# [ Security and Privacy Preserving Schemes in Smart Homes using Blockchain]

**by [ Amjad Mohammed Qashlan]**

Thesis submitted in fulfilment of the requirements for the degree of

**[ Doctor of Philosophy]**

under the supervision of [ **Dr. Priyadarsi Nanda ( Principle supervisor)**
**Dr. Manoranjan Mohanty ( Co-supervisor)**]

University of Technology Sydney
Faculty of [ Engineering and Information Technology]

[ November 2021 ]

# Certificate of Original Authorship

I, *Amjad Qashlan* declare that this thesis, is submitted in fulfilment of the requirements for the award of *Doctor of Philosophy* in the *School of Electrical and Data Engineering /Faculty of Engineering and Information Technology*, at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

SIGNATURE: ________________________________

[Amjad Qashlan]

DATE: 30th November, 2021

# Abstract

Emerging technologies such as the Internet of Things, sensors, and communication networks have been integrated into traditional homes to provide a wide range of smart home services to simplify and improve people‚Äôs lifestyles. However, as the Internet of Things has grown in popularity, so have the concerns it poses. As a result, concerns like data privacy, security, and decentralisation of IoT systems present substantial threats to the future of smart home IoTs.

This thesis presents efforts towards a blockchain-based smart home framework which supports data confidentiality, differential privacy, and robustness. The thesis achieves three novel contributions. We first deploy a private blockchain using Ethereum smart contracts for a smart home to ensure only the homeowner can access and monitor home appliances. The smart contracts are designed to allow devices to communicate without the need for a trusted third party. Our prototype demonstrates three key elements of blockchain-based smart security solutions for smart home applications: smart contracts, blockchain-based access control, and the performance evaluation of the proposed scheme.

Next, we propose an authentication scheme that integrates attribute-based access control using smart contracts with an ERC-20 Token (Ethereum Request For Comments) and edge computing to construct a secure framework for IoT devices in a smart home system. The edge server provides scalability to the system by offloading heavier computation tasks to edge servers. We present the system architecture and design and discuss various aspects of testing and implementing smart contracts.

Finally, we conduct a performance evaluation to demonstrate the feasibility and efficiency of the proposed scheme. The core features that blockchain technology is leveraged upon are a trust-less environment, immutability and transparency, which come at the cost of a lack of data privacy. Therefore, we propose a privacy-preserving architecture for smart home-based blockchain. The architecture utilises differential privacy machine learning algorithm to send private IoT smart home data to the cloud and achieve data privacy. The main objective of the model is to protect privacy with high accuracy when aggregating the data from traffic analysis, linking and mining attacks by adding Gaussian noise. The implementation of our model ensures better accuracy and improved model utility. The goal of the privacy protection scheme used in our architecture is to enable smart home data to be used without disclosing privacy and provide published data to different service providers with lower information loss and higher data utility.

# DEDICATION

*To my honorable parents ( Mohammed & Mona ), my soulmate (Hesham) my lovely little angels ( Saja, Yosif, Salem and Moaaz )*
*and to the memory of my Father-in-law (Salem), I wish we were around in your last days.*
. . .

# Acknowledgments

Firstly, I would like to express my sincere gratitude to Almighty God for guiding me and blessing me with physical and mental strength along this PhD journey.

I would like to express my sincere gratitude to my supervisor, Dr. Priyadarsi Nanda, whose expertise, understanding, and patience added considerably to my graduate experience. I am greatly thankful for his continuous encouragement that guided me throughout this journey of becoming a successful researcher. Dr. Nanda's guidance, patience, and continuous support were essential to the completion of this thesis. His knowledge and invaluable feedback immensely helped me go forward in my research. I am greatly indebted to him for helping me get through the challenging times. I would like to thank my Co-supervisor, Dr. Manoranjan Mohanty, who provided valuable and expert feedback on my work. His support, co-operation, and generosity throughout the research tenure are truly undeniable.

This research would not have been possible without the support from my family. I would like to thank my parents, Mohammed and Mona, for motivating and encouraging me to be the best. Thanks for all your prayers that enlightened my journey and kept me going. Special thanks to my kids, Saja, Yosif, Salem and Moaaz, for the numerous sacrifices, continuous support, endless love and all the sorry cards that melted my heart. Many thanks to my husband, Hesham, for sharing my dream and making it come true together and for all the effort he put in to make me realise how strong and independent I am.

I would like to extend my gratitude to my sisters, Baraa and Eman, for their love, support and care all the time and, of course, many thanks to all my brothers; Bandar, Amjed, Ahmad, Ayman, Osama.

A big thanks to all my friends ( Hana, Maha, Abrar, Abeer, Amani, Enaam, Maryam, Wafaa, Aisha) for sharing PhD advice, parental tips and being my sisters in foreign land. I also wish to thank all my colleagues from the School of Electrical and Data Engineering at the University of Technology Sydney. Specifically, I would like to thank Annie, Ambar, Nisha and Upasana for creating a friendly atmosphere in the group and assisting me in whatever manner possible.

Although this journey got harder and harder as I cried in frustration or got some sleepless nights, it taught me that there will always be light at the end of the tunnel.

# List of Publications

## Journal Papers :

1. **A. Qashlan**, P. Nanda, X. He, and M. Mohanty, Privacy-preserving mecha-nism in smart home using blockchain, IEEE Access 9 (2021) 103651-103669. **Impact factor of 3.367**

2. **A. Qashlan**, P. Nanda, and M. Mohanty, Differential Privacy Model for Blockchain based Smart Home Architecture, Submitted to Future Generation Computer Systems Journal (FGCS), Under review, **Impact factor of 7.187**

## Conference Papers

1. **A. Qashlan**, P. Nanda, and X. He, "Security and privacy implementation in smart home: Attributes based access control and smart contracts," in 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, pp. 951- 958. **(Rank A)**

2. **A. Qashlan**, P. Nanda, and X. He, "Automated Ethereum smart contract for blockchain based smart home security", in Smart Systems and IoT: Innovations in Computing, Springer, 2020, pp. 313-326

# TABLE OF CONTENTS

# LIST OF FIGURES

# List of Tables

CHAPTER 1

# INTRODUCTION

This chapter introduces our research area starting with the introduction in Section 1.1, the motivation of our research in Section 1.2, and the research questions we devise in Section 1.3. Section 1.4 describes the research objective and section 1.5 details the research methodology and research contribution. Section 1.6 provides the structure and the organization of this thesis.

## 1.1 Introduction

With the rapid advancement of automation technology, home automation systems are improving their technology to take advantage of the revolution of industry 4.0, which is the current trend in manufacturing technologies for automation and data interchange. The term "automation" refers to a variety of control systems that do not require human interaction. It entails the control and operation of a wide range of equipment, machinery, and industrial mechanisms. It will lead the management and real-time monitoring of home appliances through the Internet by merging Internet of Things (IoT) apps. Every day, the IoT grows from small scale machines to enormous machines that can share data and perform tasks while people are occupied in other activities [119].

The term *home automation* or *smart home* refers to a living arrangement that is equipped with technology to monitor and support the well-being of its residents. In other terms, it is the process of automatically managing home appliances utilising a variety of control systems. In recent years, various control system techniques are being used to operate and monitor electrical home appliances and different types of sensors such as lights, fans, motion sensors, temperature sensors, and others [19]. According to researchers [119] "the primary function of a smart home is to have more intelligent monitoring and remote control so that daily activities are automated without user intervention or with the user's remote control in a more convenient, efficient, safer, and less expensive manner " (p.10488).

