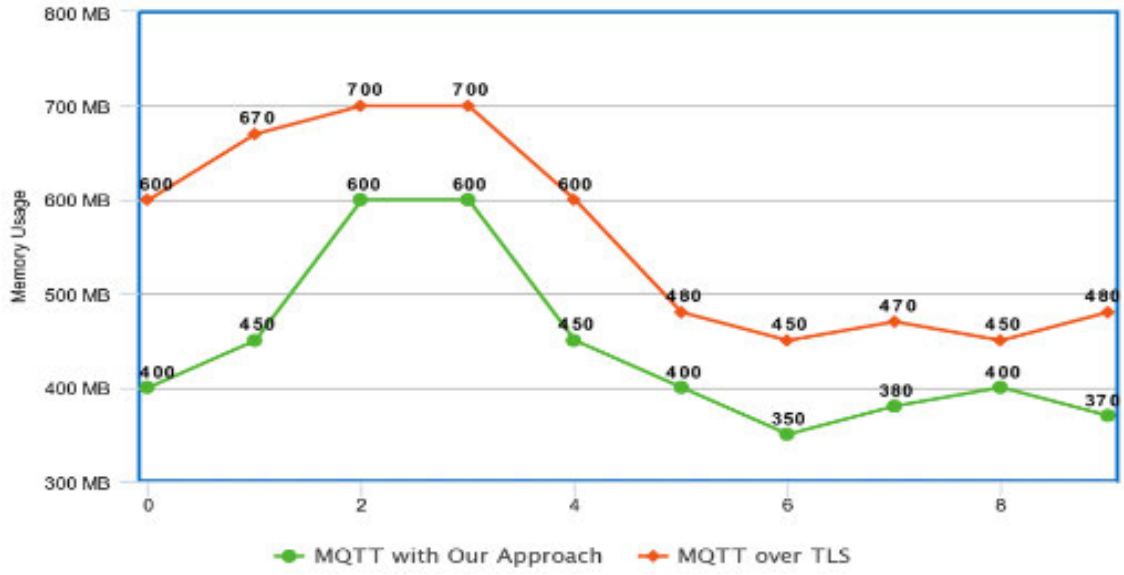

Figure 5.5: Memory Utilisation

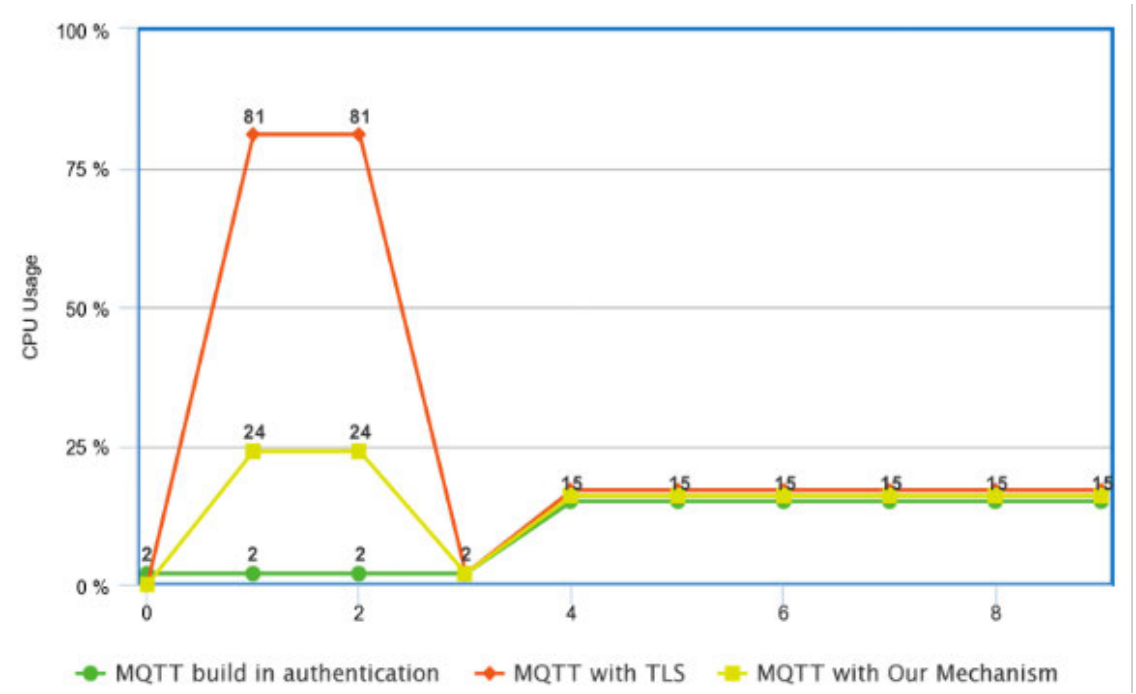

Figure 5.6: CPU Utilisation

unveiled that the initialisation of the authentication is the heaviest operation, as shown in Figure 5.6 below. TLS handshake has shown significant CPU overhead when the clients connecting to the MQTT broker, particularly with a certificate that uses a large key length. It has also shown a high drop in processor usage after the publishers and subscribers have connected.

5.8.2 Transactions cost

In this section, we are going to evaluate the gas cost of each event that happens in the system. We used the Rinkeby Ethereum test network for testing our proposed system. For reliable results, we will rely on using a public test network compared to a private one. Our result showed that the highest cost is the cost associated with the deployment of the smart contract to Ethereum network. However, this action will only be performed once when setting up the system for the first time. The cost of deploying our smart contract was around 0.000101ETH Ethers, which corresponding to $0.03732657 using the average quotation of $369.57 per Ether on October 17, 2020 (Coinmarketcap, 2022). In contrast, the cost of calling a function in the smart contract was around 0.000041ETH, which equates to $0.01515237 per transaction. Nonetheless, this cost is variant, as it depends on the time expected for sending the transaction and the storage and computational resources required. However, it is necessary to stress that our implementation is not based on the main Public Ethereum network because of the financial costs included. Instead, we built our implementation based on Rinkeby test net, which provides a blockchain testing environment with similar characteristics to the Ethereum's main public network, and without financial cost as Rinkeby provides a faucet to request free Ethers to this testing network.

5.8.3 End-to-end delay

We observe that in order for a user to complete the authentication and get authorised access to the broker, two main transactions need to be performed, which call functions in the smart contract to facilitate the authentication process. The first function is the acessToken (), which is called from the broker to authenticate the user through the smart contract. The problem with this function is that the values that need it to perform as a source of entropy to facilitate the generation of the random challenge are depending on the block mining process because it needs values such as block's timestamp and the block difficulty, which will not be

available until the block is mined. Therefore, for a user to get authenticated via a first factor and receive a challenge to be signed by the user, it needs around 13-15 second. Furthermore, it will also need another 13-15 seconds to sign a challenge to be sent back to the smart contract. Besides, other functions are defined as view functions, in that they incur no CPU overhead, delay or cost as they only read the state of the blockchain without doing any forms of modification, such as the getClientToken(), which is used by the broker to retrieve the token that submitted by the user. However, the issue of blocks being mined too slowly will not be a big problem for two-factors authentications and similar approaches. In comparison with real-world applications that depend on a third-party for implementing two-factor authentication, it has also shown a considerable delay. For instance, emails and SMS experiencing a considerable delay at the busy networks. Therefore, this would only be a problem for applications that need argent access to the public ledger. Moreover, our approach can be significantly improved by implementing it in a private permissioned blockchain. In addition to the private chain, replacing the PoW consensus mechanism with another less computational mechanism such as PoS or PoA will significantly reduce the time needed to mine the block and would also improve the transactions speed.

5.9 Security analysis

5.9.1 Compromised Website

A common threat that will affect any website is the attackers who have gained access to a website where a user has an account. Therefore, attackers might gain access to the user's password and other accounts that use the same password. In our system, we eliminated using a user password to gain access to the user's dashboard and authenticate to the MQTT broker. Instead, we rely on verifiable credentials, which refer to the assigned DID. Our approach assures a high level of confidentiality and integrity. It is backed with the issuer's DID and its cryptographic

proof to secure data transfer between users and our web application.

5.9.2 Cryptographic attack on the user's keypair

The cryptographic schema used by Ethereum is being widely accepted within the cryptocurrency systems, which based on the KECCAK-256 hash function. The probabilities of guessing a randomly produced Ethereum private key are $1/(2^{256})$, which is equal to 1 in 115 quattuorvigintillions (ABDELRAZIG ABUBAKAR et al., 2021). This is approximately around the number of atoms in the universe. We sustain the same brute-force resistance similar to other Ethereum keypair. Therefore, if this a potential attack in our users' keypairs, it would can then be possible in any other Ethereum wallet.

5.9.3 Physical access to a user's device

Our solution will guarantee strong security against an attacker who gained physical access to a user's device, as Uport client application on the user's smartphone stores the user's private key securely in an encrypted file. The application decrypts the user's private key once the user has provided their fingerprint or device pin code. However, in a situation where the attacker possesses the user's mobile device and their pin or fingerprint, the attacker will have total control over the identity linked to it. The attacker can then add new subscribers and publishers to the user net of trust. The attacker will also be able to sign a transaction from the victim account to the smart contract and get authorised access to the MQTT broker. To overcome the risks of such an attack, we proposed the use of the recovery mechanism provided by Uport identity. Uport addresses problems such as this with a delegation mechanism, which built into special contracts called the controller contract and the proxy contract. This allows Uport users to recover their identities in the event of such an attack.

5.9.4 Attacker on the network

Our system relies on Ethereum public test network, it therefore, transactions will be available for all nodes involved in the blockchain system and other users who perform a simple lookup in the blockchain. Therefore, more attention needs to be considered to ensure that attackers do not compromise the system by authenticating themself to the MQTT broker using an existing token. Our system remains resistant against such attacks as our smart contract ensures that a unique challenge is produced and mapped to a single user. We further require that a user sign this challenge using his/her private key in order to complete the process and grant access to the system.

5.10 Conclusion to this chapter

This chapter has presented a proof-of-concept design and implementation of a blockchain-based authentication and authorisation schema. The provided solution relied on Ethereum blockchain. Our solution further provided a decentralised identity model that allows users to have full control over their identity and data without the need for centralised authority to manage identity. In this chapter, we described the implementation of our approach and the specific technologies that were needed for the implementation. We have also provided analysis on the performance together with the associated costs that our system uses. We observe that our approach will provide a lightweight approach to facilitate authentication of MQTT protocol in a distributed and secure way. However, our approach has shown a high delay since the transactions require to be added to the blockchain but still within an acceptable range compared with similar two-factor authentication approaches that use a third party to maintain authentication like SMS and Emails. On the contrary, our approach performs much better in the computation and storage overhead as it has shown a negligible CPU and memory usage compared with the TLS, which allows resource-constrained IoT devices to work in parallel with other

applications without suffering from the overloading problem. It is feasible that our approach satisfies the security requirements for IoT applications and meet future demands. Overall, we hope our design provides advantages in the area of user authentication compared to current alternatives.

sixth Chapter

Blockchain-based Decentralised Authentication and Access Control mechanism for Medical Wearable Sensors

---

6.1 Introduction

Recent years have seen an increase in medical big data, which can be attributed to a paradigm shift experienced in medical data sharing induced by the growth of medical technology and the Internet of Things. The evidence of this potential has been proved during the recent covid-19 pandemic, which was characterised by the use of medical wearable devices to help with the medical data exchange between the healthcare providers and patients in a bid to contain the pandemic. However, the use of these technologies has also raised questions and concerns about security and privacy risks. To assist in resolving this issue, this paper proposes a blockchain-based access control framework for managing access to users' medical data. This is facilitated by using a smart contract on the blockchain, which allows for delegated access control and secure user authentication. This solution leverages blockchain technology's inherent autonomy and immutability to solve the existing access control challenges. We have presented the solution in the form of a medical wearable sensor prototype and a mobile app that uses the Ethereum blockchain in

a real data sharing control scenario. Based on the empirical results, the proposed solution has proven effective. It has the potential to facilitate reliable data exchange while also protecting sensitive health information against potential threats. When subjected to security analysis and evaluation, the system exhibits performance improvements in data privacy levels, high security and lightweight access control design compared to the current centralised access control models. The remainder of this chapter is organised as follows. The remainder of this chapter is organised as follows. Section 6.2 provided a background on the EoHT. Blockchain for e-health is discussed in section 6.3. The problem statement is discussed in section 6.4, and the related work is discussed in section 6.5. Then, in section 6.6, we looked at the proposed solution along with the system design in section 6.7. In section 6.8, we described the implementation of our solution. The evaluation of the proposed method is presented in section 6.9. Finally, we concluded the chapter in section 6.10.

6.2 Background

There have been reports of a dramatic rise in the number of medical patients across different parts of the globe, making it difficult for patients to access healthcare services. However, the healthcare industry has experienced significant growth and changes in E-health applications, which have been attributed to the rise of innovative technologies such as wearable medical devices and mobile cloud computing (Javaid and Khan, 2021). The rise of the Internet of Things and wearable technology has brought opportunities to help solve such challenges in the healthcare domain. Such technology is facilitated by big data analytics and cloud computing, which collect data from numerous individual devices and pool them into big health data that can be used to derive valuable insights. This data can be used by hospitals and medical institutions to link to other Electronic Health Record (EHR) Data in a bid to facilitate disease diagnoses, disease treatment, and health monitoring. This data can also be useful to insurance companies in coming up

with strategic and detailed policies guided by individual characteristics, which will be more beneficial to customers since they will get to choose insurance plans that fit their medical needs (Tariq et al., 2020). The availability of wearable sensors and mobile devices have enabled patients to handle their health data at home and share it with a healthcare provider, facilitating timely medical access and support from healthcare personnel. With the Internet of Health Things in place, healthcare providers can monitor their patients and offer them care remotely, which helps in healthcare delivery and is also economically beneficial to patients. The Internet of Health Things also allows the tracking of patient health by healthcare providers, who can, in turn, advise the patients and offer them the required medical services. However, there are cases where patients will be unable to track and manage their health records shared with the healthcare provider or even find it default to do so.

6.2.1 Blockchain and e-health

The advanced blockchain technology developments were first applied in crypto-currencies such as Bitcoin (Nakamoto, 2008) and Ethereum (Buterin et al., 2014). While blockchain technology remains significant in cryptocurrencies, it can also be used on any application requiring secure authentication, such as IoHT. This is because blockchain technology comes with a secure cryptographic technique, which can be used to identify and authenticate systems and users, thus facilitating access control in a secure, distributed, and scalable manner. Using blockchain in such a system is critical for data control. E-health can rely on new security features of the blockchain-based access control, which are more advantageous compared to traditional access control solutions. One such advantage is creating immutable ledgers containing transactions to be used in a data-sharing system, thereby guaranteeing high system integrity and trustworthiness. Therefore, once a transaction has been recorded, it cannot be altered or modified by anyone since blockchain only records transactions and does not permit recovery actions to its records (Tariq et al., 2020). Blockchain-based smart contracts have also proved

critical in user verification and authentication (Ghaffari et al., 2020). The strict access control policies in smart contracts are essential in the authorisation of users and in detecting and preventing potential threats to IoHT systems. Finally, the use of smart contract technology with blockchain eliminates the need for a central server, therefore maintaining fairness among transacting parties. Since all the smart contracts in the blockchain are public, all connected users will have a copy of the smart contracts, getting equal rights to exercise control over contract operations (Ghaffari et al., 2020).

6.3 Problem Statement

A secure data sharing infrastructure is needed to handle the sharing of health data between institutions. However, this is marred by several changes regarding interoperability, security, and privacy. Health data is categorised as highly privacy-sensitive and storing it in a public cloud increases the risk of unauthorised access and exposure. The current use of a centralised architecture in healthcare requires a centralised trust for it to function properly. There is also the challenging task of effective health data integration and healthcare systems operability, in addition to users having little to no say regarding data collected on their health. To help achieve self-sovereignty and increase the adoption of wearable devices and mobile platforms, there is a need for improved versions of IoHT systems that protect user privacy and provide user-centric access control. Operating on a central authority has its own share of risks, such as single point failure, which is often solved by using third parties to provide data backups, effectively increasing the risk of exposure (Ometov et al., 2021). This necessitates the need to develop efficient access control solutions for medical data sharing. On the other hand, there is considerable potential for blockchain and decentralised technologies, which exclude the use of a third party to manage the trust. Decentralisation of trust is increasingly becoming a dominant trend, creating opportunities to manage authentication and authorisation in a decentralised and autonomous manner. Therefore, blockchain technology can

revolutionise healthcare applications by providing promising solutions that promote healthcare security and performance. Through this research, we have worked on designing a secure access control framework for medical wearable devices.

6.4 Related work

The global Electronic Health Records (EHR) market is predicted to expand from around 30 billion in the year 2020 to 40 billion by 2025. However, the security of the EHR remains a major challenge in managing EHR data (Al-Sarawi et al., 2020). A significant of current researches are focused on security and privacy in the IoHT. For instance, the literature in (Newaz et al., 2020)] (Xue, 2019) (Nahapetian, 2016) analysed the side-channel attacks that may be influenced by devices such as wearable medical devices and smartwatches. Similarly, the authors in (Zhang et al., 2014) (Nguyen et al., 2016) discussed the security issues of key negotiation, data encryption and integrity during transmission. There have been several efforts (Wu et al., 2017) (Joshitta and Arockiam, 2017) (Diez et al., 2019) to address these security issues by proposing access control mechanisms for IoHT. However, the issue with these solutions is that they are centralised. The authentication data needs to be stored in a centralised local server, prone to a single point of failure. In addition, when encryption is used for authentication, some complex encryption algorithms will also bring some problems, such as low computational efficiency, increasing hardware power consumption, etc. Lately, several research efforts were geared towards finding solutions to the key challenges of access control on IoHT by proposing integrating blockchain technology with IoHT to meet the IoHT security needs. For instance, the work presented in (Hammi et al., 2018) investigated the blockchain applicability to overcome various security issues in IoT and proposed a blockchain-based authentication mechanism. The main issue with the proposed approach is that one system's devices cannot interact with the devices in other systems. Hence, it is not suitable for many distributed IoT applications that require interaction with devices in different systems. Another approach was highlighted

in (Guo et al., 2019), which proposes a hybrid architecture that provides decentralised access control to the e-heath data by utilising both blockchain and edge nodes. The proposed architecture uses blockchain to manage users' identities and access control policies. The e-health data is stored on the off-chain edge nodes and implement policies defined in Abbreviated Language For Authorisation (ALFA) to apply attribute-based access control that relies on the blockchain-based access control logs. The proposed system was implemented using Hyperledger Fabric blockchain. Similarly, the authors in (Yue et al., 2016) proposed an approach to allow patients to grant access to their health information stored on a blockchain to assigned individuals. In the previously proposed blockchain-based approaches, the users' health information is stored on the blockchain. However, the blockchain's ability to securely keep data has been questioned because of its complete transparency, which is in conflict with the concept of confidentiality. Moreover, a key feature of the blockchain is its ability to maintain data integrity while also making it almost impossible to alter. Nevertheless, this feature may have both advantages and disadvantages. The reason for this is that errors in data that are submitted to the distributed ledger cannot be fixed, and in the event that individuals need to delete their personal health data, they are unable to do so. In addition, solving the privacy issue by placing the entire health records into a private blockchain would considerably enlarge the entire chain size, demanding more storage at each node.