The smart home IoT ecosystem is undergoing a transition triggered by the advancement of information and communication technologies. Maintaining the confidentiality, integrity, and authenticity of data is a must when it comes to IoT. Hence, it is critical to meet the security considerations of the smart home network. IoT refers to the integration of devices with the Internet. Such devices are called IoT devices, and they support the expansion of Internet connection beyond the usual standard devices like computers, laptops, smartphones, etc. Some common examples of IoT devices include:

- Smart Lighting- Smart lighting can be used for energy saving by adapting lighting to the ambient conditions and by switching lights on/off according to the user needs. This can reduce the use of energy to a great extent. Saving energy also helps in reducing costs.

- Smart Door lock- Crimes such as burglary and theft can happen to anyone at anytime. Smart door locks allow you to lock your home or give access to anyone from anywhere using a smartphone application.

- Smoke Detection- This application can be used for sensing the smart home environment for healthy living. In the case of fire, it can raise an alert to a nearby fire station and a user via email/SMS, informing them about the situation.

A rising number of devices are likely to become part of the IoT in the future, each of which is designed to collect, store, and communicate a vast amount of data. These data may be utilised to offer real-time information about a person's health and finances, as well as their locations, contacts, habits, behaviours, and activities. Finally, IoT devices provide an environment in which information about each individual can be kept, analyzed, monitored, made available, and shared with other networked devices and possibly other users.

The advantages provided by the smart homes are numerous, yet they are not widely adopted either by the general mass or by older people [6]. This can be attributed to the fact that different technologies, which function as the core towards providing these services, are still in a stage of development or pending commercialization. Another reason is most of the research on smart homes focus on the underlying technologies, sensors, actuators, and various other services that they can provide [6].

According to new statistics [27], older Australians have embraced smart home appliances and have grown more tech-savvy. The survey found that, 33% of older Australians lack confidence and believe they are not tech-savvy enough to recognise the hazards associated with smart gadgets. These individuals refuse to utilise smart home devices because they do not feel they can adequately safeguard themselves. Smart appliances are vulnerable, according to 21% of respondents [27]. Therefore, one of the key challenges in smart home applications is to ensure security and privacy.

The following list summarises our research approach to smart home security and privacy. The security is achieved by offering an authentication scheme that integrates attribute-based access control using smart contracts with ERC-20 Token (Ethereum Request for Comments) and edge computing. Privacy is achieved by employing differential privacy based on machine learning and designing a privacy-preserving and secure decentralized Stochastic Gradient Descent (SGD) algorithm with our blockchain architecture.

- Security and privacy: The security of smart home systems is one of the most significant aspects to protect the privacy of smart home consumers [124]. The communication between real-life objects creates considerable challenges in trust, security, and privacy. Many security threats and attacks currently exist for the IoT. Due to the enormous size of data transmissions, adversaries such as the man-in-the-middle (MiM) attack and denial of service (DoS) and distributed denial of service (DDoS) attacks may target important data transmissions in the network.

- Scalability: Since IoT based smart homes support a massive number of devices that connect and communicate with each other, scalability is one of the significant challenges facing the middleware approach [63]. As the number of devices increases, the risk of network failure, too, increases. Hence, to perform efficiently in a small and big IoT environment, a reliable middleware is essential to control the number of devices and effectively address scaling difficulties.

- Authentication: Authentication and identity management (IdM) are a set of processes and technologies used to manage and secure access to information and resources. IdM uniquely identifies items, and authentication requires establishing the establishment of identity between two communicating parties. [137]. Because numerous users and devices must authenticate each other through trusted services, it is critical to investigate how to manage identity identification in a smart home.

- Authorization: Authorization enables one to determine if the person or object, once identified, is permitted to access the resource [86]. Access controls are commonly used to implement it. Authorization and access control are critical for establishing a secure connection between several devices and services. Unauthorized access to a system controller, particularly at the administrator level, makes the entire system insecure. Since many smart home devices may be operated by battery and networked wirelessly with a low operational duty cycle, flooding a network with requests can lead to denial of service [97].
- Privacy: Protecting the user's personal information is the primary focus of privacy protection. It could be the person's name, location, movement, or any other information about the individual (that the person does not want to share with others). Personal photos, movies and other digital data are stored in smart homes. Cameras on smart devices can be activated remotely and photos and videos can be accessed from anywhere. Furthermore, microphones on a variety of gadgets may be able to receive private phone calls and text messages. [140].

The encryption/decryption ID authentication method, cryptographic keys and hash algorithm have been proposed and examined in previous works [151]. However, all these solutions are based on centralised topology and require a trusted intermediary with huge computational and storage capacities. Thus, new solutions are needed to converge smart home security toward a decentralised model. Blockchain technology has the potential to overcome these issues because of its distributed and secure nature.

Blockchain is an underlying technology and closely linked to the cryptocurrency-Bitcoin, the digital currency introduced by the pseudonym Satoshi Nakamoto In 2008 [114]. Blockchain refers to a distributed and decentralised public ledger that keeps all transactions executed in a peer-to-peer network and shared among participants. A consensus mechanism is used to verify every transaction in the chain. This decentralised information reduce the ability to tamper with data.

The adoption of foundational technologies typically happens in different phases. Each phase is defined by the novelty of the applications and the complexity of the coordination efforts needed to make them workable. Blockchain applications are in their early phase [82]. Blockchain technology can be considered as one of the main drivers to achieve a substantial cost saving. According to Satander FinTech study, distributed ledger technology could reduce financial services infrastructure cost between US$15 billion and $20 billion per annum by 2022, providing the possibility to decommission legacy systems and infrastructure and significantly reduce IoT costs [68]. Therefore, despite the current high cost of Ethereum implementation for smart homes, it is likely that this cost will be decreased after the technology is matured over time. In our research, we mainly investigated the security and privacy aspects of data flow between various entities using blockchain based scheme for smart home application.

In this research, managing and controlling the IoT devices in the smart home system without a trusted intermediate party is critical in the canalised topology. The aim is to introduce blockchain architecture, which provides decentralised, secure peer-to-peer networks that allow non-trusted members to interact securely without a trusted authority. Moreover, there is a developing sense of urgency for blockchain based smart home framework designs to include mechanisms for the preservation of user privacy. Protecting the privacy of smart home data aims to enable data usage without disclosure as well as to reduce data loss when providing published data to service providers.

This research analyses a real-time interaction model between users at the smart home and a fully validating private blockchain node to authenticate users of smart homes and IoT devices. We also incorporate a differential privacy technique into our suggested approach to protect data privacy. Our proposed model resolves the issue of the traditional access control method based on the centralized design and meets the access control requirements in IoT by combining attributes-based access control, differential privacy, and edge computing. In this study, we created the Ethereum blockchain and several smart contracts, and our implementation shows improved performance of our suggested model. With the production of ERC-20 tokens, we achieve more fine-grained access control than the current scheme while using edge computing to reduce computation costs.

## 1.2 Research Motivation

IoT smart home devices, while providing advantages to users, also brings many threats because of their poor or inconsistent implementation of security and privacy protocols. Current IoT devices rely on a client-server model and a canalised architecture with huge computational and storage capacities. The centralized architecture has different weaknesses such as a single point of failure, a central authority and limited transparency.

Therefore, the existing approach is expensive due to the high costs of cloud server infrastructure, network equipment and the associated maintenance. Furthermore, no existing platform supports communication between all smart home devices or guarantees the interoperability of the services offered by different manufacturers on the cloud [41]. Thus, using a standard peer-to-peer decentralised approach can overcome and reduce the costs corresponding to the maintenance and infrastructure of the client-server model and share the processing and space requirements of devices on a smart home network. Blockchain architecture can provide an appropriate solution that suits the need for such a platform.

However, a lack of user privacy as a result of blockchain's wide adaption and implementation remains a major concern [48]. Data confidentiality has subsequently emerged as an issue of primary importance as smart home generated data contain sensitive content including user health information and location details. The main concerns about the integrity of blockchain are around attacks related to user privacy such as linking attacks. Such attacks utilise accessible data recorded in blocks to gain access to private information by tying the information to alternative datasets or relevant background knowledge. Attackers may have a greater chance of working out target smart home privacy data [76].

The use of differential privacy can create a level of indistinguishability in statistical blockchain data, leaving the analyst unable to predict with any certainty the accessibility of individual blockchain nodes. Differential privacy is a good fit for use in blockchain technology to preserve the individual's identity during a broadcast. While ensuring that the information remains useful for completing transactions, differential privacy can still perturb the person's identity to the network and an adversary will be unable to determine the sender's or receiver's actual identity.

This research aims to achieve the secure and private communication of IoT devices within a smart home by proposing a novel lightweight Ethereum blockchain based multi-tier, integrated with edge computing and a differential privacy enhancement model.

## 1.3 Research Questions

The thesis investigates the answers to the following research questions:

1. How can Ethereum smart contracts be used to secure access to smart home devices?

2. What benefits are achieved using a blockchain-based scheme in comparison with an existing scheme?

3. How can our proposed scheme be used to enhance data privacy?

4. How effective is our proposed scheme against current as well as future cyber threats?

   a) Developing a new scheme for the threat analysis procedure.

   b) Thoroughly testing the proposed scheme against major cyber threats.

   c) Enhancing blockchain based authentication and privacy measures.

## 1.4 Research Objective

The existing literature on blockchain architecture primarily indicate the opportunities and challenges of using blockchain in general. However, only a few researchers have investigated the smart contract applications; the majority of current research is conducted in the bitcoin environment rather than smart contracts [163]. Regarding the IoT domain, a considerable amount of current research practices have been proposed, but it lacks implementation and testing as a proof of concept, especially in relation to smart homes. Consequently, this gap will be one of the focus of this research. There is a need for a framework that supports a smart home system based on blockchain, powered by smart contracts.

Furthermore, while blockchain is regarded as the future of data storage due to its decentralised structure, several issues are still to be resolved before it is implemented in daily life scenarios. A significant parameter in blockchain applications that needs further development is data preservation and transaction privacy. That is, blockchain's distributed nature means that an individual's identity or personal information may be leaked during transactions. To overcome aforementioned issues and protect privacy, it may be useful to integrate differential privacy based on machine learning with the latest blockchain technology. Differential privacy is efficient at preserving privacy in statistical databases and real-time settings.

This research aims to investigate the potential use of an Ethereum smart contract on constrained IoT devices and establish decentralised security in a smart home environment and also, to investigate whether it is possible to implement the access management aspect of it globally. This is done by implementing and evaluating simplified smart contracts as a proof of concept over the Ethereum blockchain.

Moreover, this research examines the main threats that need to be solved in smart home environment and what are the main security vulnerabilities that will be covered by the proposed model. Furthermore, to achieve privacy preservation, the architecture utilises the differential privacy machine learning algorithm to send private smart home data to the cloud. Our research aims to provide a privacy-preserving data aggregation method in the context of smart homes that agree to provide their data to a cloud server, so this cloud can learn privately from the data produced from IoT devices inside people's homes and then deliver these data to an external entity to provide better services for home users. The proposed framework based on Ethereum smart contracts aims to achieve end-to-end secure and private communication of smart home devices within the private blockchain network. The specific objectives of this research project are as follows:

- to develop a smart home framework that allows IoT devices to communicate securely with each other.
- to simulate IoT smart home devices using Raspberry pi to build a prototype of a private Ethereum network powered by access control and smart contracts.
- to conduct a performance evaluation of the proposed prototype in terms of processing time, gas cost and block size. Also, to calculate its memory consumption and compare the results with existing models.
- to design a privacy-preserving and secure decentralised stochastic gradient descent (SGD) algorithm using blockchain.
- to design a threat model to simulate a denial of service (DOS) attack and linking attacks that are particularly relevant to smart homes and to perform security and privacy analysis to prove the security strength of our proposed scheme and analyse the effectiveness of the prototype to prevent attacks.
- to observe the results of the performed attacks and analyse how well the proposed scheme withstands the attacks.

## 1.5 Research Methodology and Research Contributions

This research will mainly be based on the theoretical analysis and practical understanding of blockchain technology and smart home systems. Information for the research will be generated through the literature review, expert opinions, software design, experimental simulation, and performance analysis.

To achieve the research objectives discussed in section 1.4, we divide this project into six tasks:

**Task 1: Review the existing literature about blockchain technology and current smart home architecture.**

A broad literature review was conducted on the related topics, and knowledge was acquired on blockchain architecture, application use cases and security along with an understanding of the different platforms to build and implement a blockchain network. Also, information on current smart home architectures is gathered and the main security threats in a smart home environment are identified to be addressed by the research.

**Task 2: Develop a smart home framework based on blockchain architectures.**

Current trends and platforms are further examined in this task. Also, the existing works on blockchain, especially in relation to IoT applications, are reviewed. Details and abstract ideas on the revolution of blockchain and the promising solutions it has brought are studied. Then, Ethereum blockchain is selected to build an initial prototype of a smart home system with possible scenarios.

**Task 3: Implement a private Ethereum network for the proposed scheme.**

IoT devices are simulated and connected to the private network; two smart contracts are written to determine the behaviour of devices in the network through access and monitor transactions.

**Contribution 1**: This work is based on our publication[126], which discusses the implementation of private blockchain using Ethereum smart contracts for smart homes to ensure only the homeowner can access and monitor home appliances.

**Task 4: Present an architecture involving authentication rules and logic based on Ethereum smart contracts integrated with edge computing.** This work proposes ERC-20 token generation and an attribute based access control mechanism that utilizes Ethereum smart contracts integrated with edge computing (servers) for authenticate user access to IoT smart home devices.

**Task 5: Evaluate the proposed scheme's performance and undertake a security analysis.**

During the task, we discuss the performance evaluation of the proposed scheme and compare it with existing models with respect to various performance metrics. Also, we examine how the scheme accomplishes security goals (confidential, integrity, and availability), and is able to overcome modification and denial of service (DoS) attacks.

**Contribution 2**: This work is based on our publication [127], in which we propose an architecture involving authentication rules and logic based on Ethereum smart contract integrated edge computing.

**Task 6: Apply machine learning on the differential privacy mechanism to send data from a private smart home to the cloud.**

After achieving the security and authentication structure using smart contracts, the next objective is to move towards model privacy. After this, differential privacy machine learning was added on top of blockchain to achieve privacy-preserving and secure a decentralized model.

**Contribution 3**: This work is based on our publication [128]. We develop a privacy preserving and secure decentralized stochastic gradient descent (SGD) algorithm using blockchain. We apply the algorithm on our own dataset which was generated from a pcap file of the private network traffic of our experiment.

**Task 7: Examine the accuracy and utility of two metrics on our proposed algorithm using different datasets.**

The proposed model's effectiveness is defined in terms of accuracy, utility and privacy leakage. The experiments are performed using three publicly available datasets, UNSW-NB15 dataset, NSL-KDD datasets and ToN-IoT.

**Task 8: Design the threat model and inference attack scenarios and perform privacy analysis.**

Conduct further research on the latest privacy threats and then design a threat model and attack scenarios to ensure the privacy of our scheme.

**Contribution 4**: This work is based on our publication ( Differential Privacy Model for a Smart Home-based Blockchain Architecture) communicated to the Future Generation Computer System (FGCS) journal.

## 1.6 Structure of the thesis

This thesis is organized into three parts and a total of seven chapters. Summary of each chapter is given as follows.

- *Chapter* 2 presents a literature review on the topics, background and related works relevant to the research area. The chapter reviews smart home systems and blockchain technology. Also, this chapter discusses the integration of blockchain and cloud computing, the security and privacy mechanism in blockchain and finally, the threats and attacks related to the research are explained.