Our proposed solution differs from other existing solutions in that it does not record users' sensitive data on blockchain. However, blockchain is used in our proposed method to provide decentralised authentication and access control for wearable medical devices and its data. The smart contract in our system use to manage user access. Furthermore, unlike the previously stated techniques, our solution provides a decentralised identification system and introduces the concept of self-sovereign identity. Furthermore, our system enabled decentralised identification for IoT sensor devices, and the smart contract is used to manage the identities of users and devices in a decentralised manner.

The main contributions in this chapter can be summarised as follows. First,

proposed Blockchain-based authentication and authorisation mechanism for medical wearable devices. Second, the development of decentralised access control and data access delegation approach for sensitive medical data propagated from the users' health wearable devices. In addition, demonstrated a proof-of-concept implementation of the proposed solution along with performance evaluation and security analysis, which obviously proves the viability of our system to satisfy the IoT security requirements.

6.5 The proposed solution

This study aims to adopt blockchain technology to design a secure framework for medical wearable devices. To solve privacy and security challenges in IoHT systems, we propose a blockchain-based decentralised identity system and privacy-preserving data access control method to facilitate data sharing with the authorised people. This will ensure that users have full control over their personal data instead of having it stored and managed by a third party. The system's architecture will adopt a public blockchain structure. The proposed solution exhibits the effectiveness of a blockchain-based access control mechanism, which allows for delegated access permissions facilitated through a distributed ledger for information, services, and any devices within the Internet of Health Things systems. To achieve secure data sharing, users will have access permissions managed via a smart contract. This means that the resource owner can set access rules and eliminate the need for a third party to manage the data and determine who can access the data. To ensure users' health information security and privacy, it will be stored on a database hosted on a trusted platform rather than on the blockchain. The use of blockchain technology in this system remains for delegated access control and secure user authentication. The proposed solution can be applied to various IoT applications since it is suited to meet specific IoT requirements regarding defence against attacks, lightweight, distributed nature, and scalability while also meeting the CIA (Confidentiality, Integrity, and Availability) security triad requirements.

6.6 System design

The proposed blockchain-based IoT platform will comprise four layers, as depicted in Figure 6.1 : the users' application layer, the healthcare IoT sensors layer, the connectivity layer, and the blockchain service layer.

6.6.1 Users' application layer

The application layer will provide a user interface for presenting data collected from wearable devices into the users' application interface. Wearable devices, such as body temperature sensors and ECG, will collect health data from the users and then upload them to a secure database hosted by our application. Once the data is recorded, the users will remain the sole owner of their health data. Additionally, this layer allows users and resource owners to manage their accounts and set users' policies by communicating with the smart contracts on the blockchain. Only the owner can grant, revoke or deny access to data to third parties such as medical personnel. The users can grant the health providers access to their data when seeking medical treatment or record data relating to a certain treatment to help the professional monitor their health and improve or adjust treatment.

6.6.2 Healthcare IoT sensors layer

This layer is made up of different medical wearable devices that are capable of communicating, storing data, and computing. Wearable devices help transform original health data into an understandable format and sync it to an online account. A user account can have either a single wearable device or multiple wearable medical devices.

6.6.3 The connectivity layer

The connectivity layer offers routing management and is responsible for network management, message brokers and security management, etc. The communication

is made through the Message Queuing Telemetry Transport (MQTT) protocol (Standard, 2014) which allows the transmission of sensors data from the wearable device to the users' application layer. MQTT is a publish/subscribe protocol where the broker has to push information to the client, as opposed to the request/response model that is currently used in the HTTP protocol, which requires a client to request data. Once the medical sensors layer collects raw data, it is sent to a master MQTT broker, which it works as a messaging broker to aggregate and distributes the data to the resource owner or any other individual granted access by the data owner (Standard, 2014).

6.6.4 The blockchain layer

We use an Ethereum-based blockchain on the proposed model to help store information in a distributed manner. The reason for using an Ethereum-based blockchain is because it allows the deployment of smart contracts in their blockchain system. Smart contracts will be responsible for any authentication and authorisation processes. Since using smart contracts would make interfacing data stored on the blockchain possible, we developed them in such a way that they register the client's remote devices and can set the users' policies. The smart contracts used in the system will also store a trusted mapping between a public key authorised by the user and its access token. The client's identity will have the public part of the asymmetric key pair and will be able to verify that the user accessing the blockchain is using an authentic message signed by the key pair's owner.

(1) Send connect request to the broker. (2) The broker sends the user ID to the smart contract for authentication. (3) Smart contract generates a random value (challenge) (4) the users receive the random value and sign transaction using the user's private key to be sent to the smart contract. (5) The smart contract sends back the result of the process to the broker (6) The broker will then grant the user access and connect the users.

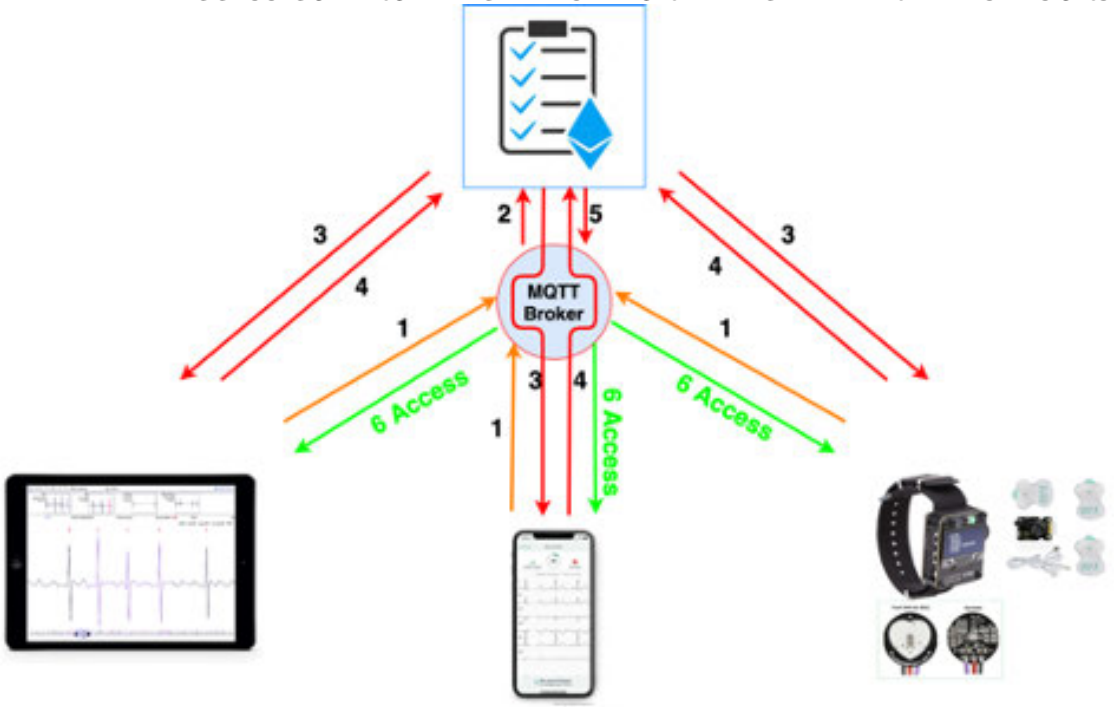

Figure 6.1: The proposed system design

6.6.5 System Entities

The proposed system is comprised of a number of entities such as clients, an MQTT messaging server, and a smart contract. The clients include health wearable devices, healthcare providers, and resource owners.

6.6.5.1 Resource owner

The system's resource owners are the Ethereum clients, who have both a public and private key. Users issue transactions signed by their private key, whose hash is taken to be the user's address and associated with their access token. Before using the system, the users will be required to set user policies, assign users to specific topics and register their remote devices. After registration, the users will be able to control access to their health data. This will be achieved through the mobile app, which can share and communicate user data securely to the smart contract. Our approach does not require the data to pass through a central authority as users can use their device storage to store their credentials.

6.6.5.2 MQTT messaging server

The MQTT broker is responsible for providing sensor readings and processing user requests. The broker is similar to other users in the system since they are all on the Ethereum blockchain and have both private and public keys. Anytime a connect request is sent by a client, the MQTT broker will extract the public part of the key pair, which is the client's ID. This information will then be sent to the smart contract for verification. Once access is allowed, the smart contract generates a challenge to serve as a one-time password for authenticating clients. The clients, in turn, sign the challenge with their private key. The MQTT broker will then validate the procedure's correctness and grant or deny access. The MQTT broker can either run on a specific host or reside on the cloud.

6.6.5.3 Smart contract

The proposed model exploits the advantages of smart contracts and blockchain technologies in performing delegated authorisation in IoT systems. Smart contracts help to store an immutable record of both user policies and authorisation information. They also offer resilience by executing smart contract code across all blockchain nodes. The smart contract will be used to implement policies such as on-chain access control decisions. The smart contracts will help with issuing tokens used by both publishers and subscribers to authenticate to the MQTT broker. Using smart contracts ensures that users do not have to store their tokens locally as it helps generate access tokens and manage their authentication process. The integrity and authenticity of the access token, which serves as an OTP (one-time-password), can be verified by a simple lookup in the distributed ledger. The web3 JS library, which uses the RPC to interact with the smart contract, will be used to access all the services written in the smart contract.

6.6.5.4 Healthcare providers

The user will appoint the healthcare providers, such as doctors, to provide medical treatment or suggestions. The system will allow the healthcare provider to upload the user's medical treatment data to the medical health record for the purpose of sharing it with other medical professionals but only after obtaining the patient's permission. The health provider can also request access to the user's health data from a wearable device or medical treatment data from the record through the proposed system. The smart contracts help grant access to every data request and access on the blockchain by implementing policies.

6.6.6 Systems Interactions and Information Exchange

Interactions between the entities in the system are also known as transactions and have to be validated before they can be confirmed. The interaction occurs in two phases, which include:

6.6.6.1 The registration phase

For the registration, the user starts by downloading a mobile app on their smartphone phone. The user needs to create an Ethereum account and get private and public keys. The private key will be used to sign transactions from the user's account and will be stored on the user's device. The users' public keys will be associated with their identities. For the user to send the required information (topic name, user's role, and the remote device ID), needs to submit transactions to the smart contract. The mobile app will then rely on the functionalities of smart contracts to set policies and add the IoT devices to the user's list of trusted devices.

6.6.6.2 The Authentication phase

Anytime a client uses the client mobile app to request access to the broker, the application will start the authentication process by sending a connect request to the broker and pass the client ID. Besides, the broker verifies the users' permissions

by sending the user's ID to the smart contract to verify the user's permissions. Once the request is permitted, the smart contract will generate a challenge, which will require the user to sign it using the private key. The user will then get a notification on the app to either approve or deny the signing request. Once the broker retrieves the verification result from the contract, an authorised connection with the client is created.

6.7 Implementation

This section explains the technologies used and the implementation process of our system. The study used a case study involving a patient who is being remotely monitored by a doctor through medical devices such as body temperature and ECG sensors. The prototype is made up of four components, including the messaging communication system, the wearable medical devices, a smart contract to help interface data on the blockchain, and the client application.

6.7.1 The blockchain implementation environment

We implemented the Ethereum blockchain smart contracts as a proof of concept. The choice of Ethereum blockchain is based on its ability to deploy smart contracts and the support that comes with its popularity. To implement our smart contracts, we used Solidity, a Turing complete language that helps develop smart contracts in the Ethereum blockchain systems. We then used the Ethereum-based IDE Remix to write, evaluate, and deploy smart contracts across the Ethereum network. The IDE also comes with a compiler that can be used to test the functionality of smart contracts.

6.7.2 The communication protocol

The MQTT protocol will be the main communication protocol used to facilitate data transfer from the IoT devices to the user application. We utilised the open-

source Mosca MQTT broker (Collina, 2013) to help with the implementation of the MQTT server as it helps with data exchange between subscribers and publishers or between a mobile application and a wearable medical device. To help bridge the blockchain framework to the MQTT application for the authorisation and authentication of clients, we used a JavaScript API.

6.7.3 The users client app

We will use a web socket to establish communication between the user's mobile app to the MQTT broker and facilitate real-time data exchange. Our application will help manage the user interface, managing user profiles, and communicate with the smart contracts on the blockchain. We build a native mobile app using JavaScrip. To make communication between the Ethereum blockchain and our application possible, the proposed system uses a web3.js Ethereum JavaScript API that interacts with an Ethereum node run on Infura.

6.7.4 The wearable device

To build the medical wearable device, we utilised an ESP32 DevkitC v4 board. We also added an AD8232 ECG sensor to help with ECG monitoring. We then used the Zerynhth studio, which offered a platform we could use to program microcontrollers using C and Python programming languages and offers an open-source Python library that can interact with smart contracts and the Ethereum blockchain. The Zerynth Ethereum library uses the JSON-RPC interface to send transactions and interact with Ethereum nodes. This makes it possible to make transactions and fetch status information. This library offers access to two companion classes that help with building a higher-level interface, and they include transaction and contract. The contract class can help call smart contracts and their methods, while the transaction classes help develop a correctly signed transaction that is ready to be sent.

6.8 Evaluation

6.8.1 Security analysis

This section will highlight how our proposed system will resist the potential security threats for an IoT system. Medical data sharing systems face great security concerns regarding the protection of sensitive patient information from potential security threats and attacks. We will start by evaluating the security margin of our model when faced with different threats.

6.8.1.1 Man in the Middle attack (MITM)

The proposed model will utilise the cryptographic signature and random challenges to help prevent potential MITM attacks. The system will remain safe from such attacks since the smart contracts produce only one unique challenge, which can be mapped to a single user. Additionally, the model will also require users to use their private key when signing the challenge to access the system.

6.8.1.2 Sybil Attack

To help protect the model from a Sybil attack, each device will be assigned a unique identifier stored on the blockchain. Each entity connected to the model will have a single key pair at any particular time. The private key will only be known by its user while the public key remains visible to all entities signed into the blockchain network. This means that the adversary cannot access a private key, which is used to sign the transaction for data access. Without the private key, adversaries cannot forge a user's signature. Additionally, all invalidated transactions are removed from the blockchain network, making our system resistant to external attacks.

6.8.1.3 Denial of service (DOS) Attack

During a DoS attack, the attacker makes it difficult for an authentic user to access the service in the network by increasing traffic or launching fraudulent transactions.

However, due to the decentralised nature of blockchain, our model becomes resistant to DoS and DDoS-related attacks. The large number of mining nodes in Ethereum makes it highly resistant to DDoS attacks. Therefore, our model ensures high data availability for authentication data that is saved on the Blockchain. Even if a node fails or becomes unreachable, the network as a whole will continue to operate. As a result, our system will continue to operate at a high level of availability.

6.8.2 Security of our system

Every model comes with three main security requirements that model designers need to address, and they include confidentiality, integrity and availability. In this section, we analysed the security of our system based on these requirements:

6.8.2.1 Tamper-proof

The system does not allow the modification of the users' policies, access rolls or credentials as they remain immutable. Due to the chronologically nested blocks, which contain a hash of the previous block and the current timestamp, transactions remain immutable unless one individual takes over 51% of the network's computational power. The blockchain records every access request and access activity, meaning that any changes made to the data can be audited and tracked. Additionally, each transaction has a digital signature that provides the non-repudiation, integrity, and authentication of each transaction.

6.8.2.2 Privacy preservation

The users' credentials remain highly sensitive and should not be disclosed to any third party without approval from the patient. To tackle the issue of privacy, we ensured that all transactions made in the blockchain remain secure by utilising a decentralised digital identity, where every user connecting to the blockchain gets a unique account with a random public key. The decentralised identity allows users

to own their credentials data and manage their own identities rather than have it managed by a third party.

6.8.2.3 Confidentiality

To help maintain confidentiality, the system will make use of asymmetric encryption technology to ensure that every access to the system is done only after authorisation. The authorisation will be a randomly generated unique token that will have to be signed by a user's private key to authenticate the client to the messaging server.

6.8.2.4 Availability

Our model ensures that data associated with verification and authorisation processes that are stored on the Blockchain is always available. On a per-node basis, each node replicates and updates transaction data. This means that the network will continue to function even if a node leaves it accidentally or maliciously or otherwise the node becomes inaccessible. As a result, our system will continue to operate at a high level of availability and reliability.

6.8.3 Performance analysis

We present a set of experiments to characterise the impact of our approach on wearables medical devices. Our measurements setup depends on an ESP32 with a 240MHz Dual Core CPU WROOM-32D and 4MB flash. We analysed our approach in comparison to the current security mechanisms. For this, we adopted the TLS protocol, which supports different cryptographic algorithms. From the evaluation, we observed that the current TLS utilises a higher memory than our mechanism because it needs to allocate additional buffers. To establish a TLS session, enough free heap memory is required. A single TLS session requires around 40KB of additional heap memory. Moreover, compared to our approach, the CPU processing overhead when using TLS is also higher since cryptographic operations are involved, especially with a certificate that uses a large key length, as shown in Figure 6.2.