- *Chapter* 3 describes the initial implementation of Ethereum smart contracts. The use of blockchain infrastructure is proposed to secure smart home transactions. Using a private Ethereum blockchain, smart home IoT devices are configured. Smart contracts are built to specify the behaviours of IoT devices on the network. As a proof of concept, raspberry Pis was used to simulate IoT devices in one smart home scenario.

- *Chapter* 4 presents a novel lightweight Ethereum blockchain-based multi-tier edge-smart home architecture. The multi-edge servers work as local blockchain miners. Rules and policies are enforced using smart contracts in an automated manner to regulate smart home IoT devices based on the Attribute-Based Access Control (ABAC). This chapter describes ERC-20 token generation and the attribute-based access control mechanism that utilizes Ethereum smart contracts integrated with edge computing (servers) to authenticate user access to IoT smart home devices. Smart contracts issue access tokens directly, with no need for an intermediary or trusted third party.

- *Chapter* 5 focuses on the privacy aspect of our smart home-based blockchain architecture. This chapter presents the use of differential privacy machine learning as a privacy-preserving mechanism. In this chapter, the dataset was produced by generating a pcap file using Wireshark to capture the network packets in our private network.

- *Chapter* 6 focuses on assessing the differential privacy algorithm's performance using three publicly available IoT datasets. We aim to achieve a better privacy guarantee to protect smart home data with better accuracy and utility.

- *Chapter* 7 concludes the thesis and summarizes the research outcomes along with future works which may be conducted to further enhance the security and privacy architecture developed in this research.

# Part I

# Automated Ethereum Smart Contract for Blockchain-based Smart Home Security

# CHAPTER 2

# LITERATURE REVIEW

This chapter presents a survey of the literature on the main topics of the thesis. Section 2.1 overviews smart homes and Section 2.2 overviews blockchain technology. In Section 2.3, cloud infrastructure and support for smart homes is discussed and in Section 2.4, security mechanisms and privacy-preserving techniques based on blockchain are detailed in Section 2.5. Finally, threats and attacks are discussed in Section 2.6.

## 2.1 Smart Home

### 2.1.1 What is a Smart Home?

The inception of information and communication technology is associated with the advent of the Internet. Since then, Internet services have been evolving at an astonishing rate, to the point where technology specialists have had to create a novel terminology that explains how one can fully utilize the revolutionary characteristics of these services. This terminology is the Internet-of-Things, and it includes an array of applications, such as smart homes, that depend on the Internet [61].

Consequently, the idea of enriching the existing literature with novel research approaches proposed over the last ten years regarding various patterns of data used in smart home systems is mainly attributed to the exponential growth in the data itself, and this alone has proven to be more than enough to cause a revolutionary transformation in relation to how smart homes are designed and further developed [172].

To understand how smart homes function, one should first comprehend the essence of the unique integrative and communicative network that a smart home creates, within which several devices and appliances can be used, controlled, and regulated remotely, not to mention the ability of each device to send and receive messages from other devices in such a distinctive manner as to assists the residents of these homes to build a personalized environment that fits their lifestyles [17].

In this vein, researchers define a smart home as "one in which a communication network links sensors, appliances, controls, and other devices that allow for remote monitoring and control by occupants and others to provide frequent and regular services to occupants and the electricity system" [67](p.96). This serves as a useful paradigm for us to build upon.

Therefore, as seen in Fig. 2.1, the key technical attribute of smart home systems seems to revolve around creating a central control hub that is mainly responsible for playing the role of mediator element between several other appliances and household items and the owner of the smart home who operates these appliances and tools. This central control hub is called the Home Gateway, and it uses a number of communication protocols that, in turn, link the external network to the home network [95].

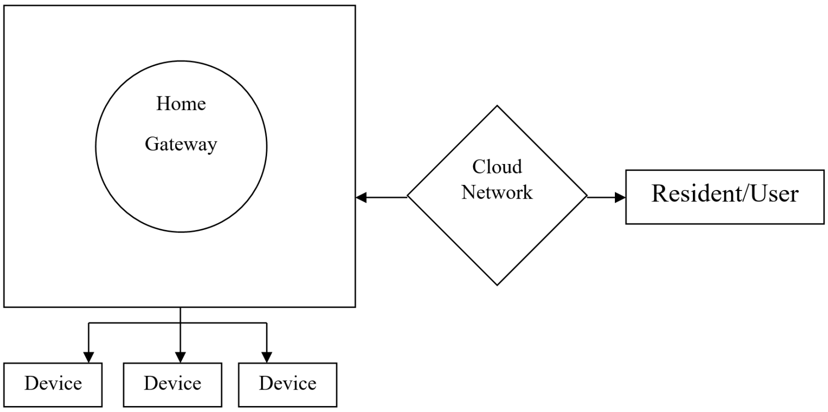


Figure 2.1: Depiction of the traditional structure of a smart home system

Some authors state that smart homes require a well programmed safety system that prevents others from breaking into the house or hacking into its devices and using them for undesirable purposes [5]. Security is important, but that does not mean that it is automatically factored in. It is also important to keep in mind that these systems also need to be user-friendly, with an interface that allows almost anybody to effectively and swiftly utilize their functions.

According to experts, smart homes require a number of elements to be present, including [95]:

- a master system that involves communication networks, servers, and workstations.
- an easy-to-use interactive terminal for family members that allows them to control and operate all smart household equipment and appliances.
- smart sockets that can be installed in a manner that connects appliances to power outlets.
- smart appliances, including television units, refrigerators, air conditioners, cookers, and washing machines that are programmed to send/receive data to/from the smart home master system.
- smart home security systems that prevent hackers and intruders from breaching the privacy of the household or using its items without permission. These systems include smart cameras, smoke and gas-leak sensors, and emergency buttons.

### 2.1.2 Centralized Smart Home Architecture

Generally speaking, traditional smart home systems are controlled by a centralized architecture. This form of architecture places all kinds of appliances and household equipment into one network that is controlled via a home gateway, which in turn can be used by the smart homeowner to reach these appliances and equipment from one central hub of control [155].

This home gateway system serves as the only element allowing every single device within the home network to establish some sort of communication with the external network. Smart home gateways are also meant to carry out specific functions such as gathering information regarding energy consumption, processing the smart homeowner's data, and also accurately monitoring his/her location [165].

Thus, the main idea of using centralized architecture revolves around having one central processing gateway that is responsible for receiving, managing, analysing, and processing data that will later be used by the end-users to take action [120]. Thus, not only remote access but centralization of that access is important for understanding the possible ways in which the benefits of smart technology may be reaped.

It is suggested in [7] that centralized architectures require a Central controller whose main function is to gather information related to appliances and household equipment to further process this information. This controller is considered the most vital element within the processing procedure, as it controls all appliances and household items, as well as smart grid data with regard to the amount of energy that these devices and items are meant to consume [7]. The use of this data is key to the benefits of certain types of smart home technology.

Therefore, centralized smart home architecture is a type of programmed control scheme that can be installed in a computer system. It uses a number of outlets and sources, from which it gathers various patterns of information. These outlets and sources encompass client interfaces, actuators, sensors, and control calculations, which makes centralized smart home architecture clearly different from the type of architecture known as distributed smart home architecture [105].

### 2.1.3 Security and Privacy Issues Related to Centralized Architecture

Centralized architecture is regarded as the main system through which IoT applications operate. However, it has faced a great deal of criticism as far as security and privacy issues are concerned, due to its rigid and vulnerable programming that can easily be infiltrated by hackers and malware, which endangers all information stored within the system and causes the users to fall prey to exploitation [39]. We intend to consider whether these criticisms are justified.

Researchers indicate that hackers normally find systems programmed using a centralized architecture to be the perfect environment for practising their nefarious activities against smart home users [94]. The intrusion occurs through tampering with the home gateway's data first, then it becomes easier for the attacker to start using household items and devices, or even steal critical information about the smart home users themselves.

The main problem with centralized architecture as regards privacy and security issues lies in the fact that it uses one control hub that directs every single command, which could allow the whole smart home system to collapse at any point if just one small malfunction occurs. It also makes the task of hacking easier for attackers, as all they need to do is to hack into the home gateway to gain access to the entire system [129].

One way of hacking into a centralized smart home architecture system is to simply use a decoy disguised as a member device within the smart home network, which can easily allow the attacker to contaminate the home gateway network and obtain the data it contains, which exposes the system to further privacy and security risks [88].

### 2.1.4 Decentralized Smart Home Architecture

Decentralized smart home architecture is constructed differently to its centralized counterparts. As one author puts it, decentralized architecture systems are becoming a better alternative that offers much more effective protection schemes than other centralized smart home architecture systems [37]. Decentralized architecture is also considered to to be beneficial in allowing owners to enjoy data sovereignty, as well as preserving the confidentiality of data.

Some researchers indicate that decentralized architecture could also be referred to as distributed architecture which is evident through an actualized control system and addressed and used as a processing framework that is disseminated, which further embeds all software components into one home automation network [105].

### 2.1.5 Security and Privacy Issues Related to Decentralized Architecture

Decentralized architecture-based systems are seen as an effective solution that could be used as an alternative to centralized systems, due to their strong ability to ensure privacy and security for system users. This gives decentralized architecture systems an immense advantage as against centralized architecture systems when it comes to creating a more secure environment that maintains users' privacy [144].

Research also clarifies that experts in the field of programming and IoT always tend to use decentralized architecture systems as the 'perfect' architecture that aims to reach a high level of privacy and security for all users in residential environments [4].

Although decentralized smart home architecture is supposed to be more secure and offers more protection and privacy than its centralized architecture counterpart, it still faces a few issues when it is applied in larger networks. At that point, the decentralized architecture system becomes exposed to a number of threatening risks related to security and privacy. These issues can be listed as follows:

- **Iteration Issues**: Decentralized architecture provides network agents with the ability to communicate directly with each other without using a central control hub as a mediator [81]. Although this might prove to be secure, the mandatory iterative attempts to communicate may pose some threats, leaving the system vulnerable on the security and privacy front.

- **Inconsistency Issues**: Decentralized architecture-based systems are somewhat inconsistent when it comes to their overall level of performance, which may create an opening for attackers within this inconsistent performance timeline, putting users' privacy and security at risk [102].
- **Dissemination Issues**: Decentralized architecture-based systems are always supported by a dissemination scheme that allows multiple users to gain access to the stored data, which creates an environment where unauthorized users can infiltrate the dataset as administrators and steal as much data as they want [153].

It is therefore important to consider the notion that the security benefits of decentralisation are not an open-and-shut case.

## 2.2 Blockchain Technology

### 2.2.1 Blockchain Overview

Blockchain is a useful form of technology that is used to store data in a decentralized way, to prevent most attackers and hackers from committing any sort of criminal act against users, such as blackmail, information exploitation, and using the data for fraudulent purposes. Blockchain is well known for its reliable database infrastructure and it can be utilized in various fields and contexts [157].

Blockchain originated for the first time as a novel technological approach, during the 1980s and the 1990s, and then witnessed a revolutionary shift in 2008 when it was acknowledged and applied in the field of crypto currency [15].

Blockchain normally consists of a number of blocks structured in a consecutive manner [46, 87, 98, 110, 176, 184], as Fig. 2.2 shows. Each one of these blocks is designed and programmed to contain the core data of a given system, and clustered into two sections. The upper section, known as the "Header", includes hashes of current and previous blocks, timestamps, and other relevant information, while the lower section, known as the "Body", contains the main data of the system. The technical terminology can be further demonstrated as follows:

- Main Data: This refers to any form of data that is related to the core service provided by the blockchain application. This data could involve the IoT or banking data records.

- Hash: This refers to the most important factors in any blockchain. A hash is a function that meets the encrypted demands needed to solve for a blockchain computation and developed based on the information present in the block header.
- Timestamp: This refers to how long it takes for a block to be generated, i.e., the timeline of each block from inception to completion.
- Other Information: This refers to any other related information that has something to do with the blockchain application's main data, operated and utilized by the user.

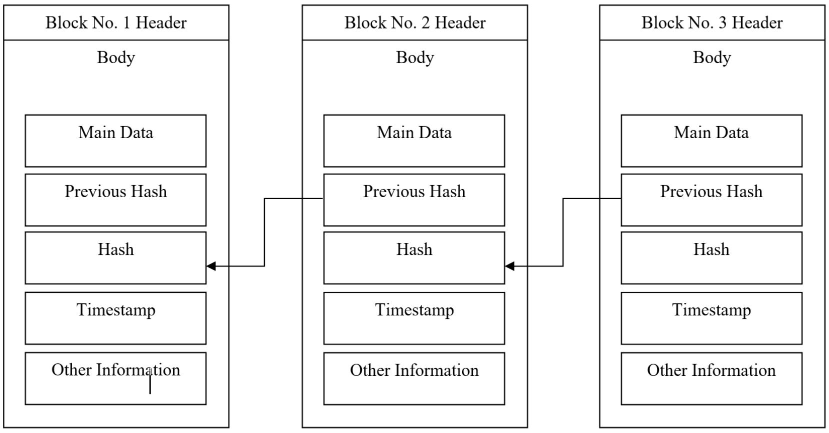


Figure 2.2: Depiction of a traditional blockchain

On the other hand, a blockchain's consensus function indicates the state of compatibility among all nodes related to the blockchain, which means that every node is sequenced in an organised manner with each other node, ensuring that all data transferred from one node to another remains the same, without being altered or hacked in any way, shape, or form [98]. Blockchains are divided into the following three types:

- **Public Blockchains:** This type of blockchains is publicly available for anyone to see, download, compare to other blockchains, and even construct new blockchains that are added to the main one.
- **Private Blockchains:** Unlike the first type, private blockchains are more centralized and require a few authorized users to control and view them. This is because only a small group of participants controls the network. Anyone else who would like to view or edit the blockchain should ask for the main user's permission.

- **Consortium Blockchains:** This distinctive type of blockchain is highly favoured by executive managers in various institutions, thanks to its unique privacy and security features that only allow those who work and communicate with the institution to view and use the blockchain.

Blockchain is thus clearly a technology suited for when privacy is needed, hence our giving it major consideration.

### 2.2.2 Blockchain Technology for Smart Home Security and Privacy

Blockchain technology has been recently chosen by a number of smart homeowners as the most secure way to ensure high levels of privacy and protect their smart household items from being manipulated and abused. Blockchain technology allows smart homeowners to safely send their critical information to other institutions such as banks and online markets without having their privacy breached by any third party service providers [110]. Consequently, blockchain is becoming increasingly popular in smart homes, and a variety of recent research papers have been published on this topic. Table 2.1 summarises the most recent works on blockchain and smart homes. Only three works discuss both the security and privacy of their proposed architectures, demonstrating the value of looking into this matter in the present research.

One work describes a smart district model that combines the IoT with blockchain along with user access to the power grid. This prototype model enables users to work with the power grid via blockchain. Users with solar panel configurations can use the network and its blockchain mechanisms to purchase and/or sell energy. This illustrates how IoT applications using blockchain may be performed and replicated in real-world situations. The authors also draw attention to significant prerequisite factors for smart home systems for consideration in the design and development of new smart home applications [93].

One work describes the design of an Energy-Chain which is a safe energy trading scheme for automated homes that uses blockchain in the smart grid ecosystem. The scheme provides comprehensive security assessments of the frameworks related to communication, costs, and computation times [3].