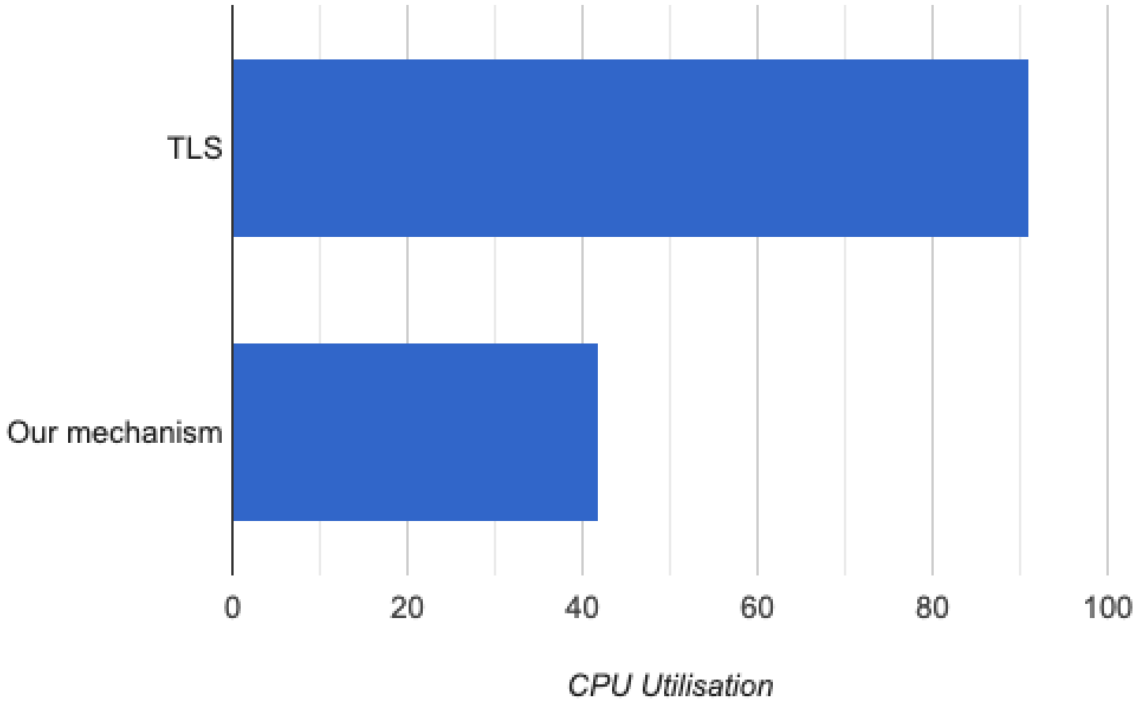

Figure 6.2: The CPU overheads of our blockchain transactions vs TLS

In addition, using TLS on the ESP32 requires a significant amount of energy. Our approach is beneficial in terms of energy consumption and significantly reduces the required energy for IoT devices.

To measure the overall time of establishing a secure connection, we utilised the internal time function of the micro-controller. As shown in Figure 6.3, our approach has shown a higher execution time overhead. This is an expected issue when adopting the public Ethereum blockchain. This is due to the time needs it to finalise the transactions, which is around 13 seconds on average. However, this is a well-known issue with other realâworld applications that depend on a third party to maintain trust. Still, this can be significantly enhanced by adopting the private blockchain or utilising a reduced mining consensus that can provide immediate block finality and reduce the time needed to finalise the transactions. Moreover, any action in the Ethereum blockchain requires a certain amount of gas to send transactions and interact with the smart contract. Our result revealed that smart contract deployment is the highest cost, as shown in Figure 6.4. Nonetheless, this cost is variant, as it depends on different parameters, such as the transaction

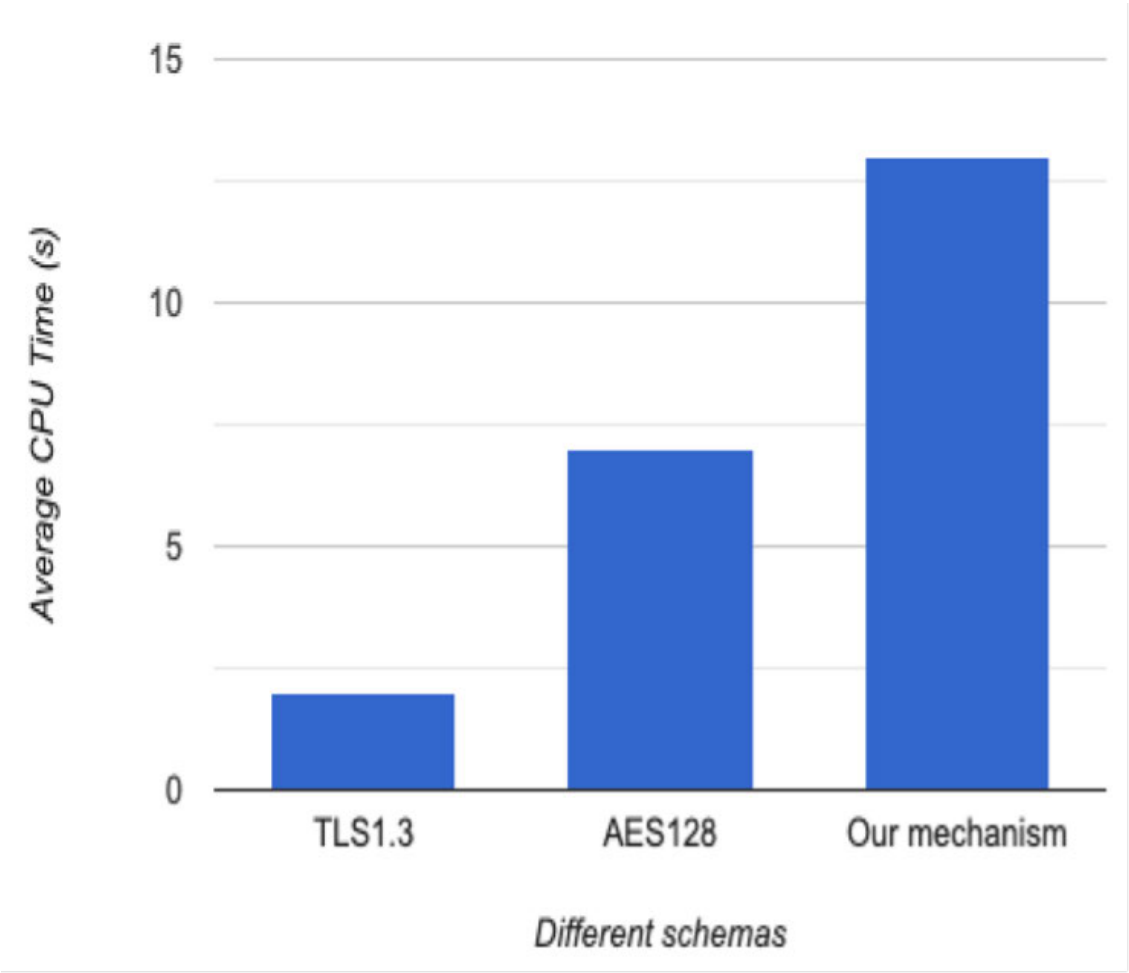

Figure 6.3: End to end delay

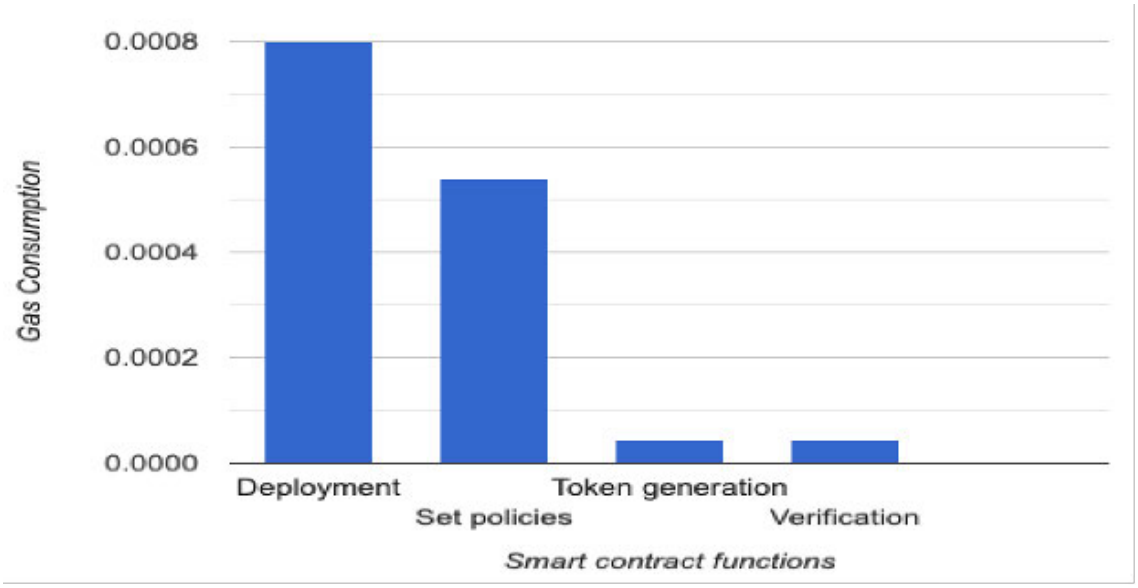

Figure 6.4: The gas cost of each event that happens in the system

speed and the storage required. However, it is necessary to emphasise that our implementation is based on the Rinkeby testing environment, which requires no cost as Ethereum provides free Ethers to this testing network.

6.9 Conclusion to this chapter

In this chapter, a blockchain-based authentication and access control mechanism is proposed for wearable medical devices. The chapter presented a proof-of-concept design and implementation of a lightweight and secure framework for medical wearable devices. The proposed approach helps to solve privacy and security challenges across wearable medical devices. It provides a privacy-preserving access control mechanism and facilitates secure users' authentication. This allows the system's users to have full control over their credentials rather than maintaining them by a third party. We have presented the solution in the form of a medical wearable sensor prototype and a mobile app that uses the Ethereum blockchain in a real data sharing control scenario. The security of our system and an attacks model have been analysed to evaluate the ability of our system to meet the security requirements of the IoT systems. Our solution showed enhancements in security and users' privacy compared to the current centralised models. In addition, we provided an analysis of the performance and the associated transactions costs. We observe that our approach provides negligible memory and CPU usage compared with the current TLS and prove suitable for resource-constrained devices. It is feasible that our approach satisfies the security requirements for IoT applications and meet future demands. We hope that our approach improves the privacy and security of users' sensitive health data. On the other hand, the proposed approach shows a significant delay due to the characteristics of the public blockchain since transactions need to be appended to the block but are still within an acceptable range.

seventh Chapter

PoNW: A Secure and Scalable Proof-of-Notarized-Work Based Consensus Mechanism

---

7.1 Introduction

Through this thesis, we have demonstrated that blockchain has the ability to overcome the security and privacy challenges associated with IoT authentication. However, the integration of IoT and blockchain introduces new blockchain-specific difficulties that must be addressed before this integration can be realised. The consensus procedures used in the blockchain system play a vital role in the performance of the blockchain network. This is due to the fact that the core of blockchain technology is distributed computing, as well as the collaboration mechanism of group trust within the distributed computing environment, which may answer the IoT's collaboration ability, trust relationship, and security protection concerns.

Despite the benefits of decentralised trust relationship management over centralised models, there are still questions about its scalability, efficiency, and safeness. Such existing blockchain systems as Bitcoin and Ethereum rely on open consensus protocols like Proof-of-Work (PoW), which has been criticised for its high energy usage. This is because reaching consensus over the ledger state requires synchron-

ous communication between all nodes in the network. The approach has also been demonstrated to have significant performance and throughput constraints. This is because PoW consensus algorithms are predicated on the idea that a node is less likely to launch an attack on the network if it contributes heavily to keep it running. In particular, the solution given by PoW-based blockchains makes Sybil attacks harder to undertake by having miners carry out computationally intensive operations that, in theory, cannot be carried out by a single person.

Commonly called as "mining," the work entails doing many calculations until a solution is found. Finding a nonce, or random number, that causes the SHA-256 hash of the block header to start with a certain number of zeros. As a result, miners must provide evidence that they have put in the required effort to find a solution. When the issue has been resolved, it is simple for other nodes to check if the solution is correct. The blockchain's inefficiency in throughput, scalability, and energy consumption stems from the mining process, which is not ideal for an IoT infrastructure. In addition, solving the Byzantine General issue is the foundation of the alternative consensus protocols currently in use, which have been shown to boost performance significantly. To add each block, one user must be chosen. In the meantime, other nodes in the blockchain network can verify the block's validity. The difficulty with randomly performing this pick is that it leaves the system vulnerable to security breaches.

In trying to rectify this issue, this chapter proposes the introduction of a new hybrid consensus protocol known as the Proof of Notarized Work (PoNW). The new proposed protocol addresses the issues related to performance and throughput by developing a secure and scalable consensus model that can preserve the security characteristics of the PoW consensus protocol. When this hypered model is paired with a Byzantine Fault Tolerance (PBFT) verification, the system gains the ability to replace the probabilistic finality in current PoW with absolute finality in a matter of seconds, solving the issue of scalability. The PoNW concept reduces the number of nodes that need to achieve consensus, thereby reducing the overall energy

consumption in the current PoW. In addition, we propose using a decentralized random beacon to select nodes to participate in the mining process randomly. Therefore, our algorithm promises to achieve higher scalability and consistency levels without conceding its decentralization. Thus, shown more potential in terms of increased capability of an optimized decentralization, efficiency, practicality, and security. Finally, the study will look into the proposed algorithm's security and provides threats model to insure an acceptable failure probability. Results from the security analysis have shown that our consensus algorithm ensures forks cannot occur, and it remains secure and consistent even amid numerous attacks.

7.2 Blockchain consensus

It was in 1993 when M. Naor and C. Dwork introduced the Proof of Work concept (Dwork and Naor, 1992), which would later be applied on a larger scale by Satoshi to allow a distributed and trust-less consensus, at the advent of his Bitcoin Crypto-currency in 2008 (Nakamoto, 2008). A major benefit of the PoW consensus protocol is the presence of a robust algorithm that can ward off malicious participants. The concept has proved to work under being put under various tests in real-word scenarios and remains the foundation of cryptocurrencies such as Ethereum, Bit-coin, and several other blockchain applications. All the transactions taking place in a PoW based consensus is recorded, verified, and broadcasted among all the participants existing in the decentralized peer-to-peer network. In doing so, the process makes the whole system resistant, stable, and immutable. However, for this to happen, there is a need for half of the computing resources to uphold honesty. While a security property requires an honest majority to work, this can be very costly in terms of scalability, as all the participants need to be kept in the loop of what is happening and agree implicitly (Nakamoto, 2008). The rapid evolution that blockchain technologies have undergone has resulted in a growing demand for increased quality of services provided by them. This, in turn, has led to the meteoric rise of key challenges that arise during the design phase of

blockchain protocols, particularly because the performance posted by the adopted consensus mechanisms will be a significant deciding factor of the blockchain network's performance in terms of network scalability, robustness to arbitrarily behaving nodes, speed of consensus finality and data consistency, etc (Croman et al., 2016). The performance of the first generation of blockchain consensus protocols was limited by two factors; transaction throughput and the confirmation latency, which is a result of the consensus used in the blockchains that require synchronous communication for the blocks to be persistent. Therefore, clients have to wait for up to ten minutes before a transaction can be confirmed in Bitcoin and around 15 seconds in Ethereum (Xiao et al., 2020). The second generation of blockchains has later emerged as a solution to the challenges faced by the first generations of blockchain. This second-generation resulted in using the traditional Byzantine consensus algorithms, which allow for an immediate strong consistency. Since then, there has been an emergence of algorithm alternatives to PoW, such as the Delegated Proof of State (DPoS) (Larimer, 2014) and the proof of stake (PoS) (King and Nadal, 2012). Other alternatives, such as IOTA (Popov, 2016), propose replacing a blockchain data structure with a Directed Acyclic Graph (DAG) data structure. However, the previously proposed approaches cannot provide a considerable throughput improvement without first is conceding with regard to other significant factors (Xiao et al., 2020). These include security and decentralization since most of the proposed approaches can guarantee maximum performance in an environment where a participant's behaviour is expected. Although the current blockchain systems that relay on Byzantine consensus mechanisms can guarantee stronger consistency in a short time, it does not scale well for a large number of nodes (Poelstra et al., 2014).

7.2.1 Sharding

With previous generations suffering from the issue of scalability, the architecture of the third blockchain generation was geared towards solving this. This generation

of blockchain proposed the use of sharding, which is a prominent approach used to overcome the throughput and scalability limitations present in existing blockchain systems (Dang et al., 2019). Sharding uses a variety of different methods to assign blockchain nodes to different groups (shards). Nodes that belong to the same shard form a committee and work in parallel to achieve consensus. As a result, this allows blockchain systems to scale to larger networks. Although sharded blockchains proving more potential compared to the traditional BFT, there was still a need to ensure the per-subchain consensus protocol runs across hundreds of participating in adversarial environments (Dang et al., 2019). As the number of nodes achieving the consensus is minimized, the probability of an adversary being able to abort the system becomes higher. This, therefore, shows that one cannot avoid the scalability requirement of BFT consensus by simply changing the architecture.

7.2.2 Problem identification

From the above, it can be noted that there does not exist a single consensus protocol able to provide all the scalability, consistency, and decentralization properties (Ismail and Materwala, 2019). Systems based on a PoW consensus architecture fail to guarantee immediate finality due to its major scalability issue. While these systems can prevent arbitrary changes to the state by using validation, it allows for the creation of two or more valid continuations forking. Additionally, there have been known cases where the participants place preference in their own state for such purposes as performing a double-spend attack or earn a block mining reward (Lin and Liao, 2017). In the same way, DPoS faces the challenge of decreased decentralization while the PoS consistency is challenged by the Nothing-at-Stake problem. It can also be noted that PBFT experiences massive network scalability problems, forcing it only to be used for consortium chains. Therefore, in this work, we are looking to address the previously mentioned issues by developing a secure and scalable consensus mechanism that can preserve the security characteristics of the PoW consensus protocol, while also improving its scalability, and reducing

its energy consumption. When making a comparison between pure consensus protocols and hybrid consensus protocols such as ours, the hybrid consensus protocols have shown more potential in terms of increased capability of an optimized decentralization, efficiency, practicality, and security.

Through this study, we have worked on developing a blockchain-based consensus protocol model, in addition to its system design and the required set of data structures. In doing so, we aim to formally study its implementation features, security-related primitives, and characteristics, which are crucial in solving the following key challenges: energy consumption, probabilistic confirmation time, scalability, and decentralization. In addition, we provided the construction of a new hybrid consensus algorithm that strikes a balance between the PBFT and PoW consensus mechanisms. We proposed a secure random model to select participants to perform PoW to stop an adversary from concentrating its presence in one committee and exceeding the byzantine tolerance threshold. Furthermore, we proposed a ranking mechanism to resolve chain fork, which is based on the Pseudo-Random Process along with a permutation function to arrange selected committee members into sequential order. Finally, we provided security analysis of the model together with a threat model which ensures a certain acceptable probability of failure.

7.3 The proposed consensus protocol

This section describes the proposed PoNW algorithm, which provides an energy-efficient protocol that is very robust and can solve issues of scalability and is suitable for permissioned and open blockchain. However, in this model, we propose our algorithm for permissioned blockchain models that controlled by a single federation or entity as this can be useful for a blockchain system whose applications revolve around reduced energy and faster transactions of the PoW, such as IoT. This is because IoT applications rely on a permission blockchain. However, even with the permissioned nature of the private blockchain, IoT remains prone to attacks, such as device capturing and cloning. Additionally, IoT devices are characterized by a

key limitation in hardware resources and are energy constrained. In the proposed consensus model, nodes will not be involved in the mining and verification until the random beacon mechanism selects it. This will allow IoT devices to perform their application-specific tasks while at the same time, mining blocks.

7.3.1 Random Beacon

The random beacon is the source of autonomy and unpredictability in the system and is used to produce unpredictable random values. Based on the asymmetric public key cryptography concept, a digital signature produced from the random beacon is a unique and unpredictable value that can be used as a source of randomness to generate random values from it (Kelsey et al., 2019). The centralized random beacon model can be susceptible to manipulation, as the signer will have control of the random beacon process, which is dependent on the signer's private key. This can affect the process of generating random values and makes it vulnerable to manipulation. Furthermore, it can also be a single point of failure. If a signer who selected by the centralized random beacon to generate the next signature is hacked or is offline, it can halt the random value process. In addition, if a malicious adversary controlled a signer node can then send conflicted random values to more than one client. To solve the previously mentioned issue, the BLS threshold signature has been used to provide a decentralized random beacon that can be operated by all the members of the threshold committee. Therefore, the decentralized random beacon can act as a trusted third party. In addition, the produced output does not need to agree on by running a full consensus. The random beacon in our consensus performs as a verifiable random function (VRF) and utilized as a method for randomness-based sharding on top of the PoW consensus protocol. The random beacon in our PoNW algorithm relies on BLS signature as introduced in Dfinity consensus (Hanke et al., 2018). The output of the VRF cannot be predicted by anyone utile released for all clients.

7.3.1.1 The BLS Signature Scheme

BLS is a unique deterministic pairing-based signature scheme introduced by Dan Boneh, Ben Lynn, and Hovav Shacham (Boneh et al., 2001). This scheme provides properties of uniqueness, non-interactiveness threshold signature, which allows a shorter threshold signature comparing to other similar approaches, where K out of N signature shares are adequate to generate a valid combined threshold signature. Irrespective of which subset is signed, it produces the same threshold signature that will be verified with the group public key. It also provides a friendly distributed key generation mechanism. Algorithms 2-6 defines these methods.

Algorithm 2 BLS parameters

1: Two elliptic curves: E_1 and E_2.
2: E_1 and E_2 have two elements P_1 & P_2 of prime order p.
3: Two groups G_1 and G_2 of prime order r on two elliptic curves E_1 and E_2.

Algorithm 3 Generators

1: $P_1 \in G_1$
2: $P_2 \in G_2$
3: Bi-linear and non-degenerate pairing: $G_1 \times G_2 \rightarrow G_T$

Algorithm 4 Key Generation

1: Secret key is a random bit string between 1 to $p-1$ bits: $SK = x$
2: $SK : x \pmod{p}$
3: Public key: $PK = xP_2 \in G_2$

Algorithm 5 Signature generation

1: Input: M (Message)
2: Output: TS - the threshold signature
2: Sign: $SK = xM$
3: Message hashed: $H(M) \in G_1$
3: Signature: $TS = xH(M)$

Algorithm 6 Signature Verification

1: Input: PK, $H(M)$ and TS.
2: Output: $True/False$
3: $\hat{e}(TS, P_2) = \hat{e}(H(M), PK)$

7.3.2 A comparative analysis of the PoNW and similar consensus mechanisms

A significant of all the researches produced recently on blockchain consensus algorithms have shown focus on addressing throughput limitations, scalability improvement, and reducing the energy consumption of the current PoW consensus protocols. For instance, the authors in (Shang, 2018) demonstrate a durable strategy for consensus against non-cooperative behaviours. It is essentially concerned with a hybrid network of non-rational agents. A filtering approach is used to evaluate the likelihood of convergence. Convergence is an issue in this method. However, a consensus challenge for numerous delta operator systems has sparked additional attention to the distributed technique for modelling continuous-time operations at fast sampling (Chen et al., 2017). Authors in (Cheng et al., 2018) proposed the use of a hybrid consensus protocol termed the Deterministic Proof of Work (DPoW), which promises to provide impressive consistency and scalability, with no downtime in decentralization. The proposed consensus comes in two major parts. The first part works on solving the PoW cryptographic puzzle while the second part works on verifying the proposed result's correctness. In doing so, the system provides the users with benefits associated with the PBFT and PoW protocols and often referred to as a map-reduced PoW mining algorithm. However, this is a conceptual model and still, there is more work that needs to be improved, including the reputation score in the verifier election process. Nevertheless, the authors offered some suggestions for how to address this issue in their future work. For instance, they suggest using it in conjunction with the random seed to adjust the likelihood of a given individual being selected as a verifier.

Similarly, the study in (Eyal et al., 2016) introduced the Bitcoin-NG consensus protocol. This protocol works on reducing the transaction's processing latency by combining PoW with Byzantine's tolerance. The main idea here is decoupling the miner election's process from transaction verification by using two different types of blocks. These include Micro blocks and Key blocks. The function of

the Key blocks is to use PoW in serving as a leader selection. The leader from key blocks then assumes the responsibility of creating Micro blocks, which are crucial for transactions requiring the leader's signature without needing a power-consuming PoW. One key downside, however, is, even with that potential, the Bitcoin-NG houses a number of challenges such as history rewriting and even deliberate forking. Another consensus protocol was presented in (Cicada, 2016), the paper introduced a PoW consensus mechanism that allocates miners randomly into small mining pools called the distributed proof-of-work consensus. According to Cicada white paper, this consensus protocol uses a Distributed Hash Table to reduce storage overheads. The system then uses small amounts of energy by reducing the number of nodes being used to achieve mining when compared to the original PoW. However, the system is not without challenges as it has been criticized for experiencing difficulties in implementing the miner's selection process results (Sanders and Liebig, 2019).

In addition, several research efforts were geared towards finding a solution towards the key challenge of reliance on consensus algorithm by a small group of trusted replicas. One such example is the Entangled proofs of Work and Knowledge (EWoK) (Armknecht et al., 2017). This mechanism divides nodes into shards. Additionally, this mechanism requires workers to store every part of the suggested blockchain data independently. While this mechanism promises to improve issues of sharding, it increases the problem of cross-sharding communication overhead. This is because miners are incentivized to store the shards locally in an attempt to gain an advantage in solving the next PoW hash-based puzzle. Similarly, the work in (Lundbæk et al., 2018) introduced the practical Proof of Kernel work (PPoKW); it is another leaderless consensus algorithm. This algorithm is based on a low-energy PoW consensus that works to reduce the number of nodes in the PoW cryptographic puzzle and does the selection of nodes randomly to carry out the mining processes. This algorithm makes its node selection in a similar way to the approach in (Gilad et al., 2017), which is based on a cryptographic sortation. However, one key criticism of this algorithm is its storage of the white list into the

chain as it gives rise to scalability issues (Sanders and Liebig, 2019). Additionally, the VRF model has to deal with the Last actor abort. This challenge encompasses a scenario where the last actor can reveal their commitment during the process of generating random value.

The review and discussion of similar works have concluded that the solutions based on sharding use a variety of different methods to assign blockchain nodes to different groups. Although sharded blockchains have proven to be more potential compared to the traditional BFT, there was still a need to ensure the per-subchain consensus protocol runs across hundreds of participating in adversarial environments (Dang et al., 2019). As the number of nodes achieving the consensus is minimized, the probability of an adversary being able to abort the system becomes higher. However, our consensus differs from other similar works in that it solves the concern of the probability of an adversary aborting the system due to the reduction of the number of nodes that achieve consensus. For this, our proposed mechanism introduced a secure random model to randomly select participants to perform PoW. Thus, stopping an adversary from concentrating its presence in one committee and exceeding the byzantine-tolerance threshold. Additionally, we proposed a ranking mechanism to resolve chain fork, which is based on the Pseudo-Random Process along with a permutation function to arrange selected committee members into sequential order.

7.4 System Components

The proposed consensus protocol (PoNW) is comprised of a number of entities. In this section, we begin by describing the components of the PoNW consensus, formalising the required concepts, and then illustrating the system model in Figure 7.1.

7.4.1 Block Structure

The block in our PoNW consensus has the structure of: $B = (p, r, z, d, o)$

where:

- p: is the previous block
- r: is the round number
- z: is the notarization of the previous block
- d: is the data payload, a set of transactions and state
- o: the block creator (owner)

7.4.1.1 Chain structure

The chain (C) represents a set of a sequential order of blocks ($B_0, B_1, ..., B_{r-1}$), where r is the round number of the block B_r. The previous block is $H(B_{i-1})$ for all i 0.

The notarization of the previous block represents a valid threshold signature of B_{i-1} for all i 0, where B_0 represent the genesis block; B_1 is the first block after the genesis block; and B_{r-1} is the head of the chain C. If more than one node submitted a block, which in return produce a fork of more than one chain available: *Chain 1* $= C$ and *Chain 2* $= C^{'}$, where head of the chain C is the head of the chain $C^{'}$. Then S is the set of blocks in the chain and $C(S)$ is a chain of set of blocks S, and which donate the largest common prefix of chains $C(B)$, where $B \in S$.

7.4.1.2 Nodes

Nodes in the blockchain network $1,2,3,...n_i \in N$. Each node $i \in U$, where U is the set of all nodes in the blockchain system. Each node i has a public and private key pair: pk_i indicates the node's public key; and sk_i indicates the node's private key. In a private (permissioned) blockchain model, the set of public keys for all nodes in the blockchain is known for all nodes.

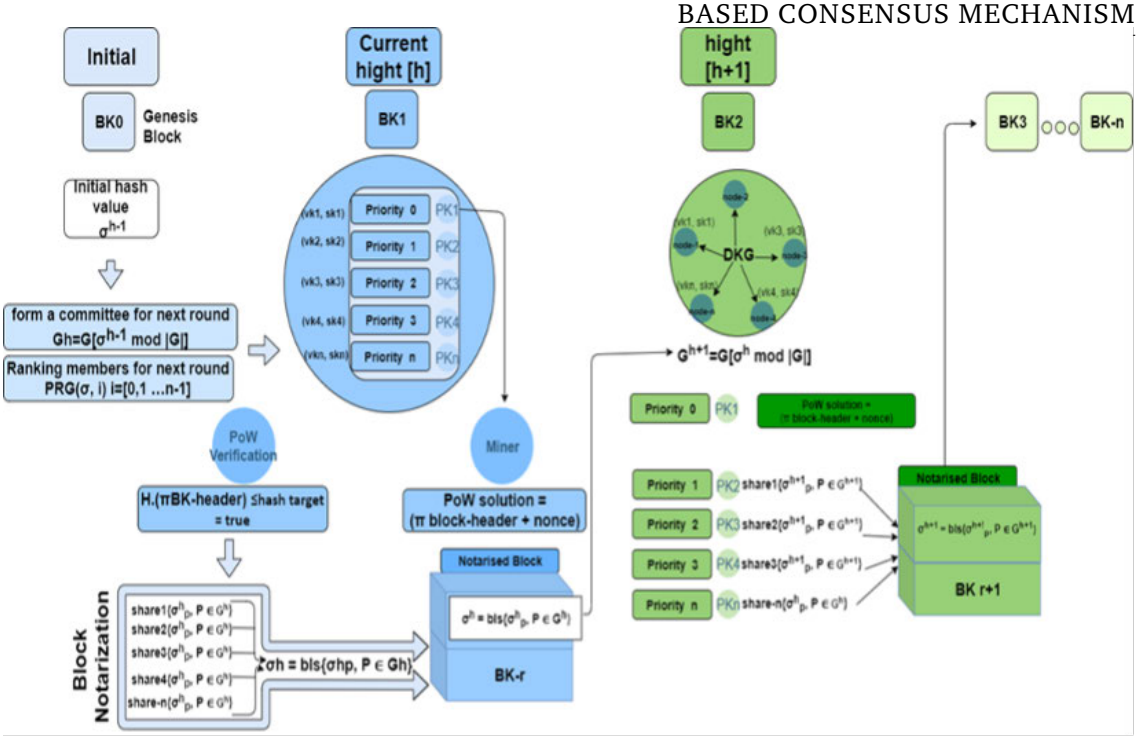

Figure 7.1: Proof of Notarized Work system model

7.4.1.3 Groups

At each round, a group is created and nodes $i \in U$ in the blockchain network are allocated randomly into a single or multiple portion. Where a one group forms a committee, we always have a single group active for current round to agree on a block (notarization) and to drive the randomness process for the following rounds.

7.4.1.4 Byzantine nodes

A group is fault-tolerant and any subset of threshold size can distribute signature shares to combine it into a single threshold signature. Every member in the group can then combine the received signature shares to produce the group signature. This will produce a unique deterministic signature, which will be the same irrespective to which members signed.

7.4.1.5 Decentralized Notary

The block notarization process in our consensus is decentralized, which generated by all the group members. The notarization in the block is the threshold signature

under a block created by a leader who selected by a random beacon from the previous round. The notary members are looking to agree on the correctness of the cryptographically solved block in the current round. The notarization is not a consensus. However, the notarization process can be used to reach consensus about a block during the normal process of the current round. Before it can consider a block as a notarized. A block needs to receive enough signature shares from the notary members. This will reduce the time need it to finalize a block, as the minimum threshold number required to sign a block will act as a Byzantine agreement. Thus, it does not need a separate consensus protocol to achieve this, which provides a very fast block finality at the same time of generating the random beacon (Figure 7.1).

The blockchain system initialized with an initial hash value stored on the genesis block. The produced hash forms a random beacon $\sigma h-1$, which is going to be used to select committee members for the first round $Gh = G[\sigma h-1 mod|G|]$. The DKG process leaves each member with a public verification vector and its secret key shares (vk, sk). If more than one miner solves the block cryptographic puzzle, preference is given to the highest-ranked node. The notary members at the first round r verify that the block Br is solved correctly. Then they sign the block and send their secret shares to be combined in a single threshold signature to form the block notarization, which is then be used to select committee for the next round r+1. After that, members of the next committee sign the previous threshold signature just after beginning the new round r+1 to produce new random beacon Gh+1= G $[\sigma hmod|G|]$, which is going to be used to generate the following random beacon and so on.

7.5 PoNW vs PoW

The PoW consensus protocol is a robust mechanism that deters malicious participation. Ethereum, Bitcoin, and many other blockchain applications are based on PoW and have been tested in real-world situations. The security properties of the PoW

demand an honest majority, yet keeping everyone informed and agreeing implicitly might be costly for scalability. In addition, due to scalability issues, PoW consensus solutions cannot ensure immediate finality. This is a validation process to prevent arbitrary state changes.

On the other side, our PoNW technique preserves the security of the PoW consensus protocol while enhancing its scalability and energy consumption. Compared to public consensus protocols, such as PoW, we can see that our hybrid consensus mechanism has more promise for optimised decentralisation, efficiency, practicality, and security. Our PoNW is a hybrid consensus mechanism that balances PBFT and PoW consensus mechanisms. To prevent an adversary from concentrating its presence in one committee and surpassing the byzantine-tolerance barrier, we devised a secure random mechanism to pick PoW participants. The Pseudo-Random Process and a permutation function were used to rank committee members to resolve the chain fork.

7.6 PoNW Properties

7.6.1 Faster Block Finality

Finality is a concept that guarantees the previous transactions is irreversible, and can never change. This is a significant property, which measures the time needed to wait before it can guarantee that the transaction written in the blockchain cannot be changed. Therefore, most of the blockchain systems today can provide probabilistic finality, which cannot guarantee immediate finality. Such as in PoW which relies on the longest chain of work. Due to the competition between miners to mine a current block, it is possible to have more than one miner creating more than a block at the same height. As a result, the chain will divide it into more than one fork. Thus, to decide which chain is the valid chain from all other forks, a different fork resolution process used to choose between the forked chains. For an instant, GHOST protocol used in Ethereum (Sompolinsky and Zohar, 2015),

and the longest chain rule is used in Bitcoin (Nakamoto, 2008). In our PoNW the highest weight chain based on the ranking of the nodes, which is derived from the threshold signature. The node ranking process represents the weight of the nodes that can to add blocks to the chain. Therefore, this approach provides a valuable solution to select between the competed chains. In case if more than one node submitted a valid block, preference is given to the highest-ranked node.

7.6.2 Block Notarization

Our PoNW provides a fast finality by proposing the use of block notarization process similar to the one that defined in Dfinity (Hanke et al., 2018). Notarization represented as a threshold signature that generated collectively by all nodes in the notary group. This work differs from the traditional PoW, as in our PoNW, the highest-ranked chain is not based on the longest chain of work. Instead, it relays on the random beacon itself. In PoNW, the list of all active nodes in the network is known. The ranking process is driven from the threshold signature to generate an ordered list of ranked nodes that allowed to add a block to the blockchain. As a result, this will provide a secure mechanism of randomly ranking nodes based on the publicly verifiable ranking process that is driven from the distributed random beacon. Therefore, an adversary cannot interfere with the ranking mechanism, as this requires the majority to contribute to generating the threshold signature. If the notary group receives a block, they first check to see whether the block is valid or not. If the block is not valid, they discard it. The notaries will notarize the highest-ranked block if it is valid by signing it with their secret shares and broadcast it. The valid signature can be generated once the block has received a majority signature that is required for the threshold signature. This signature will represent a notarization for the block so that block can be added to the blockchain. Therefore, notarization will resolve any fork in the network, and the chain will only add the notarized blocks. As a result, this will help to achieve finality in a subsequent normal round. A valid block proposed at the height h must reference a block

that was notarized at h-1. In the current round r, a block Br will be finalized and appended to the final chain just after receiving a notarization for Br+1. It means that a block can be finalized after two confirmations plus the relay time as the notary can run at the same speed as the random beacon. Therefore, notarization will provide a fast finality in a few seconds. The many advantages offered by the BLS would perfectly justify the small degradation of performance when is compared with 10 minutes finality time in Bitcoin and 15 seconds in Ethereum.

7.6.3 How to Relay Between Committees

The unique threshold signature ξr-1 that produced in the previous round r-1 will be used to prioritize the nodes that are going to mine a block Br at the current round r. ξr is the threshold signature for the current round r. The notary members at the current round r that selected by ξr-1 are going to verify that the block Br is solved correctly, and then sign it. Each member sends his signature shares to be combined in a single threshold signature ξr. When block Br received signature shares from the majority requires for the threshold, the block considered as notarized. The notarization on the block is aggregated signature from previous rounds. After that, members of the next committee sign the previous threshold signature ξr just after bringing the new round r+1 to produce new random beacon output ξr+1, which is going to be used to generate the following random beacon. The new produced unique threshold signature ξr+1 will then rank miners for the coming rounds and so on.

7.6.4 Random Beacon Distributed Key Generation

The random beacon provides a verifiable and friendly distributed key generation process that does not need for a trusted dealer. It allows a set of n parties to collectively generate the secret key shares and the group's public key that required for the scheme. Distributed Key Generation (DKG) algorithms is an integral part of any threshold cryptosystems, as it provides an efficient key pair (private & public)

generation process that need it to initialize the threshold cryptosystem. In our PoNW consensus, we proposed using a non-interactive DKG protocol based on Gennaro, Jarecki, Krawczyk and Rabin [GJKR] protocol (Gennaro et al., 2002).