Some researchers have conducted a case study on a blockchain-based smart home system. They provide details of the central building blocks of the smart home tier as well as exploring the transactions and practices related to the components being

Table 2.1: Summary of the most recent works on blockchain and smart homes

| Reference Number | Main Contribution | security | privacy |
|---|---|---|---|
| [93] | An IoT/blockchain-based smart district approach to improving power grid access for end-users. | No | No |
| [3] | In the smart grid ecosystem, an energy-chain for automated home using blockchain is being developed. | Yes | No |
| [47] | As a sample case study on blockchain, a smart home system is used. | Yes | Yes |
| [73] | Three sensors are used in a blockchain-based Smart Door Lock system to detect motion and distance between nodes. | No | No |
| [108] | IoT systems in smart homes have been created using an efficient lightweight integrated blockchain (ELIB) architecture for IoT. | No | No |
| [94] | Ethereum-based smart home solution that reduces IoT device security, integrity, and authentication concerns, as well as issues with centralised gateways. | Yes | No |
| [46] | A secure and lightweight blockchain-based design for a smart home that is supervised by its owner is proposed. | Yes | Yes |
| [109] | A secure and efficient five-layered blockchain-based IoT system framework that is flexible enough to be adopted by smart houses is proposed. | Yes | No |
| [91] | A novel lightweight blockchain and smart contract-based architecture for smart home hierarchies is presented. | Yes | No |
| [185] | For the proposed architecture, each IoT device maintains the distributed ledger locally, and each smart home uses a local miner to execute transactions on a private or public blockchain. | Yes | No |
| [173] | Smart contracts, which can be formed with Ethereum, are used to store data collected by smart home sensors. | No | No |
| [150] | To ensure secrecy, integrity, scalability, and availability, consortium blockchain is integrated with cloud computing and the smart home architecture. | Yes | Yes |
| [145] | A data privacy-focused smart home solution is developed based on consortium blockchain. | No | Yes |
| [16] | A smart home architecture that includes a private blockchain, a smart home miner (SH miner), local storage connected to smart home sensors (SH sensors), and actuator devices is proposed. | No | No |
| [2] | Blockchain in smart home is explained with the main three tiers: smart home, overlay, and cloud storage. | Yes | No |
| [139] | A new paradigm in which mobile agents, including data migrated between two different household devices, are protected and maintained in a secure blockchain architecture is presented. | No | No |
| [174] | A private blockchain-based access control (PBAC) scheme which involves employing a private blockchain to provide an unforgeable and auditable foundation for smart home systems is proposed. | Yes | No |

described. In addition, the authors have conducted an analysis of the security and privacy outcomes in blockchain-based smart homes. They conclude that the method being proposed demonstrates low-level processing overheads and is suitable for low-resource IoT devices. In turn, the authors assert that their study represents an initial step towards optimising blockchain for connected smart homes [47]. We can therefore be assured that blockchain has already been considered in this context, and is not viewed as an off-the-wall suggestion.

Other proposed a blockchain-based smart door lock system in 2017. The system included plain blockchain processes where three users function as a node to carry out proof of work (PoW). Three sensors are embedded into the system to detect the nodes' motion and distance. Single homeowners (i.e. single node), however, are yet to be discussed as part of the solution. Single nodes raise concerns related to the process by which the blockchain-based door verifies the transactions produced by the single node [73].

Other researchers report their design of an efficient lightweight integrated blockchain (ELIB) model using public blockchain, the cloud, and smart contracts for IoT systems. Their model was applied in smart homes to evaluate its performance and although it reduced processing times and demonstrated satisfactory performance outcomes, the use of the cloud runs the risk of increasing overall system cost [108]. We attempt to offer a better attempt to achieve the aims.

In One paper proposes an Ethereum-based smart home solution to minimise of confidentiality, integrity, and authentication problems with IoT devices. The design also addresses centralised gateway concerns but not the added computational complexities created by blockchain [94].

In addition, a blockchain-based secure and lightweight architecture for smart homes has been proposed. The architecture permits the centralised supervision of the local blockchain by the smart homeowner. All local device and overlay node communication utilises a shared key provided by the miner to support communication security [46]. The authors report that they employed lightweight hashing to uncover transaction anomalies; data confidentiality, integrity, and availability is assured along with safeguards against DDoS attacks. The architecture takes advantage of cloud storage to avoid memory issues with smart home devices. We consider DoS (and DDoS) in due research.

One paper highlights the extensive and difficult to manage security aspects of the implementation of blockchain in IoT settings [109]. The authors propose a five-layered state-of-the-art framework to develop more secure and efficient blockchain-based IoT systems. The framework includes the basic IoT layers in addition to an added storage layer to support enhanced data transmission in a blockchain-based permissioned network. They also use the cloud to store IoT sensor records in response to the limited storage capacity of sensing devices. This enhances the security features related to transactions such as minimal block creation time, integrity, accessibility, availability, scalability, and immutability. Specifically, a blockchain is created in the storage layer when block variations emerge while running consensus algorithms and mining functions. The model's design also has sufficient adaptability to appeal to smart homes, businesses, schools, and smart cities. The question of block creation time was to prove significant in our research.

IoT home devices do not have a great deal of computing power or storage capacity. In addition, they may incur high costs and consume a lot of time when data streaming. A new lightweight blockchain and contract-based hierarchy architecture has been proposed to improve the security levels in smart homes [91]. Specifically, smart contracts are scripts embedded into the private blockchain and are activated by the IoT device when specific conditions are met. This latter point informed our work.

Architecture is presented to support the local storage of the distributed ledger by each IoT device [185]. A local miner is utilised by the smart home for processing transactions in both private and public blockchains. This local miner may also store the data on the device, add other devices to a private blockchain, and insert IoT devices with smart contracts. In response to the limited computing and storage capabilities of IoT devices, the authors established time limits for uploading private blockchain data to local miners. The authors argue that the private blockchain data should be uploaded to local miners every 10 days and that the last five blocks only should be maintained for subsequent transactions.

Researchers report the design and implementation of an Ethereum-based decentralised smart home system [173]. As a software platform, Ethereum emerges from blockchain technology to support developers to assemble and implement decentralised applications. As such, they have used Ethereum to develop smart contracts to store sensor data. The use of Ethereum for smart contracts has enabled the authors to design a system prototype to simulate a smart home application. The model is designed to automatically update humidity and real-time temperature sensors in smart homes when triggered by certain events, demonstrating the benefits underpinning what we seek to do. The authors do not, however, mention that their system is costly to run and some of the design elements require further improvement, a matter that is pertinent to the present research.

Other authors report how they merged consortium blockchain with cloud computing in their system's architecture to improve data confidentiality, integrity, scalability, and accessibility, and thus smart home safety and security [150]. Their system demonstrates how blockchain-based smart home networks can manage transactions utilising green cloud computing. A green broker is utilised to reduce environmental externalities.

Elsewhere, researchers present their consortium blockchain-based smart home system design to address data privacy issues specifically [145]. The model's performance has been positively evaluated using simulation methods; however, no explanation is provided in relation to its energy consumption and processing time, a matter which I do not ignore. Lastly, the impact of Ethereum on a smart home system and developed smart home architecture including private blockchain, a smart home miner, sensors linked to local storage, and actuator devices has been reported [16]. Their architecture is a modified form of the design developed by the authors in [47], but with added Ethereum applications and smart contracts. The system can generate policies for handling transactions which include specifying individuals authorised to access and monitor data.

These authors further claim that Ethereum-based blockchain may be less effective in time-sensitive conditions given it requires around 20 seconds to complete a transaction, which is too long for situations where an urgent response is needed [16]. While we improve on that timescale, there are indeed limits to speed.

Other work reports a thorough study that indicates that blockchain technologies are extremely effective in securing smart homeowners' information, based on what blockchain technology has to offer in relation to a decentralized architecture system, which does not place data in harm's way, by allowing the database to be expanded and preventing past records from being altered [2]. Blockchain technology also has the ability to withstand hazardous and malicious attacks waged by hackers and other online attackers.