7.6.5 Distributed Key Generation Process

The threshold group members will generate a shared secret key without knowing the individuals and public keys. When the number of the threshold group members who agreed to sign on the message is satisfied, a new single threshold signature produced, which is the result of the combination of the signature shares of the threshold group members. Then the threshold signature can be verified by anyone who knows the group public key. As a result, each member of the group can contribute to generating a secret key that needs it for signing the group's messages. Moreover, the DKG process produces a group verification vector, which includes the public key for the group. Each member in the group can combine all the verification vectors that been received from other members to produce a single verification vector that can be used to verify a message signed by the group. Each member of the group will generate a verification vector and advertise it publicly so other members can see it. Each member will generate a secret key contribution share for other members in the group and posted to other members. Members of the group send their secret key contribution shares between each other. For the verification of shares received, each member validates the contribution share that received from other members against the verification vector of the sender who sends it and then saves it. Finally, after all the group's members receive their shares, they contribute to produce the group's secret key. The group verification vector can then, use it to derive any of the member public keys.

7.6.6 Pseudo Random Number Generation

As we discussed earlier, the decentralised random beacon will drive the process of randomly selecting nodes for the next committee. We agree that the random beacon

is derived it from the unique deterministic threshold signature: ξ. Therefore, we need a PGR to generate a sequence of random values from the threshold signature ξ. Given that $PRG(\xi, i)$ for $i = [0,1,...,n]$, the random sequence values $PRG(\xi, i)$ can then be inserted as an input for a permutation function, to arrange a set of group's members into a sequential order $1...,|U| \rightarrow U$. An example of this, the permuted congruential generator (PCG) 212, which provides an efficient statistical performance with a small state size. This will produce an ordered list of nodes identified by its public keys $P_1, P_2, ...P_n$. To form the current group G_r we need a seed ξ and the group size n. The seed will be the previous threshold signature ξ_{r-1} and the group size is n. Members of the current group G_r for the current round 'r' will be derived from the previous threshold signature $\xi_{r-1} \pmod{n}$. Algorithm 1 represents the process of forming a group.

$$G_r = i_1, i_2 ... i_n$$
$$G_r = G_i, i = \xi_r \pmod{n} \tag{7.1}$$

Therefore, using a pseudo-random number generation process, along with permutation function, will only allow blocks from the highest-ranked nodes to be added to the cryptography chain. The notarization from the highest-ranked node will be valid when signed with the secret shares and then can be broadcasted. In this case, the inclusion of notarization helps resolve any forks in the network, and only a notarized block will be added to the chain. From the evaluation, it is clear that our algorithm offers a higher level of security against chain forks.

7.7 Security Analysis

This section focuses on carrying out an analysis of the security of our solutions in regards to the model previously highlighted under section 4. In the analysis of the decentralized random beacon, the main assumption is that the model uses a cryptographically strong pseudo-random generator in the system's genesis to

generate the initial seed. In the case that a central system authority creates the Genesis Block, the system can be used in creating the requisite seed for generating it from a source characterized by high entropy. However, it should be noted that the model shall not set up a threshold signature scheme by relying on a trusted third party. Therefore, in this case, the group G shall set up the group public key and the secret key shares by running a DKG for the BLS, when initializing the blockchain system. Lastly, the signing process shall be repeated in non-interactive mode. In this section, the focus will be on potential factors that can be used by the adversaries to attack the proposed system together with ways to mitigate the occurrence of such threats. In terms of the security model, the key assumption is that honesty is maintained by at least two-thirds of the nodes. Therefore, if more than a third of the nodes are faulty, the algorithm fails to reach a consensus. For this system, the maximum number of nodes before breaking the consensus is 33% and could be made up of comprised nodes or offline nodes. For example, considering the assumption of the BFT mechanisms, in a network with 1000 nodes, it requires no more than 333 nodes are faulty in order to the blockchain system to be considered as a safe. In the case that the consensus nodes are divided into four shards, the consensus nodes will be divided into the quarter with each group assigned 250 nodes. Achieving a consensus, in this case, will require the group to work in a parallel fashion. Therefore, an adversary will only need 83 nodes to fail a consensus. This shows that sharding reduced the system's fault tolerance from 333 nodes to 83 nodes. However, modern-day technology has made it possible for sharding techniques to rely on a sort of randomness in assigning the nodes to their shards, reducing the probability of all 83 nodes being in one shard. In the case that the adversary controls 250 of all nodes in the system, there is a high possibility that all 83 malicious nodes out of the 250 will be in one shard. The previous assumption requires a higher number of nodes in each shard in reducing this high probability. It is a trade-off between the minimum number of nodes and security per shard. Therefore, while having a large number of shards with a reduced number of nodes improves a system's throughput, it increases the probability of having a shard

compromised by malicious nodes.

7.7.1 Threats Model

In analyzing our protocol, the attack shall be deemed as originating from an adversary that has control over a certain fraction of all participant's machines. The underlying assumption is that the adversary's probability to break any cryptographic primitives is negligible. This is because, with the number of nodes undertaking in consensus reduced significantly, the probability associated with aborting an algorithm increases. The good thing is that the number can undergo optimization to strike a balance between reliability and performance. The model showed that the security of the PoNW consensus algorithm is upheld only after the bounds highlighted under equations (4) is upheld. The bounds are maximal, and the network may prove to be much secure when subjected to lesser stringent conditions.

We begin assuming $f(G)$ is number of Byzantine nodes in a group G and n is the group size, we have Assumption 1, where β 2:

$$|U| \ \beta f(U) \tag{7.2}$$

And Assumption 2:

$$n \ 2f(G) \tag{7.3}$$

Each group G in the system represents a random sample of all the nodes in the blockchain system U. Given Assumption 2, each group G is honest, and each group has a fixed size of n.

To calculate the probabilistic of G honest we used the formula:

$$X \sim Hypergeometric(N,K,n) \tag{7.4}$$

Formula 7.4 gives a random variable distributed hyper-geometrically with the elements of the population given as N, K and n. It has a probability calculated

using Formula 7.5.

$$P_x(k) = P_r(X = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}} \tag{7.5}$$

Formula 7.5 signify the probability function of the hypergeometric distribution, and where:

- N: is the population size.
- K: is the number of success states in the population.
- n: is the number of draws.
- k: is the number of observed success.
- (ab) is a binomial coefficient.

The function is positive when:

$$Max(0, n + K - N) \leq K \leq Min(K, n) \tag{7.6}$$

Regarding to the hypergeometric Distribution formula, all items of the population is sampled and the result of the draws is classified. In our example, a group is drown from the total number of publication without replacement. To demonstrate this, we used the hyper-distributed probability code, a Python program developed by Tari labs available in GitHub (Kevoulee, 2019).

Figure 7.2 shows the calculations of the probability of the adversary controlling the blockchain system using the hyper-distributed probability python code, with the elements of the BFT threshold given as:

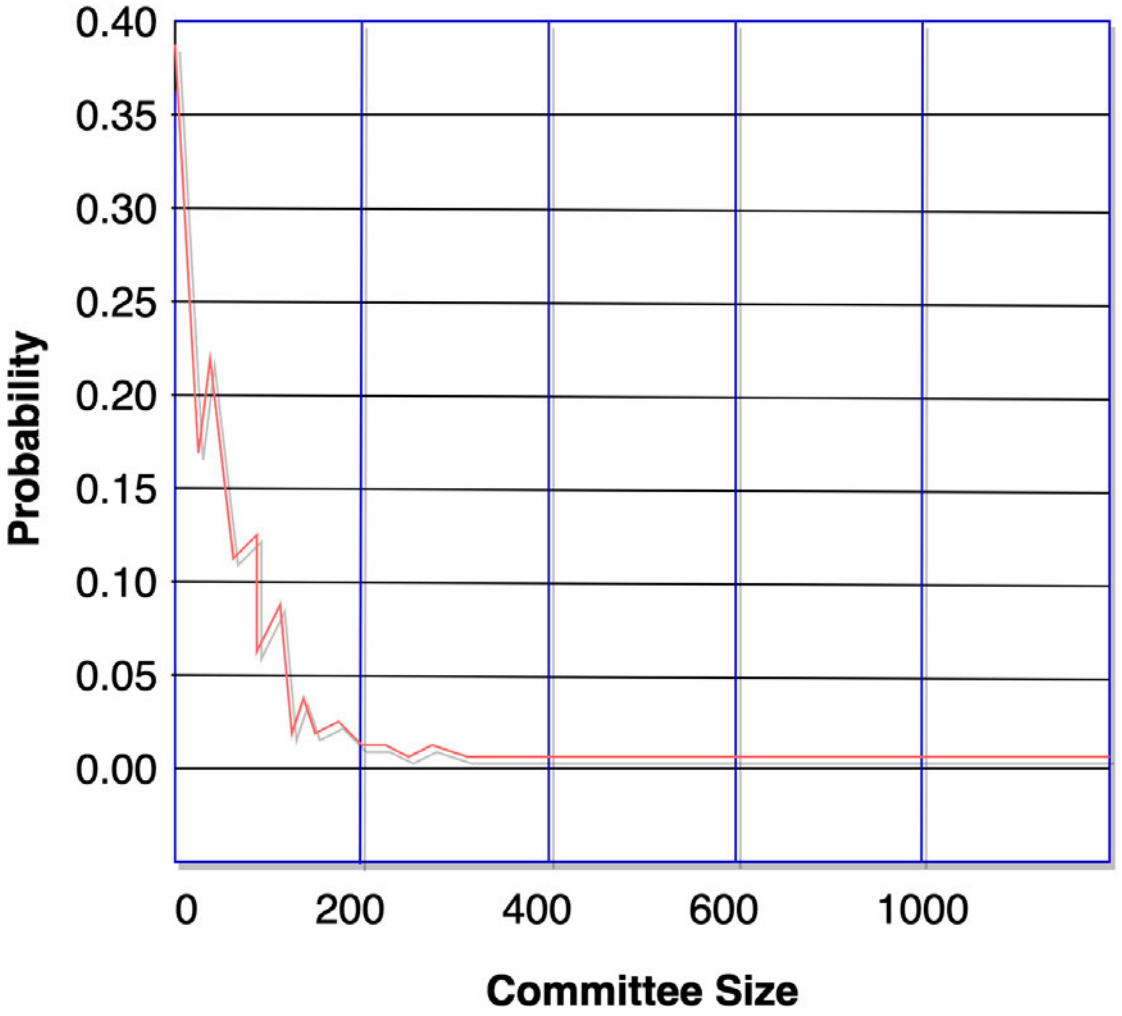

Figure 7.2: Committee size against probability with the total publication size of 1000.

- N: is the population size = 1000
- K: is the number of success states in the population is 60.
- n: is the number of draws (committee size from 1 to 1000) .
- k: is the number of observed success. This donates the BFT threshold, which assumes two-thirds of the nodes are honest, which is 67%.

7.8 PoNW's performance in comparison to other consensus protocols

Blockchain consensus maintains three important properties: security, decentralisation, and scalability. Many Blockchain consensuses are only able to pick two of these properties while having to compromise on the third. This commonly affects the design choices of consensus mechanisms. Table 7.1 and 7.2 illustrate the comparison of different consensus protocols with our proposed method.

Table 7.1: The comparison between different consensus mechanisms' characteristics

Consensus	Security	Privacy	Reduce mining	Less risk possibility	Less block creation time	Reference
PoW	✓					(Nakamoto, 2008)
PoS		✓			✓	(Siim, 2017)
PoBT	✓		✓			(Biswas et al., 2019)
PoC			✓			(Dziembowski et al., 2015)
PoAh		✓		✓		(Puthal et al., 2020)
RPoS					✓	(Li et al., 2020)
PoNW	✓	✓	✓	✓	✓	(Abubakar et al., 2020)

Table 7.2: The performance of different consensus mechanisms

Consensus	Block creation time	Transactions per second	Adversary tolerance	References
PoW/Bitcoin	10 min	7	<51%	(Nakamoto, 2008)
PoW/Ethereum	15 sec	15	<51%	(Buterin et al., 2014)
PoS/Ethereum	12 sec	TBD	<51%	(Buterin, 2016)
PoS/Microchain	9 sec	230+	<33%	(Xu et al., 2019)
PoS/PIVVX	60 sec	173	<51%	(paper, 2018)
Algorand	4.5 sec	1000	<33%	(Chen and Micali, 2016)
Hybrid/Decred	5 min	14	<51%	(Documentation, 2016)
PoC/Brustchain	4 min	80+	<50%	(Gauld et al., 2017)
PoI/NEM	1 min	4000	<51%	(Xiao et al., 2021)
PoNW	4 - 5 sec	<1000	<33%	(Abubakar et al., 2020)

7.9 Possible Attacks

It is crucial that attention is paid to the functioning of the consensus model under both normal and adversarial conditions. For such an environment, the consensus mechanism has to be prepared in dealing with the following attacks.

7.9.1 Randomness Manipulation Attacks

The randomness generation process is one that is prone to frequent attacks. One such attack is the randomness manipulation attack. While using a proof-based consensus protocol to generate randomness, the generated randomness can be manipulated by any insider malicious attacker who can either withhold valid blocks or refuse to mine. This can force the system to rely on a single source in the generation of random beacons. In such a case, the random value process tasked with the generation of random beacons can be halted in the case that a signer

selected by the random beacon offline or is hacked. Additionally, a malicious adversary can send conflicting values to various clients when he or she gets control of a signer node. In the attempt to prevent attackers from manipulating values that are generated from the random generation process, the PoNW switched to using a decentralized random beacon to generate randomness. For our model, we decided to rely on a BLS threshold pairing signature scheme as the default random beacon. By using this, the model is guaranteed of a stable decentralized random beacon that is difficult to manipulate, as it requires a minimum number of the threshold members to be generated.

7.9.2 Chain fork

Our PoNW leverages a permutation function with a Pseudo-Random Process in an attempt to sequentially arrange the selected committee members. In doing so, the algorithm allows for the selection between the competed chains. This is an outstanding breakthrough as the proposed PoNW consensus algorithm provides a solution to issues of forking by proposing the use of a ranking process stated in section 4.2.6. In case more than one node submitted a solution for the block puzzle, the algorithm will add blocks that mined by the highest-ranked nodes in the cryptography chain. The notarization of the highest-ranked nodes done by the notary nodes of the block will be valid when signed with the secret shares. In this case, the inclusion of notarization helps resolve any forks in the network, and only the notarized blocks will be added to the chain. From the evaluation, it is clear that our algorithm offers a higher level of security against the mentioned attacks.

7.10 Conclusion

In this chapter, we proposed the PoNW consensus algorithm, which is a hybrid approach based on a reduced mining algorithm combined with a PBFT verification. Our protocol has shown the potential to achieve a high level of consistency and security by using a decentralized random beacon, which acts as a Verifiable

Random Function (VRF) that requires the contribution of a majority of the group members by sending their signature shares to be used in the production of a unique, unpredictable, and deterministic threshold signature. The system then proceeds to use the threshold signature in carrying out the node selection required for the next group. The study has also provided an analysis of the consensus protocol's security model, together with estimations regarding the probability of an adversary controlling the consensus mechanism. The analysis showed that the PoNW is resistant against the 51% attack and also increased this threshold by 66.6%, which achieves great levels of consistency and greater security in maintaining decentralization. It is our belief that the PoNW is a representation of a major step towards the development of more secure decentralized applications. The low latency achieved by the algorithm allows for a myriad of applications, which were complex or impossible to achieve with previous latency consensus methods. We hope that the proposed mechanism will help pave the way for additional research in this area.

eighth Chapter

Conclusion

8.1 Introduction

Authentication and access control solutions are the fundamental components that must be handled in order to create secure and reliable Internet of Things applications. Despite the pervasiveness of IoT in many facets of our daily life, existing authentication and access control solutions confront a number of security concerns. This thesis addresses two of these issues; primarily, it focuses on enhancing the privacy of persons and providing lightweight IoT authentications and access control solutions in order to strengthen the security of IoT systems using decentralised techniques as the primary strategy. In order to accomplish this, we first looked into the issues that need to be resolved and analysed the requirements that needed to be met. Frameworks and models were then suggested as well as architectural designs. In our research, we used a wide range of tools, programming languages and techniques from a variety of scientific fields as well as best practices and concepts from several study disciplines. We employed blockchain technology extensively to decentralise and increase the security of IoT networks. This chapter provides a summary of the research contributions made in this thesis and a discussion of possible future works. Following is a brief description of the work described in this thesis, followed by a discussion of potential future work, and finally some concluding remarks.

8.1.1 Summery

The key contributions of this thesis are presented below, and the work that has been done is summarised chapter by chapter.

- **In Chapter 4:** we provided a proof-of-concept design and implementation of a blockchain-based two-factor authentication system for web-based access to sensor data. The proposed method provided a user-centric and lightweight authentication solution. The proposed method takes advantage of the smart contract scripting power of the Ethereum network. The proposed approach excludes the usage of a third party for two-factor authentication process maintenance or OTP generation and validation. We utilised blockchain-based decentralised identity capabilities to provide individuals complete control over their authentication information instead of having it maintained by a third party. Moreover, our method enabled customers to keep their IDs directly in their mobile wallet application. Therefore, it allows users to represent themselves as actual persons. Chapter 4 goes further and provides performance and security analysis to prove the feasibility of the proposed solution. Based on evaluation results, the proposed method is efficient and capable of facilitating trustworthy authentication.

- **Chapter 5:** has presented a proof-of-concept design and implementation of a blockchain-based authentication and authorisation schema for MQTT messaging protocol. Blockchain technology based on Ethereum was used to implement the proposed solution. Our solution also included a decentralised identity paradigm, which gives users complete control over their identity and data without the need for centralised identity management. The chapter described the implementation of our approach and the specific technologies that were needed for the implementation. Our system's performance and associated expenses have also been examined

in detail. We believe that our solution will provide a lightweight method for facilitating MQTT protocol authentication in a distributed and safe manner. Since transactions are added to the blockchain, there has been some delay, but it is still within an acceptable range compared to previous two-factor authentication systems that involve a third party to maintain authentication like SMS and Emails. Our method, on the other hand, outperforms TLS in terms of computation and storage overhead because of its minimal CPU and memory usage, allowing resource constrained IoT devices to run many applications simultaneously without experiencing overloading. It is clear that our strategy satisfies the security criteria for Internet of Things applications and is adequate to satisfy the needs of the future.

- **In chapter 6:** a blockchain-based authentication and access control system for the Internet of Healthcare Things is proposed in this chapter. The chapter demonstrated the design and implementation of a lightweight and secure framework for medical wearable devices as a proof-of-concept. The proposed approach contributes to the resolution of privacy and security issues on Internet of Healthcare Things systems. It offers a privacy-preserving access control method and makes safe user authentication easier. A third-party administrator is no longer required to maintain the system's users' credentials. A medical wearable sensor prototype and accompanying smartphone app, built on the Ethereum blockchain, demonstrated the feasibility of our solution. Analysis of our system's security and an attack model has been accomplished to assess our system's capacity to meet the security criteria of IoT systems. When compared to current centralised solutions, our method improved security and consumer privacy. In addition, we gave a performance analysis as well as the associated transaction costs. Our technique utilises low memory and CPU compared to the current TLS, making it ideal

for devices with limited resources. Our method will satisfy the security requirements for Internet of Things applications and fulfil future needs. We hope that our method would improve the privacy and security of sensitive health data belonging to users. In contrast, the suggested method showed a considerable delay due to the peculiarities of the public blockchain, as transactions must be attached to the block. However, the delay is still within an acceptable range.

- **In chapter 7:** The PoNW consensus algorithm was introduced in this chapter, which is a hybrid technique based on a simplified mining algorithm paired with PBFT verification. Our protocol has demonstrated the ability to achieve a high level of consistency and security by utilising a decentralised random beacon that acts as a Verifiable Random Function (VRF) that requires the contribution of a majority of group members by sending their signature shares to be used in the production of a unique, unpredictable, and deterministic threshold signature. The system then uses the threshold signature to do the node selection necessary for the following group. In addition, the study examined the consensus protocol's security model and made estimates about the possibility of an adversary controlling the consensus mechanism. The investigation revealed that the PoNW is resistant to a 51% attack and improved this barrier by 66,6%, hence achieving high levels of consistency and enhanced decentralised security. We believe that the PoNW is a significant step toward the creation of more secure decentralised applications. The algorithm's low latency enables a variety of applications that were difficult or impossible to implement with earlier latency consensus approaches.

8.1.2 Future work

Despite all of the efforts that have been made in the previous research and the outcomes that have been provided in this thesis. There are still significant

issues regarding the performance and throughput of the existing blockchain systems, The following provides a synopsis of the future work that is intended to address those concerns with regard to each of the key chapters of this thesis.

- **Chapter 4:** In our future work, the primary focus will be on validating the interoperability of the proposed decentralised two-factor authentication (2FA) technique with various IoT frameworks. We plan to conduct experiments with a variety of consensus algorithms and distributed ledger technologies in order to speed up the rate at which transactions are processed, as well as to cut down on the delay of the transaction.

- **Chapter 5:** For the work proposed in chapter 5, the authentication capability of the smart contract is going to be expanded in the work that we plan to do in the future so that it can handle a wide variety of regulations and conditions. In addition, we will implement our solution by making use of a variety of blockchain systems, such as Hyperledger and IOTA, and we will investigate a wide range of consensus mechanisms in order to enhance the performance and reduce the end-to-end delay.

- **Chapter 6:** Regarding the work introduced in chapter 6, we plan to extend the decentralised authentication and access control model. First, we plan to expand the privacy model of the proposed framework to serve many types of users, including patients, doctors, other health providers, as well as insurance companies. As a second future work, we aim to explore different consensus mechanisms and blockchain systems, such as Hyperledger and IOTA, to improve transaction speed and reduce the end-to-end delay. Third, we plan to extend the smart contract functionality to handle various rules and conditions. Finally, we plan to improve the system's usability by building a genuine wearable medical device with different medical sensors and exploring different boards and microcontrollers.

- **Chapter 7:** As distributed consensus on a blockchain aids in decentralising IoT, we believe its inclusion in IoT is critical. Therefore, the future plan is to extend the hybrid blockchain consensus mechanism for IoT that was proposed in chapter 7 according to several directions. First, in our current design, we provided a consensus algorithm of principle. Therefore, for future work, we plan to implement the PoNW consensus in a subsequent practical system and evaluate it in big data scenarios for large scale networks. Our other future works include a comparative study of various benchmark security solutions in large-scale networks and evaluating different threat attack scenarios.

8.1.3 The open research directions and the conclusion remarks

The exponential increase of blockchain adoption in IoT opens up various research areas. Many blockchain-based techniques have recently emerged with the goal of facilitating the establishment of decentralised IoT applications. This will provide blockchain-based interoperability for the Internet of Things. Additionally, it will allow new opportunities and scalable data-driven business models. For example, adopting a smart contract in the blockchain will simplify transactions by eliminating intermediaries. As a result of the technology replacing intermediaries, trades can be completed faster and at a lower cost. However, the integration of blockchain technology with IoT is still in its infancy, and serious issues about its functional and practical practicality remain. The general research implications of blockchain technology are diverse. For instance, the transactional transparency contained in the revolution of this technology is considered a double-edged sword. The blockchain has been questioned regarding its ability to securely store data, as the completed transparency is contrary to confidentiality. Besides, the principal characteristic of the blockchain is preserving the integrity of data by rendering it immutable. Nevertheless, this feature may be a double-edged sword as well. The reason is that errors that shared over the distributed ledger cannot be corrected, or in the

case of users need to delete their personal data. Furthermore, the lack of central authority and intermediation of an authenticated system is unlikely to be defined uniformly by all countries in the world. Thus, limiting the widespread adoption of this technology. Besides, one of the significant research questions is to investigate whether the removal of the intermediary in the blockchain technology is turns out in reality to be a redefining of intermediary in disguise as each blockchain system retains control over the system by redefining their roles within the technology. Finally, I hope and believe that by writing my thesis, I have made my humble contribution to enhancing the security and privacy of IoT authentication and access control by utilising blockchain and distributed ledger technologies. It is visible that the contributions submitted in this thesis satisfy the security requirements for IoT applications and meet future demands. Last but not least, it is hoped that this study shall be a source of motivation for further research into this field.

References

Meneghello, Francesca, Matteo Calore, Daniel Zucchetto, Michele Polese and Andrea Zanella (2019). “IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices”. In: *IEEE Internet of Things Journal* 6.5, pp. 8182–8201.

Levi, Albert and M Ufuk Caglayan (1997). “The problem of trusted third party in authentication and digital signature protocols”. In: *Proceedings of the 12th International Symposium on Computer and Information Sciences*. Citeseer, pp. 317–324.

Nakamoto, Satoshi (2008). “Bitcoin: A peer-to-peer electronic cash system”. In: *Decentralized Business Review*, p. 21260.

Teicher, Jordan (February 7, 2018). *The little-known story of the first IoT device*.

Ashton, Kevin et al. (2009). “That âinternet of things’ thing”. In: *RFID journal* 22.7, pp. 97–114.

Vailshery, L (2021). *Global IoT and Non-IoT Connections 2010–2025*.

Wójcicki, Krzysztof, Marta Biegańska, Beata Paliwoda and Justyna Górna (2022). “Internet of Things in Industry: Research Profiling, Application, Challenges and OpportunitiesâA Review”. In: *Energies* 15.5, p. 1806.

Barua, Arup, Md Abdullah Al Alamin, Md Shohrab Hossain and Ekram Hossain (2022). “Security and Privacy Threats for Bluetooth Low Energy in IoT and Wearable Devices: A Comprehensive Survey”. In: *IEEE Open Journal of the Communications Society*.

Raghuvanshi, Abhishek, Umesh Kumar Singh and Chirag Joshi (2022). “A review of various security and privacy innovations for IoT applications in healthcare”. In: *Advanced Healthcare Systems: Empowering Physicians with IoT-Enabled Technologies*, pp. 43–58.

Trnka, Michal, Amr S Abdelfattah, Aishwarya Shrestha, Michael Coffey and Tomas Cerny (2022). “Systematic Review of Authentication and Authorization Advancements for the Internet of Things”. In: *Sensors* 22.4, p. 1361.

Iqbal, Waseem, Haider Abbas, Mahmoud Daneshmand, Bilal Rauf and Yawar Abbas Bangash (2020). “An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security”. In: *IEEE Internet of Things Journal* 7.10, pp. 10250–10276.

Qi, Qiao, Xiaoming Chen, Caijun Zhong and Zhaoyang Zhang (2020). “Physical layer security for massive access in cellular Internet of Things”. In: *Science China Information Sciences* 63.2, pp. 1–12.

Ramesh, Reethika, Leonid Evdokimov, Diwen Xue and Roya Ensafi (2022). “VPNalyzer: Systematic Investigation of the VPN Ecosystem”. In: *Network and Distributed System Security*.

Jaigirdar, Fariha Tasmin, Carsten Rudolph and Chris Bain (2019). “Can I trust the data I see? A Physician’s concern on medical data in IoT health architectures”. In: *Proceedings of the Australasian computer science week multiconference*, pp. 1–10.

Rasool, Raihan Ur, Hafiz Farooq Ahmad, Wajid Rafique, Adnan Qayyum and Junaid Qadir (2022). “Security and privacy of internet of medical things: A contemporary review in the age of surveillance, botnets, and adversarial ML”. In: *Journal of Network and Computer Applications*, p. 103332.

Ullah, Shafi, Raja Zahilah Radzi, Tulha Moaiz Yazdani, Ali Alshehri and Ilyas Khan (2022). “Types of Lightweight Cryptographies in Current Developments for Resource Constrained Machine Type Communication Devices: Challenges and Opportunities”. In: *IEEE Access* 10, pp. 35589–35604.

Ukwandu, Elochukwu, Mohamed Amine Ben-Farah, Hanan Hindy, Miroslav Bures, Robert Atkinson, Christos Tachtatzis, Ivan Andonovic and Xavier Bellekens (2022). “Cyber-security challenges in aviation industry: a review of current and future trends”. In: *Information* 13.3, p. 146.

Buterin, Vitalik et al. (2014). “A next-generation smart contract and decentralized application platform”. In: *white paper* 3.37, pp. 2–1.

Raj, Aparna and Sujala D Shetty (2021). “IoT eco-system, layered architectures, security and advancing technologies: A comprehensive survey”. In: *Wireless Personal Communications*, pp. 1–37.

Darwish, Dina (2015). “Improved layered architecture for Internet of Things”. In: *Int. J. Comput. Acad. Res.(IJCAR)* 4, pp. 214–223.

Jabraeil Jamali, Mohammad Ali, Bahareh Bahrami, Arash Heidari, Parisa Allahverdizadeh and Farhad Norouzi (2020). “IoT architecture”. In: *Towards the Internet of Things*, pp. 9–31.

Gupta, Brij B and Megha Quamara (2020a). “An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols”. In: *Concurrency and Computation: Practice and Experience* 32.21, e4946.

Dunkels, Adam and Jean-Phillipe Vasseur (2010). *Interconnecting Smart Objects with IP.* Morgan Kaufmann.

Shelby, Zach, Klaus Hartke and Carsten Bormann (2014). “The constrained application protocol (CoAP)”. In.

Rayes, Ammar and Samer Salam (2019). “IoT protocol stack: a layered view”. In: *Internet of Things From Hype to Reality*. Springer, pp. 103–154.

Winter, T et al. (2011). *Rpl: Ipv6 routing protocol for low power and lossy networks. draft-ietf-roll-rpl-19*.

Standard, OASIS (2014). “MQTT version 3.1. 1”. In: *URL http://docs. oasis-open. org/mqtt/mqtt/v3* 1.

Newell, Jared, Quazi Mamun, Md Zahidul Islam et al. (2021). “A Generalised Logical Layered Architecture for Blockchain Technology”. In: *arXiv preprint arXiv:2110.09615*.

Szabo, Nick (1996). “Smart contracts: building blocks for digital markets”. In: *EXTROPY: The Journal of Transhumanist Thought,(16)* 18.2, p. 28.

Freeman, Herbert (1961). “On the encoding of arbitrary geometric configurations”. In: *IRE Transactions on Electronic Computers* 2, pp. 260–268.

Androulaki, Elli et al. (2018). “Hyperledger fabric: a distributed operating system for permissioned blockchains”. In: *Proceedings of the thirteenth EuroSys conference*, pp. 1–15.

Al-Breiki, Hamda, Muhammad Habib Ur Rehman, Khaled Salah and Davor Svetinovic (2020). “Trustworthy blockchain oracles: review, comparison, and open research challenges”. In: *IEEE Access* 8, pp. 85675–85685.

Leiponen, Aija, Llewellyn DW Thomas and Qian Wang (2022). “The dApp economy: a new platform for distributed innovation?” In: *Innovation* 24.1, pp. 125–143.

Beck, Roman, Jacob Stenum Czepluch, Nikolaj Lollike and Simon Malone (2016). “Blockchain–the gateway to trust-free cryptographic transactions”. In.

Sheth, Harsh and Janvi Dattani (2019). “Overview of blockchain technology”. In: *Asian Journal For Convergence In Technology (AJCT) ISSN-2350-1146*.

Cachin, Christian, Marko Vukolic Sorniotti and Thomas Weigold (2016a). “Blockchain, cryptography, and consensus”. In: *IBM Res., Zürich, Switzerland, Tech. Rep* 2016.

Ferdous, Md Sadek, Mohammad Jabed Morshed Chowdhury, Mohammad A Hoque and Alan Colman (2020). “Blockchain consensus algorithms: A survey”. In: *arXiv preprint arXiv:2001.07091*.

Pahlajani, Sunny, Avinash Kshirsagar and Vinod Pachghare (2019). “Survey on private blockchain consensus algorithms”. In: *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*. IEEE, pp. 1–6.

Gao, Yue, Jinqiao Shi, Xuebin Wang, Qingfeng Tan, Can Zhao and Zelin Yin (2019). “Topology measurement and analysis on ethereum p2p network”. In: *2019 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, pp. 1–7.

Benčić, Federico Matteo and Ivana Podnar Žarko (2018). "Distributed ledger technology: Blockchain compared to directed acyclic graph". In: *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, pp. 1569–1570.

Chow, Sherman SM, Ziliang Lai, Chris Liu, Eric Lo and Yongjun Zhao (2018). "Sharding blockchain". 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, pp. 1665–1665.

Dang, Hung, Tien Tuan Anh Dinh, Dumitrel Loghin, Ee-Chien Chang, Qian Lin and Beng Chin Ooi (2019). "Towards scaling blockchain systems via sharding". In: *Proceedings of the 2019 international conference on management of data*, pp. 123–140.

Team, Z and P Barrett (2018). "The Zilliqa project: A secure, scalable Blockchain platform". In: *Zilliqa*, pp. 1–18.

Goodman, LM (2014). "Tezosâa self-amending crypto-ledger White paper". In: *URL: https://www. tezos. com/static/papers/white paper. pdf*.

Dai, Patrick, Neil Mahi, Jordan Earls and Alex Norta (2017). *Smart-contract value-transfer protocols on a distributed mobile application platform*.

Buterin, Vitalik, Diego Hernandez, Thor Kamphefner, Khiem Pham, Zhi Qiao, Danny Ryan, Juhyeok Sin, Ying Wang and Yan X Zhang (2020). "Combining GHOST and casper". In: *arXiv preprint arXiv:2003.03052*.

Cachin, Christian et al. (2016b). "Architecture of the hyperledger blockchain fabric". In: *Workshop on distributed cryptocurrencies and consensus ledgers*. Vol. 310. 4. Chicago, IL, pp. 1–4.

Guegan, Dominique (2017). "Public blockchain versus private blockhain". In.

Gupta, Maanak, Smriti Bhatt, Asma Hassan Alshehri and Ravi Sandhu (2022a). "Access Control Models in Cloud IoT Services". In: *Access Control Models and Architectures For IoT and Cyber Physical Systems*. Springer, pp. 63–96.

Alnefaie, Seham, Suhair Alshehri and Asma Cherif (2021). “A survey on access control in IoT: Models, architectures and research opportunities”. In: *International Journal of Security and Networks* 16.1, pp. 60–76.

Xiong, Shuming, Qiang Ni, Liangmin Wang and Qian Wang (2020). “SEM-ACSIT: secure and efficient multiauthority access control for IoT cloud storage”. In: *IEEE Internet of Things Journal* 7.4, pp. 2914–2927.

YOSHII, Masaki, Ryohei BANNO and Osamu MIZUNO (2020). “Performance evaluation of table-based access control list applied to IoT data distribution method using fog computing”. In: *IEICE Proceedings Series* 63.E1-1.

Qiu, Jing, Zhihong Tian, Chunlai Du, Qi Zuo, Shen Su and Binxing Fang (2020). “A survey on access control in the age of internet of things”. In: *IEEE Internet of Things Journal* 7.6, pp. 4682–4696.

Al-Shaboti, Mohammed, Ian Welch, Aaron Chen and Muhammed Adeel Mahmood (2018). “Towards secure smart home IoT: Manufacturer and user network access control framework”. In: *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, pp. 892–899.

Langmead, Paige (2022). “Comparative Evaluation of Access Control Models”. In.

Heydari, Mohammad, Alexios Mylonas, Vasileios Katos and Dimitris Gritzalis (2019). “Towards indeterminacy-tolerant access control in iot”. In: *Handbook of Big Data and IoT Security*. Springer, pp. 53–71.

Aftab, Muhammad Umar, Ali Hamza, Ariyo Oluwasanmi, Xuyun Nie, Muhammad Shahzad Sarfraz, Danish Shehzad, Zhiguang Qin and Ammar Rafiq (2022). “Traditional and Hybrid Access Control Models: A Detailed Survey”. In: *Security and Communication Networks* 2022.

Bisma, Mariam, Farooque Azam, Yawar Rasheed and Muhammad Waseem Anwar (2020). “A Model-Driven Framework for Ensuring Role Based Access Control in IoT Devices”. In: *Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence*, pp. 455–460.

Jaikla, Tinthid, Chalee Vorakulpipat, Ekkachan Rattanalerdnusorn and Hoang Dang Hai (2019). “A secure network architecture for heterogeneous iot devices

using role-based access control". In: *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, pp. 1–5.

Gupta, Maanak, Smriti Bhatt, Asma Hassan Alshehri and Ravi Sandhu (2022b). "Introduction: Requirements for Access Control in IoT and CPS". In: *Access Control Models and Architectures For IoT and Cyber Physical Systems*. Springer, pp. 1–17.

Vijayalakshmi, K and V Jayalakshmi (2022). "A Study on Current Research and Challenges in Attribute-based Access Control Model". In: *Intelligent Data Communication Technologies and Internet of Things*, pp. 17–31.

Ameer, Safwa, James Benson and Ravi Sandhu (2022). "An Attribute-Based Approach toward a Secured Smart-Home IoT Access Control and a Comparison with a Role-Based Approach". In: *Information* 13.2, p. 60.

Xu, Ronghua, Yu Chen, Erik Blasch and Genshe Chen (2018). "A federated capability-based access control mechanism for internet of things (iots)". In: *Sensors and Systems for Space Applications XI*. Vol. 10641. SPIE, pp. 291–307.