A different paper presents a new paradigm in which mobile agents, including data that is automatically and autonomously migrated between two different household devices, are protected and maintained in a secure series of hashes within an architecture fortified by blockchain technology, which protects the smart home system against possible threats and attacks [139]. The paradigm has proven to be successful and easily applied in other patterns of IoT systems.

On the other hand, there is the report of the development of a new scheme known as private blockchain-based secure access control as a perfectly secure system to control household items and devices in smart home systems [174]. The scheme serves as a shield that blocks both internal and external attacks and threats. It functions in a unique yet simple fashion, as the blockchain technology stores access records privately while minimizing the communication and computational overhead. We attempt to achieve similar advantages, not giving external actors write access to the blockchain.

### 2.2.3 Challenges of Integrating Blockchain Technology into Smart Home Systems

Blockchain technology has been used as an essential solution for ensuring security and maintaining data privacy for its users. However, one cannot overlook a few critical issues that bring about the following challenges regarding the use of blockchain technology in IoT applications [43]:

- data mining requires an intensive amount of computing power
- IoT devices include resources that are highly restricted
- mining blocks takes a long time
- poor scalability and hence a poor ability to cope with the increasing number of nodes in the network
- a tendency to create a staggering amount of overhead traffic due to the underlying protocols related to blockchain technology.

Other authors also identified the following challenges facing blockchain technology [2]:

- **Storage:** Data tends to take up a large amount of space due to the increasing number of nodes over time, which leads to an increase in the ledger size.
- **Scalability:** Scaling could eventually change the core characteristic of the blockchain, shifting its approach into a more centralized style of control.
- **Poor Operating Skills:** Unfortunately, only a few people are actually equipped and qualified to skilfully use blockchain technology applications.
- **Time-Consuming:** Although blockchain technology provides a great deal of security and privacy, its encryption process may take a large amount of time.

When it comes to smart home systems, researchers indicate that blockchain technology is not perfect, and could actually result in the following obstacles for users [110]:

- Enormous computational power is required in order to establish a consensus among all nodes within their respective blocks in the integrative network so that certain malicious threats against smart home units can be detected and thwarted.
- The overflow of streaming data from node to node and block to block could easily cause some issues and bring the whole process of communication to a halt, and this data requires a balanced computational scheme that uses high processing power and lightning-fast speed, which is not always the case with every blockchain technology application.
- Within home networks, whenever the number of nodes increases, the scalability of the blockchain noticeably decreases, which affects the effectiveness of transferring data and of keeping it private.
- While blockchain systems do prove to be secure and decentralized, the amount of communication and collaboration between household equipment and devices could lead to data leakage, which makes smart home systems more vulnerable to cyberattacks and threats.

## 2.3 Cloud Infrastructure and Support for Smart Homes

### 2.3.1 Cloud Computing Overview

Authors show that cloud computing offers users processing methodologies that are characterized by both flexibility and convenience, which allows for sharing and outsourcing different amounts of data related to certain contexts [69]. Others have stated that cloud computing includes an essential number of components that encompass the following three factors [11]:

1. Clients: The first component represents the end users' tool that allows them to manage a plethora of information packets held on the cloud. These tools might be any type of computing device such as mobile phones, laptops, or desktops.
2. Distributed Servers: The second component represents the servers which are responsible for the provision of high-quality security and accessibility services, managed from different geographical locations.

3. Data Center: The third component represents the servers through which information can be obtained using the process of virtualization of physical servers to host the service virtually.

A simplification of the concept of cloud computing has been offered, explaining an approach that integrates a number of resources and places them within a virtualized platform online, which allows internet users to gain access to a broadly articulated library of information regarding all fields of research, without being restricted by any spatial or temporal factors or having to use hard drives that cost the user or the operator a huge sum of money and require periodic maintenance, which proves to be a tedious process [104]. While this is not inherent to the nature of blockchain, it is still worth noting, and we will return to it in a moment. Our work builds on ideas of how smart tasks can be performed more efficiently if the computation takes place where more resources are available to do so.

### 2.3.2 Integration of Blockchain Technology with Cloud Computing

The effective gains one can make by integrating blockchain technology with cloud computing are confirmed in the literature, as this level of integration can easily enhance the overall computational performance, obviate of all challenges related to data leakage, increase the processing power of operating blocks, and transfer an exponentially increasing amount of data streaming from node to node [58].

As researchers highlight, once cloud computing is fused with blockchain technology, the capability to protect the system and the data it contains against various threats improves, since data manipulation is much more difficult to process when data is stored in multiple blocks and kept safe at various different locations [113]. According to others, using blockchain technology and cloud computing together is deemed inherently beneficial in numerous professional and academic contexts, including the healthcare sector, the educational sector such as e-learning, and logistics [33]. This is one of the questions that the present research considers for its own context.

Some authors report conducting a study that indicates how safe and effective it is to integrate blockchain technology applications with cloud computing. The study introduces a new achievement in the context of blockchain technology, cloud computing, and IoT called blockchain cloud internet-of-things (BCoT). This new technology provides a secure and consistent sharing and transferring of data on cloud IoT for many applications that function in an array of services such as the healthcare system [117].

In addition to the aforementioned studies, we can read of a study that highlights the significance of using cloud computing as a scaffold for blockchain technology applications, especially in smart home settings, and this is attributed to how cloud computing allows users to reach and use data from the cloud online, which maintains a steady flow of data and enhances blockchain scalability and overall processing power [150]. This corroborates the suitability of our approach.

## 2.4 Security Mechanisms Based on Blockchain

### 2.4.1 The CIA Triad

Based on what has been established, and to gain more insight into the interactions of the CIA triad, all three principles are further and separately elaborated as follows:

**Confidentiality:** The confidentiality principle involves keeping users' information safe, including their professions, identities, and other related information. This occurs through both the establishment of restrictions and the encryption of data related to the critical details of their life and financial transactions [111].

**Integrity:** The integrity principle uses a distinctive security measure that involves a security parameter through which it can detect the level of information accuracy, thus providing authorization based on how accurate this information is. This also helps with the maintenance of data consistency and offers the user, and nobody else, the ability to fully control and regulate their information [72].

**Availability:** The availability principle is regularly seen as a double-edged sword. At its core, it provides the ability to gain access to all kinds of information as the user sees fit. However, this often comes with a high price to pay, as security professionals are required to add more restrictions, strengthen the network, and offer more privacy options for users to properly and safely secure the information that is being accessed [123].

Both organizations and individuals who carry out their professional and daily tasks online share and receive a huge amount of data flowing from different sources which is stored on either cloud drives or traditional drives. With this in mind, researchers have argued that a set of security principles must be promulgated to protect the privacy of data users and maintain an optimally secure online environment [101]. One of the most significantly effective models used to ensure security and privacy for users is the CIA triad, which comprises these principles of confidentiality, integrity, and availability. Should any of these three principles be breached, users' information becomes exposed to a great deal of danger. However, the intensity of this danger might be:

- low and limited, with little to no effect
- medium, with noticeable critical damage
- high, which causes a hazardous impact for users with a severely damaging effect.

Therefore, blockchain technology should seek the full utilization of the CIA triad to integrate its principles of confidentiality, integrity, and availability into its applications, which can, in turn, provide access to a variety of privacy and security measures, such as encryption, transparency, resilience, and auditing [164].

### 2.4.2 Access Control

In the context of IoT systems, access control has proved to be one of the most effective modalities for assuring users' security and privacy. These users include individuals,institutions, and business organizations [22]. Access control boils down to a checkpoint that either refuses or grants users access to a certain body of information. On the other hand, blockchain technology can easily use access control to help establish a system that is more decentralized and offers an architecture that is well able to overcome single-point failures [135]. This flexibility is relevant to our approach.