Sivaselvan, N, Waqar Asif, Bhat K Vivekananda and Muttukrishnan Rajarajan (2020). "Authentication and Capability-based Access Control: An Integrated Approach for IoT Environment". In: *2020 12th International Conference on Communication Software and Networks (ICCSN)*. IEEE, pp. 110–117.

Dramé-Maigné, Sophie, Maryline Laurent, Laurent Castillo and Hervé Ganem (2021). "Centralized, distributed, and everything in between: Reviewing access control solutions for the iot". In: *ACM Computing Surveys (CSUR)* 54.7, pp. 1–34.

Al-Turjman, Fadi, Hadi Zahmatkesh and Ramiz Shahroze (2022). "An overview of security and privacy in smart cities' IoT communications". In: *Transactions on Emerging Telecommunications Technologies* 33.3, e3677.

Thakor, Vishal A, Mohammad Abdur Razzaque and Muhammad RA Khandaker (2021). "Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities". In: *IEEE Access* 9, pp. 28177–28193.

Gupta, Brij B and Megha Quamara (2020b). “An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols”. In: *Concurrency and Computation: Practice and Experience* 32.21, e4946.

Liyanage, Madhusanka, An Braeken, Pardeep Kumar and Mika Ylianttila (2020a). *IoT security: Advances in authentication*. John Wiley & Sons.

Mehta, Mihir and Kajal Patel (2020). “A review for IOT authentication–current research trends and open challenges”. In: *Materials Today: Proceedings*.

Kim, Jinsu and Namje Park (2019). “Lightweight knowledge-based authentication model for intelligent closed circuit television in mobile personal computing”. In: *Personal and Ubiquitous Computing*, pp. 1–9.

Sharma, Sapna and Shilpy Agrawal (2021). “Personal authentication based on vascular pattern using finger vein biometric”. In: *Journal of Discrete Mathematical Sciences and Cryptography* 24.5, pp. 1167–1178.

Sulaiman, Meor Muhammad Kamal Meor Muhammad, Mohd Fairuz Iskandar Othman, Wahidah Md Shah, Aslinda Hassan, Norhayati Harum, Ibrahim Mohammed Alseadoon et al. (2021). “An Online Voting System using Face Recognition for Campus Election”. In: *Journal of Advanced Computing Technology and Application (JACTA)* 3.1, pp. 39–46.

Alsahlani, Ahmed Yaser Fahad and Alexandru Popa (2021). “LMAAS-IoT: Lightweight multi-factor authentication and authorization scheme for real-time data access in IoT cloud-based environment”. In: *Journal of Network and Computer Applications* 192, p. 103177.

Al-Naji, Fatimah Hussain and Rachid Zagrouba (2020). “A survey on continuous authentication methods in Internet of Things environment”. In: *Computer Communications* 163, pp. 109–133.

Whitfield, Diffie and Martin E Hellman (1976). “New directions in cryptography”. In: *IEEE transactions on Information Theory* 22.6, pp. 644–654.

Hanaoka, Goichiro, Junji Shikata and Yohei Watanabe (2022). *Public-key Cryptography–PKC 2022: 25th IACR International Conference on Practice and Theory of Public-*

Key Cryptography, Virtual Event, March 8-11, 2022, Proceedings. Vol. 13178. Springer Nature.

Rivest, Ronald L, Adi Shamir and Leonard Adleman (1978). "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2, pp. 120–126.

Miller, Victor S (1985). "Use of elliptic curves in cryptography". In: *Conference on the theory and application of cryptographic techniques*. Springer, pp. 417–426.

Khalaf, Emad F and Mustafa M Kadi (2017). "A Survey of Access Control and Data Encryption for Database Security". In: *Journal of King Abdulaziz University* 28.1, pp. 19–30.

Albogami, Ohoud, Manal Alruqi, Kholood Almalki and Asia Aljahdali (2021). "Public Key Infrastructure Traditional and Modern Implementation". In: *International Journal of Network Security* 23.2, pp. 343–350.

Zimmermann, Philip (n.d.). *Why I Wrote PGP*. `https://www.philzimmermann.com/`. Accessed: 20 May 2022.

Chenchev, Ivaylo, Adelina Aleksieva-Petrova and Milen Petrov (2021). "Authentication Mechanisms and Classification: A Literature Survey". In: *Intelligent Computing*, pp. 1051–1070.

ITU, OF (n.d.). "ITU-Tx. 1500". In: *Network security* 10.1030-X (), p. 1049.

Rey, Anastasia Sergeevna (2021). "Analyzing the implementation of the identity management standards". In: pp. 112–119.

Nur, Mohammad and Yong Wang (2021). "An overview of identity relationship management in the internet of things". In: *2021 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, pp. 1–5.

Aldosary, Maha and Norah Alqahtani (2021). "Federated Identity Management (FIdM) Systems Limitation And Solutions". In: *arXiv preprint arXiv:2104.14018*.

Marillonnet, Paul, Mikaël Ates, Maryline Laurent and Nesrine Kaaniche (2021). "An Efficient User-Centric Consent Management Design for Multiservices Platforms". In: *Security and Communication Networks* 2021.

Recordon, David and Drummond Reed (2006). “OpenID 2.0: a platform for user-centric identity management”. In: *Proceedings of the second ACM workshop on Digital identity management*, pp. 11–16.

Pöhn, Daniela and Wolfgang Hommel (2020). “An overview of limitations and approaches in identity management”. In: *Proceedings of the 15th International Conference on Availability, Reliability and Security*, pp. 1–10.

Houhamdi, Zina and Belkacem Athamena (2020). “Identity identification and management in the internet of things”. In.

Wang, Ding and Weizhi Meng (2021). “Security and Privacy Challenges in Internet of Things”. In: *Mobile Networks and Applications*, pp. 1–3.

Thilakarathne, Navod Neranjan (2020). “Security and privacy issues in iot environment”. In: *International Journal of Engineering and Management Research* 10.

Patil, Pradnya, M Sangeetha and Vidhyacharan Bhaskar (2021). “Blockchain for IoT access control, security and privacy: a review”. In: *Wireless Personal Communications* 117.3, pp. 1815–1834.

Li, Tian, Huaqun Wang, Debiao He and Jia Yu (2022a). “Blockchain-based privacy-preserving and rewarding private data sharing for IoT”. In: *IEEE Internet of Things Journal*.

Dai, Hong-Ning, Zibin Zheng and Yan Zhang (2019). “Blockchain for Internet of Things: A survey”. In: *IEEE Internet of Things Journal* 6.5, pp. 8076–8094.

Wang, Haiyan and Jiawei Zhang (2019). “Blockchain based data integrity verification for large-scale IoT data”. In: *IEEE Access* 7, pp. 164996–165006.

Abdelmaboud, Abdelzahir, Abdelmuttlib Ibrahim Abdalla Ahmed, Mohammed Abaker, Taiseer Abdalla Elfadil Eisa, Hashim Albasheer, Sara Abdelwahab Ghorashi and Faten Khalid Karim (2022). “Blockchain for IoT Applications: Taxonomy, Platforms, Recent Advances, Challenges and Future Research Directions”. In: *Electronics* 11.4, p. 630.

Fan, Yongkai, Xiaodong Lin, Wei Liang, Jinghan Wang, Gang Tan, Xia Lei and Lei Jing (2022). “TraceChain: A blockchain-based scheme to protect data confid-

entiality and traceability". In: *Software: Practice and Experience* 52.1, pp. 115–129.

Li, Daming, Lianbing Deng, Zhiming Cai and Alireza Souri (2022b). "Blockchain as a service models in the Internet of Things management: systematic review". In: *Transactions on Emerging Telecommunications Technologies* 33.4, e4139.

Lone, Auqib Hamid and Roohie Naaz (2021). "Applicability of Blockchain smart contracts in securing Internet and IoT: a systematic literature review". In: *Computer Science Review* 39, p. 100360.

Miraz, Mahdi H and Maaruf Ali (2020). "Integration of blockchain and IoT: an enhanced security perspective". In: *arXiv preprint arXiv:2011.09121*.

Hellani, Houssein, Layth Sliman, Abed Ellatif Samhat and Ernesto Exposito (2021). "On Blockchain Integration with Supply Chain: Overview on Data Transparency". In: *Logistics* 5.3, p. 46.

Liu, Yang, Debiao He, Mohammad S Obaidat, Neeraj Kumar, Muhammad Khurram Khan and Kim-Kwang Raymond Choo (2020). "Blockchain-based identity management systems: A review". In: *Journal of network and computer applications* 166, p. 102731.

Al-Bassam, Mustafa (2017). "SCPKI: A smart contract-based PKI and identity system". In: *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, pp. 35–40.

Liu, Yuan, Zheng Zhao, Guibing Guo, Xingwei Wang, Zhenhua Tan and Shuang Wang (2017). "An identity management system based on blockchain". In: *2017 15th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, pp. 44–4409.

Axon, Louise (2015). "Privacy-awareness in blockchain-based PKI". In: *Cdt technical paper series* 21, p. 15.

Augot, Daniel, Hervé Chabanne, Olivier Clémot and William George (2017). "Transforming face-to-face identity proofing into anonymous digital identity using the bitcoin blockchain". In: *2017 15th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, pp. 25–2509.

Hardjono, Thomas and Alex Pentland (2019). “Verifiable anonymous identities and access control in permissioned blockchains”. In: *arXiv preprint arXiv:1903.04584*.

Halpin, Harry (2017). “NEXTLEAP: Decentralizing identity with privacy for secure messaging”. In: *Proceedings of the 12th International Conference on Availability, Reliability and Security*, pp. 1–10.

Azouvi, Sarah, Mustafa Al-Bassam and Sarah Meiklejohn (2017). “Who am i? secure identity registration on distributed ledgers”. In: *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, pp. 373–389.

Naik, Nitin and Paul Jenkins (2020). “uPort open-source identity management system: An assessment of self-sovereign identity and user-centric data platform built on blockchain”. In: *2020 IEEE International Symposium on Systems Engineering (ISSE)*. IEEE, pp. 1–7.

ShoCard (2020). “ShoBadge: Secure Enterprise Identity Authentication Built Using the Blockchain”. In.

Bitnation (2018). *The Internet of sovereignty*.

McCabe, Heather A and Sheila S Kennedy (2014). “Civic identity, civic deficit: The unanswered questions”. In: *Journal of civic Literacy* 1.1.

Decentralized, JolocomâA (2021). *Open Source Solution for Digital Identity and Access Management*.

Khovratovich, Dmitry and Jason Law (2017). “Sovrin: digital identities in the blockchain era”. In: *Github Commit by jasonalaw October* 17, pp. 38–99.

Alliance, ID2020 (2018). “Committed to improving lives through digital identity”. In: *ID2020 Alliance Introductory document*.

Toorani, Mohsen and Christian Gehrmann (2021). “A decentralized dynamic pki based on blockchain”. In: *Proceedings Of the 36th Annual ACM Symposium On Applied Computing*, pp. 1646–1655.

Shi, Jia, Xuewen Zeng and Rui Han (2022). “A Blockchain-Based Decentralized Public Key Infrastructure for Information-Centric Networks”. In: *Information* 13.5, p. 264.

Preukschat, Alex and Drummond Reed (2021). *Self-sovereign identity*. Manning Publications.

Laatikainen, Gabriella, Taija Kolehmainen and Pekka Abrahamsson (2021). "Self-sovereign identity ecosystems: benefits and challenges". In: *Scandinavian Conference on Information Systems*. Association for Information Systems.

Ourad, Abdallah Zoubir, Boutheyna Belgacem and Khaled Salah (2018a). "Using blockchain for IOT access control and authentication management". In: *International Conference on Internet of Things*. Springer, pp. 150–164.

Esposito, Christian, Massimo Ficco and Brij Bhooshan Gupta (2021). "Blockchain-based authentication and authorization for smart city applications". In: *Information Processing & Management* 58.2, p. 102468.

Alilwit, Norah (2020). "Authentication Based on Blockchain". In: *2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC)*. IEEE, pp. 1–6.

Zhang, Lin, Hong Li, Limin Sun, Zhiqiang Shi and Yunhua He (2017). "Poster: towards fully distributed user authentication with blockchain". In: *2017 IEEE Symposium on Privacy-Aware Computing (PAC)*. IEEE, pp. 202–203.

Deep, Gaurav, Rajni Mohana, Anand Nayyar, P Sanjeevikumar and Eklas Hossain (2019). "Authentication protocol for cloud databases using blockchain mechanism". In: *Sensors* 19.20, p. 4444.

Kim, Hyun-Woo and Young-Sik Jeong (2018). "Secure authentication-management human-centric scheme for trusting personal resource information on mobile cloud computing with blockchain". In: *Human-centric Computing and Information Sciences* 8.1, pp. 1–13.

Huh, Jun-Ho and Kyungryong Seo (2019). "Blockchain-based mobile fingerprint verification and automatic log-in platform for future computing". In: *The Journal of Supercomputing* 75.6, pp. 3123–3139.

Widick, Logan, Ishan Ranasinghe, Ram Dantu and Srikanth Jonnada (2019). "Blockchain based authentication and authorization framework for remote collab-

oration systems". In: *2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE, pp. 1–7.

Hammi, Mohamed Tahar, Badis Hammi, Patrick Bellot and Ahmed Serhrouchni (2018). "Bubbles of Trust: A decentralized blockchain-based authentication system for IoT". In: *Computers & Security* 78, pp. 126–142.

Lin, Chao, Debiao He, Xinyi Huang, Kim-Kwang Raymond Choo and Athanasios V Vasilakos (2018). "BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0". In: *Journal of network and computer applications* 116, pp. 42–52.

Ouaddah, Aafaf, Anas Abou Elkalam and Abdellah Ait Ouahman (2016). "Fair-Access: a new Blockchain-based access control framework for the Internet of Things". In: *Security and communication networks* 9.18, pp. 5943–5964.

Niu, Yukun, Lingbo Wei, Chi Zhang, Jianqing Liu and Yuguang Fang (2017). "An anonymous and accountable authentication scheme for Wi-Fi hotspot access with the Bitcoin blockchain". In: *2017 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, pp. 1–6.

Sanda, Tomoyuki and Hiroyuki Inaba (2016). "Proposal of new authentication method in Wi-Fi access using Bitcoin 2.0". In: *2016 IEEE 5th Global Conference on Consumer Electronics*. IEEE, pp. 1–5.

Mohsin, AH, AA Zaidan, BB Zaidan, Osamah Shihab Albahri, Ahmed Shihab Albahri, MA Alsalem and KI Mohammed (2019). "Blockchain authentication of network applications: Taxonomy, classification, capabilities, open challenges, motivations, recommendations and future directions". In: *Computer Standards & Interfaces* 64, pp. 41–60.

Lee, Jong-Hyouk (2017). "BIDaaS: Blockchain based ID as a service". In: *IEEE Access* 6, pp. 2274–2278.

Manzoor, Ahsan, Madhsanka Liyanage, An Braeke, Salil S Kanhere and Mika Ylianttila (2019). "Blockchain based proxy re-encryption scheme for secure IoT data sharing". In: *2019 IEEE international conference on blockchain and cryptocurrency (ICBC)*. IEEE, pp. 99–103.

Ma, Mingxin, Guozhen Shi and Fenghua Li (2019a). “Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the IoT scenario”. In: *IEEE access* 7, pp. 34045–34059.

Almadhoun, Randa, Maha Kadadha, Maya Alhemeiri, Maryam Alshehhi and Khaled Salah (2018). “A user authentication scheme of IoT devices using blockchain-enabled fog nodes”. In: *2018 IEEE/ACS 15th international conference on computer systems and applications (AICCSA)*. IEEE, pp. 1–8.

Nguyen, Truc DT, Hoang-Anh Pham and My T Thai (2018). “Leveraging blockchain to enhance data privacy in IoT-based applications”. In: *International Conference on Computational Social Networks*. Springer, pp. 211–221.

Bao, Zijian, Wenbo Shi, Debiao He and Kim-Kwang Raymond Chood (2018). “IoTChain: A three-tier blockchain-based IoT security architecture”. In: *arXiv preprint arXiv:1806.02008*.

Riabi, Imen, Hella Kaffel Ben Ayed and Leila Azzouz Saidane (2019). “A survey on blockchain based access control for internet of things”. In: *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, pp. 502–507.

Ding, Sheng, Jin Cao, Chen Li, Kai Fan and Hui Li (2019). “A novel attribute-based access control scheme using blockchain for IoT”. In: *IEEE Access* 7, pp. 38431–38441.

Zhang, Yan, Bing Li, Ben Liu, Jiaxin Wu, Yazhou Wang and Xia Yang (2020). “An attribute-based collaborative access control scheme using blockchain for IoT devices”. In: *Electronics* 9.2, p. 285.

Novo, Oscar (2018). “Blockchain meets IoT: An architecture for scalable access management in IoT”. In: *IEEE internet of things journal* 5.2, pp. 1184–1195.

Hwang, DongYeop, JungYong Choi and Ki-Hyung Kim (2018). “Dynamic access control scheme for iot devices using blockchain”. In: *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, pp. 713–715.

Ma, Mingxin, Guozhen Shi and Fenghua Li (2019b). “Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the IoT scenario”. In: *IEEE access* 7, pp. 34045–34059.

Fotiou, Nikos, Iakovos Pittaras, Vasilios A Siris, Spyros Voulgaris and George C Polyzos (2019). “Secure IoT access at scale using blockchains and smart contracts”. In: *2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE, pp. 1–6.

Ourad, Abdallah Zoubir, Boutheyna Belgacem and Khaled Salah (2018b). “Using blockchain for IOT access control and authentication management”. In: *International Conference on Internet of Things*. Springer, pp. 150–164.

Al Breiki, Hamda, Lamees Al Qassem, Khaled Salah, Muhammad Habib Ur Rehman and Davor Sevtinovic (2019). “Decentralized access control for IoT data using blockchain and trusted oracles”. In: *2019 IEEE International Conference on Industrial Internet (ICII)*. IEEE, pp. 248–257.

Pinno, Otto Julio Ahlert, Andre Ricardo Abed Gregio and Luis CE De Bona (2017). “Controlchain: Blockchain as a central enabler for access control authorizations in the iot”. In: *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, pp. 1–6.

Yu, Guangsheng, Xuan Zha, Xu Wang, Wei Ni, Kan Yu, Ping Yu, J Andrew Zhang, Ren Ping Liu and Y Jay Guo (2020). “Enabling attribute revocation for fine-grained access control in blockchain-IoT systems”. In: *IEEE Transactions on Engineering Management* 67.4, pp. 1213–1230.

Bera, Basudeb, Durbadal Chattaraj and Ashok Kumar Das (2020). “Designing secure blockchain-based access control scheme in IoT-enabled Internet of Drones deployment”. In: *Computer Communications* 153, pp. 229–249.

Sultana, Tanzeela, Ahmad Almogren, Mariam Akbar, Mansour Zuair, Ibrar Ullah and Nadeem Javaid (2020). “Data sharing system integrating access control mechanism using blockchain-based smart contracts for IoT devices”. In: *Applied Sciences* 10.2, p. 488.

Cekerevac, Zoran, Zdenek Dvorak, Ludmila Prigoda and Petar Cekerevac (2017). "Internet of things and the man-in-the-middle attacks–security and economic risks". In: *MEST Journal* 5.2, pp. 15–25.

Peña-López, Ismael et al. (2005). "ITU Internet report 2005: the internet of things". In.

Nižetić, Sandro, Petar Šolić, Diego López-de-Ipiña González-de, Luigi Patrono et al. (2020). "Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future". In: *Journal of Cleaner Production* 274, p. 122877.

Liyanage, Madhusanka, An Braeken, Pardeep Kumar and Mika Ylianttila (2020b). *IoT security: Advances in authentication*. John Wiley & Sons.

Duncan, Richard (2001). "An overview of different authentication methods and protocols". In: *SANS Institute*.

Koundinya, Vijay and Shwetha Baliga (2020). "A Review on Single Sign on as an Authentication". In: *System* 7.06.

Mail, AOL and Drop Box (2017). "Two factor authentication". In.

Sezer, Sakir (2018). "T1C: IoT Security:-Threats, security challenges and IoT security research and technology trends". In: *2018 31st IEEE International System-on-Chip Conference (SOCC)*. IEEE, pp. 1–2.

Amrutiya, Varun, Siddhant Jhamb, Pranjal Priyadarshi and Ashutosh Bhatia (2019). "Trustless two-factor authentication using smart contracts in blockchains". In: *2019 international conference on information networking (ICOIN)*. IEEE, pp. 66–71.

Alharbi, E and Daniyal Alghazzawi (2019). "Two factor authentication framework using otp-sms based on blockchain". In: *Transactions on Machine Learning and Artificial Intelligence* 7.3, pp. 17–27.

Danish, Syed Muhammad, Marios Lestas, Waqar Asif, Hassaan Khaliq Qureshi and Muttukrishnan Rajarajan (2019). "A lightweight blockchain based two factor authentication mechanism for LoRaWAN join procedure". In: *2019 IEEE*

International Conference on Communications Workshops (ICC Workshops). IEEE, pp. 1–6.

Ribeiro, Victor, Raimir Holanda, Alex Ramos and Joel JPC Rodrigues (2020). "Enhancing key management in LoRaWAN with permissioned blockchain". In: *Sensors* 20.11, p. 3068.

Burr, William E, Donna F Dodson, William T Polk et al. (2006). *Electronic authentication guideline*. Citeseer.

Remix (n.d.). `https://http://remix.ethereum.org/`. Accessed: 20 Feb 2022.

The world's most powerful blockchain development suite (n.d.). `https://https://infura.io/`. Accessed: 20 Feb 2022.

BrankoviÄ, Aleksandar (2021). *Delays and disruptions in delivering SMS OTP in a pandemic â How big of a threat are they for SMS-based 2FA?*

Thatha, Rakesh (2012). *Limitations of two factor authentication (2FA) technology*.

Siim, Janno (2017). "Proof-of-stake". In: *Research Seminar in Cryptography*.

Wood, Gavin (2015). "PoA private chains". In: *Github*.

Coin Market Cap (n.d.). *Ethereum*. `https://coinmarketcap.com/currencies/ethereum/`. Accessed: Oct. 15, 2021.

ConsenSys (2019). *Gnosis Wallet*. `https://github.com/Gnosis/MultiSigWallet`.

Capital, Unchained (2019). *TrezorMultisig2of3*. `https://www.coindesk.com/markets/2018/06/20/bithumb-31-million-crypto-exchange-hack-what-we-know-and-dont/`.

Homoliak, Ivan, Dominik Breitenbacher, Ondrej Hujnak, Pieter Hartel, Alexander Binder and Pawel Szalachowski (2020). "SmartOTPs: An air-gapped 2-factor authentication for smart-contract wallets". In: *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, pp. 145–162.

Yokotani, Tetsuya and Yuya Sasaki (2016). "Comparison with HTTP and MQTT on required network resources for IoT". In: *2016 international conference on control, electronics, renewable energy and communications (ICCEREC)*. IEEE, pp. 1–6.

Jaikar, Sagar P and Kamatchi R Iyer (2018). “A survey of messaging protocols for IOT systems”. In: *International Journal of Advanced in Management, Technology and Engineering Sciences* 8.2, pp. 510–514.

Gay, David, Philip Levis, Robert Von Behren, Matt Welsh, Eric Brewer and David Culler (2003). “The nesC language: A holistic approach to networked embedded systems”. In: *Acm Sigplan Notices* 38.5, pp. 1–11.

Hughes, Danny, Klaas Thoelen, Wouter Horré, Nelson Matthys, Javier Del Cid, Sam Michiels, Christophe Huygens and Wouter Joosen (2009). “LooCI: a loosely-coupled component infrastructure for networked embedded systems”. In: *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia*, pp. 195–203.

Tobin, Andrew and Drummond Reed (2016). “The inevitable rise of self-sovereign identity”. In: *The Sovrin Foundation* 29.2016.

Longley, D, M Sporny, B Zundel, D Burnett and G Noble (2019). “Verifiable credentials data model 1.0”. In: *W3C recommendation*. W3C.

Reed, Drummond, Manu Sporny, Dave Longley, Christopher Allen, Ryan Grant, Markus Sabadello and Jonathan Holt (2020). “Decentralized identifiers (dids) v1. 0: Core architecture, data model, and representations”. In: *W3C Working Draft* 8.

Samaila, Musa G, Miguel Neto, Diogo AB Fernandes, Mário M Freire and Pedro RM Inácio (2018). “Challenges of securing Internet of Things devices: A survey”. In: *Security and Privacy* 1.2, e20.

Dierks, Tim and Eric Rescorla (2008). “The transport layer security (TLS) protocol version 1.2”. In.

Patel, Chintan and Nishant Doshi (2020). “A novel MQTT security framework in generic IoT model”. In: *Procedia Computer Science* 171, pp. 1399–1408.

Pahlevi, Rizka Reza, Parman Sukarno and Bayu Erfianto (2019). “Implementation of event-based dynamic authentication on MQTT protocol”. In: *Jurnal Rekayasa Elektrika* 15.2.

Lohachab, Ankur et al. (2019). “ECC based inter-device authentication and authorization scheme using MQTT for IoT networks”. In: *Journal of Information Security and Applications* 46, pp. 1–12.

Bali, Ranbir Singh, Fehmi Jaafar and Pavol Zavarasky (2019). “Lightweight authentication for MQTT to improve the security of IoT communication”. In: *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, pp. 6–12.

Cruz-Piris, Luis, Diego Rivera, Ivan Marsa-Maestre, Enrique De La Hoz and Juan R Velasco (2018). “Access control mechanism for IoT environments based on modelling communication procedures as resources”. In: *Sensors* 18.3, p. 917.

Buccafurri, Francesco and Celeste Romolo (2019). “A blockchain-based OTP-authentication scheme for constrainded IoT devices using MQTT”. In: *Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control*, pp. 1–5.

Buccafurri, Francesco, Vincenzo De Angelis and Roberto Nardone (2020). “Securing MQTT by blockchain-based otp authentication”. In: *Sensors* 20.7, p. 2002.

Lundkvist, Christian, Rouven Heck, Joel Torstensson, Zac Mitton and Michael Sena (2017). “Uport: A platform for self-sovereign identity”. In: *URL: https://whitepaper. uport. me/uPort_ whitepaper_DRAFT20170221. pdf*.

Remix (n.d.). `https://http://remix.ethereum.org/`. Accessed: 20 Feb 2022.

Collina, M (2013). *Mosca: the mqtt server for node. js that can be backed up by amqp redis zeromq or just mqtt*.

paho (n.d.). `https://https://www.eclipse.org/paho/`. Accessed: 20 Feb 2022.

BC-of-Every-Thing (2021). *A Blockchain-based Authentication and authorisation mechanism for MQTT*. `https://https://github.com/BC-of-Every-Thing/DecentralisedMQTT`.

Coinmarketcap (2022). “All cryptocurrencies”. In.

ABDELRAZIG ABUBAKAR, MWRWAN, Zakwan Jaroucheh, Ahmed Al-Dubai and Xiaodong Liu (2021). “Blockchain-based identity and authentication scheme for MQTT protocol”. In: *2021 The 3rd International Conference on Blockchain Technology*, pp. 73–81.

Javaid, Mohd and Ibrahim Haleem Khan (2021). “Internet of Things (IoT) enabled healthcare helps to take the challenges of COVID-19 Pandemic”. In: *Journal of Oral Biology and Craniofacial Research* 11.2, pp. 209–214.

Tariq, Noshina, Ayesha Qamar, Muhammad Asim and Farrukh Aslam Khan (2020). “Blockchain and Smart Healthcare Security: A Survey”. In: *Procedia Computer Science* 175, pp. 615–620.

Ghaffari, Fariba, Emmanuel Bertin, Julien Hatin and Noel Crespi (2020). “Authentication and Access Control based on Distributed Ledger Technology: A survey”. In: *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*. IEEE, pp. 79–86.

Ometov, Aleksandr et al. (2021). “A survey on wearable technology: History, state-of-the-art and current challenges”. In: *Computer Networks*, p. 108074.

Al-Sarawi, Shadi, Mohammed Anbar, Rosni Abdullah and Ahmad B Al Hawari (2020). “Internet of Things Market Analysis Forecasts, 2020–2030”. In: *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. IEEE, pp. 449–453.

Newaz, AKM, Amit Kumar Sikder, Mohammad Ashiqur Rahman and A Selcuk Uluagac (2020). “A survey on security and privacy issues in modern healthcare systems: Attacks and defenses”. In: *arXiv preprint arXiv:2005.07359*.

Xue, Yukang (2019). “A review on intelligent wearables: Uses and risks”. In: *Human Behavior and Emerging Technologies* 1.4, pp. 287–294.

Nahapetian, Ani (2016). “Side-channel attacks on mobile and wearable systems”. In: *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, pp. 243–247.

Zhang, Yuexin, Yang Xiang, Xinyi Huang and Li Xu (2014). “A cross-layer key establishment scheme in wireless mesh networks”. In: *European Symposium on Research in Computer Security*. Springer, pp. 526–541.

Nguyen, Kim Thuat, Nouha Oualha and Maryline Laurent (2016). “Authenticated key agreement mediated by a proxy re-encryptor for the internet of things”. In: *European symposium on research in computer security*. Springer, pp. 339–358.

Wu, Fan, Xiong Li, Lili Xu, Saru Kumari, Marimuthu Karuppiah and Jian Shen (2017). “A lightweight and privacy-preserving mutual authentication scheme for wearable devices assisted by cloud server”. In: *Computers & Electrical Engineering* 63, pp. 168–181.

Joshitta, R Shantha Mary and L Arockiam (2017). “Device authentication mechanism for IoT enabled healthcare system”. In: *2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)*. IEEE, pp. 1–6.

Diez, Fidel Paniagua, Diego Suárez Touceda, José María Sierra Cámara and Sherali Zeadally (2019). “Lightweight Access Control System for Wearable Devices”. In: *IT Professional* 21.1, pp. 50–58.

Guo, Hao, Wanxin Li, Mark Nejad and Chien-Chung Shen (2019). “Access control for electronic health records with hybrid blockchain-edge architecture”. In: *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE, pp. 44–51.

Yue, Xiao, Huiju Wang, Dawei Jin, Mingqiang Li and Wei Jiang (2016). “Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control”. In: *Journal of medical systems* 40.10, pp. 1–8.

Dwork, Cynthia and Moni Naor (1992). “Pricing via processing or combatting junk mail”. In: *Annual international cryptology conference*. Springer, pp. 139–147.

Croman, Kyle et al. (2016). “On scaling decentralized blockchains”. In: *International conference on financial cryptography and data security*. Springer, pp. 106–125.

Xiao, Yang, Ning Zhang, Wenjing Lou and Y Thomas Hou (2020). “A survey of distributed consensus protocols for blockchain networks”. In: *IEEE Communications Surveys & Tutorials* 22.2, pp. 1432–1465.

Larimer, Daniel (2014). “Delegated proof-of-stake (dpos)”. In: *Bitshare whitepaper* 81, p. 85.

King, Sunny and Scott Nadal (2012). “Ppcoin: Peer-to-peer crypto-currency with proof-of-stake”. In: *self-published paper, August* 19.1.

Popov, Serguei (2016). “The Tangle (Whitepaper)”. In: *available at: Iota. org*.

Poelstra, Andrew et al. (2014). “Distributed consensus from proof of stake is impossible”. In: *Self-published Paper*.

Ismail, Leila and Huned Materwala (2019). “A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions”. In: *Symmetry* 11.10, p. 1198.

Lin, Iuon-Chang and Tzu-Chun Liao (2017). “A survey of blockchain security issues and challenges.” In: *Int. J. Netw. Secur.* 19.5, pp. 653–659.

Kelsey, John, Luís TAN Brandão, Rene Peralta and Harold Booth (2019). *A reference for randomness beacons: Format and protocol version 2*. Tech. rep. National Institute of Standards and Technology.

Hanke, Timo, Mahnush Movahedi and Dominic Williams (2018). “Dfinity technology overview series, consensus system”. In: *arXiv preprint arXiv:1805.04548*.

Boneh, Dan, Ben Lynn and Hovav Shacham (2001). “Short signatures from the Weil pairing”. In: *International conference on the theory and application of cryptology and information security*. Springer, pp. 514–532.

Shang, Yilun (2018). “Hybrid consensus for averager–copier–voter networks with non-rational agents”. In: *Chaos, Solitons & Fractals* 110, pp. 244–251.

Chen, Shun, Daniel WC Ho and Ming Liu (2017). “Consensus protocol for multiple delta operator systems”. In: *Systems & Control Letters* 107, pp. 1–8.

Cheng, Zhuan, Gang Wu, Hao Wu, Muxing Zhao, Liang Zhao and Qingfeng Cai (2018). “A new hybrid consensus protocol: Deterministic proof of work”. In: *arXiv preprint arXiv:1808.04142*.

Eyal, Ittay, Adem Efe Gencer, Emin Gün Sirer and Robbert Van Renesse (2016). “Bitcoin-ng: A scalable blockchain protocol”. In: *13th {USENIX} symposium on networked systems design and implementation ({NSDI} 16)*, pp. 45–59.

Cicada (2016). *Cicada A Distributed Direct Democracy and Decentralized Application Platform*. http://iamcicada.com/whitepaper/.

Sanders, Cedric and Thomas Liebig (2019). "Knowledge Discovery on Blockchains: Challenges and Opportunities". In: *arXiv preprint arXiv:1904.07104*.

Armknecht, Frederik, Jens-Matthias Bohli, Ghassan O Karame and Wenting Li (2017). "Sharding pow-based blockchains via proofs of knowledge". In.

Lundbæk, Leif-Nissen, Daniel Janes Beutel, Michael Huth and Laurence Kirk (2018). "Practical proof of kernel work & distributed adaptiveness". In: *manuscript Version* 1.

Gilad, Yossi, Rotem Hemo, Silvio Micali, Georgios Vlachos and Nickolai Zeldovich (2017). "Algorand: Scaling byzantine agreements for cryptocurrencies". In: *Proceedings of the 26th symposium on operating systems principles*, pp. 51–68.

Sompolinsky, Yonatan and Aviv Zohar (2015). "Secure high-rate transaction processing in bitcoin". In: *International Conference on Financial Cryptography and Data Security*. Springer, pp. 507–527.

Gennaro, Rosario, Stanislaw Jarecki, Hugo Krawczyk and Tal Rabin (2002). "Revisiting the distributed key generation for discrete-log based Cryptosystems". In: *RSA Security'03*.

Kevoulee, H. (2019). *hyper$_d$ist$_p$rob.py*. https://https://github.com/tarilabs/modelling/blob/master/utils/hyper_dist_prob.py.

Biswas, Sujit, Kashif Sharif, Fan Li, Sabita Maharjan, Saraju P Mohanty and Yu Wang (2019). "PoBT: A lightweight consensus algorithm for scalable IoT business blockchain". In: *IEEE Internet of Things Journal* 7.3, pp. 2343–2355.

Dziembowski, Stefan, Sebastian Faust, Vladimir Kolmogorov and Krzysztof Pietrzak (2015). "Proofs of space". In: *Advances in Cryptology–CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*. Springer, pp. 585–605.

Puthal, Deepak, Saraju P Mohanty, Venkata P Yanambaka and Elias Kougianos (2020). "Poah: A novel consensus algorithm for fast scalable private blockchain for large-scale iot frameworks". In: *arXiv preprint arXiv:2001.07297*.

Li, Aiya, Xianhua Wei and Zhou He (2020). “Robust proof of stake: A new consensus protocol for sustainable blockchain systems”. In: *Sustainability* 12.7, p. 2824.

Abubakar, Mwrwan, Zakwan Jaroucheh, Ahmed Al-Dubai and Bill Buchanan (2020). “PoNW: A Secure and Scalable Proof-of-Notarized-Work Based Consensus Mechanism”. In: *Proceedings of the 2020 4th International Conference on Vision, Image and Signal Processing*, pp. 1–8.

Buterin, Vitalik (2016). “Ethereum: platform review”. In: *Opportunities and challenges for private and consortium blockchains* 45.

Xu, Ronghua, Yu Chen, Erik Blasch and Genshe Chen (2019). “Microchain: A hybrid consensus mechanism for lightweight distributed ledger for IoT”. In: *arXiv preprint arXiv:1909.10948*.

paper, White (2018). *PIVX White Papers*. `https://pivx.org/whitepaper`.

Chen, Jing and Silvio Micali (2016). “Algorand”. In: *arXiv preprint arXiv:1607.01341*.

Documentation, Decred (2016). *Introduction to Decred Governance*. `https://docs.decred.org/governance/overview/`.

Gauld, Seán, Franz von Ancoina and Robert Stadler (2017). “The burst dymaxion”. In: *An Arbitrary Scalable, Energy Efficient and Anonymous Transaction Network Based on Colored Tangles, abrufbar unter< www. burst-coin. org/wp-content/uploads/2017/07/The-Burst-Dymaxion-1.00.pdf> oder< https://perma. cc/UP3B-RRWG*.

Xiao, Bingbing, Chenguang Jin, Zheng Li, Bingnan Zhu, Xiaoruo Li and Dong Wang (2021). “Proof of Importance: A Consensus Algorithm for Importance Based on Dynamic Authorization”. In: *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pp. 510–513.