Access control encompasses three main types known as trust-based access control (TBAC for short), role-based access control (RBAC), and credential-based access control or (CBAC) [40]. These are further elaborated as follows:

**Trust-Based Access Control (TBAC):** Normally speaking, TBAC methods always use a number of trust parameters as a mechanism for gauging the amount of trust granted by users, knowing how much data is restricted, and what type of information can be either revealed or locked away [133].

**Role-Based Access Control (RBAC):** is an access control framework that offers a mechanism that allows organizations to communicate different strands of data based on each user's role within each organization, and though it is not used in computer networks, RBAC is still used in direct communication networks [34].

**Credential-Based Access Control (CBAC):**

is an access control method that forces users who wish to gain access to certain datasets to possess some sort of authority of credentials first, as a way of enhancing privacy and security measures. Credential-based access control is further divided into two types, attribute-based access control (ABAC) and capability-based access control (CBAC) [162]. These are further explained as follows:

**Attribute-Based Access Control (ABAC):** researchers have concluded that ABAC is an effective solution that helps mitigate the problems caused by other traditional access control frameworks, due to its unique architecture that allows users to gain access to datasets stored within the system based on their attributes, and not their roles or security labels given to them by system administrators [143]. This is clearly beneficial in creating an autonomous system that does not require any form of manual intervention.

**Capability-Based Access Control (CBAC):** is another access control framework, which allows users to gain access to different datasets depending on the signature of key figures and factors in charge of operating and influencing the system, such as the copyright owner, service providers, and access periods. This could easily be effected by generating and then verifying tokens to help users gain access to data [99].

## 2.5 Privacy-Preserving Techniques Based on Blockchain

### 2.5.1 Privacy-Preservation Techniques

Blockchain technology applications face several challenges regarding the preservation of users' privacy due to the possible data leakage which may occur when the number of nodes in a block progressively increases. Therefore, some of the following privacy-preserving techniques can be implemented into blockchain technology to mitigate the serious impact of these incidents. Researchers have listed the following main types of privacy-preservation techniques [75, 122]:

- **Encryption**:This strategy is common in blockchain networks to secure transactions and data transmissions. Each user in the blockchain network is allocated two keys: a public key to use with other blockchain users and transmit messages to a specific node, and a private key to decrypt read only messages. The encryption/decryption approach protects messages and maintains the privacy of blockchain transactions. Encryption-based privacy-preserving features do, however, increase the computation and communication demands on the IoT network. For instance, nodes supported by encryption and decryption have high computation costs to produce and deliver keys, which significantly increases computation demands. Furthermore, encryption strategies can have loopholes in their mathematical formulas, resulting in a compromised capacity to deliver total data privacy [146].

- **Anonymization**: This strategy to preserve IoT system privacy has been applied by researchers in blockchain-based IoT applications such as electronic health records, financial platforms, vehicle networks, and energy systems. Researchers have proposed an increase in anonymization strategies including k-anonymity, t-closeness, and l-diversity [75]. Although anonymization delivers robust privacy guarantees to most blockchain-based IoT systems, they are prone to compromise, such as linking attacks, where data from external sources is combined with protected anonymized data to access IoT users' private data [160]. Additionally, anonymization may limit the extent to which details of records can be accessed, leaving the analyst/receiver unable to access potentially necessary details from the anonymized dataset.

- **Mixing**: Coin-mixing protocols support user anonymity when engaging in financial transactions using blockchain-based IoT systems. Traditional mixing methods are not fully decentralized, meaning a trusted third-party server is often needed to transmit transactions. These services generally take transactions from several users and intermix them to protect the transaction identity from adversaries. In 'mixing' transactions, each blockchain user in the IoT system transmits an encrypted new address to a third-party (the mixer) which is then decrypted and shuffled among other addresses before being returned to the transmitter nodes [112]. Current mixing strategies do not, however, utilise a third-party for mixing. Researchers have developed coin-shuffle and mix-coin protocols to protect user privacy during financial IoT blockchain transactions. Mixing approaches function well in financial transactions, but the level of anonymity remains low and may be compromised due to their vulnerability to interception and cyber attacks [32]. Furthermore, full privacy cannot be assured using mixing approaches because a transaction may be traced via the analysis of transactional graphs.

- **Differential Privacy:** Differential privacy is a privacy-preservation technique used to secure participants' information and personal details when this sensitive information is included within a statistical dataset, which in turn allows operators and analysts to review all information needed for their work without revealing the participants' personal details [53]. Differential privacy has solidified its significance as one of the most renowned privacy-preservation techniques that guarantees that users' and participants' personal details will be secured and protected from being infringed upon or stolen. This happens in an environment where specific relevant data could be statistically analysed for research purposes without having to endanger the users' sensitive data [85].

Researchers define differential privacy as "a precise mathematical constraint meant to ensure the privacy of individual pieces of information in a database even while queries are being answered about the aggregate" [35](p.43).

This indicates that differential privacy is nothing but a method for certain analysts who use datasets to only view and edit the data that fits their research, without reaching the participants' personal data, through adding noise to these datasets. This noise is made of randomized fragments of data to create false information that has no meaning or relation to the original dataset, hence protecting the users' vital and personal information.

The reason why differential privacy is the most suitable technique for blockchain technology applications is that it only reveals information that is being used to help specialists and users to carry out certain tasks, without jeopardizing any of their critical information or undermining the privacy of their networks. This allows blockchain technology applications and systems to function in a more effective manner and eliminates the negative impact of any privacy issues that might be related to these applications [76].

### 2.5.2 Privacy-Preserving through Differential Privacy

Blockchain is a ground-breaking technology that has revolutionised digital trading and data storage. The decentralized design of blockchain is regarded as next-generation data storage security. Nonetheless, blockchain still has issues that require solutions before it can be implemented in everyday scenarios. One significant issue requiring attention is data preservation and transaction privacy in blockchain applications. Each user in a decentralized blockchain network is identified by their public key, meaning they cannot retain full privacy or anonymity. Hence, an adversary may present themselves as a third-party to analyse transactions in the network and subsequently work out the users' identities [76]. We attempt to tackle this issue, achieving good results.

The dynamism of differential privacy means it is suitable for use in blockchain scenarios. For instance, point-wise data perturbation differential privacy techniques can introduce noise into the data without disrupting accuracy levels in real-time and broadcast it using blockchain applications [54]. The mechanisms used in point-wise data perturbation first calculate error rates and then calculate the noise value based on the error rate. Noise is then added based on the calculated value to support privacy protection. The noise value recorded is differentially private, thus leaving potential observer adversaries unable to accurately determine the actual value or existence /absence of users within the decentralized database.

Prior to the analysis of statistics from blockchain databases by a third-party, it is possible to provide user protection through the application of differential privacy. Specifically, differential privacy renders aspects of the statistical blockchain data indistinguishable. As a result, the analyst is unable to predict with any confidence the availability of particular blockchain nodes in the dataset. Differential privacy can therefore provide privacy controls for important data, meaning its application in blockchains can provide many positive privacy outcomes. It is thus not surprising that blockchain with differential privacy has been widely investigated across several fields including healthcare [30, 166], crowdsensing [48], cloud computing [60, 176], smart grids [59, 141], and data publishing [9, 42]. However, the application of differential privacy in smart home applications still needs research, an omission which we feel able to tackle.

## 2.6 Threats and Attacks

### 2.6.1 Denial-of-Service Attack (DoS) Attack

A denial-of-service Attack (DoS) attack is one of the most well-known forms of attack within the cyber realm. It simply revolves around the idea of altering the normal characteristics of a specific set of functions carried out by system users, rendering them unavailable and completely malfunctional, so that the user is no longer able to use the services or functions of this system [132]. Fig. 2.3 shows the structure of a regular DoS attack.

Thus, it could be noted that attackers who tend to rely on DoS attacks as their weapon of choice always seem to follow a specific and organized trajectory, which includes sending artificially and maliciously created fake messages to the server, causing the whole system to be brought to a halt, and cutting any form of communication between

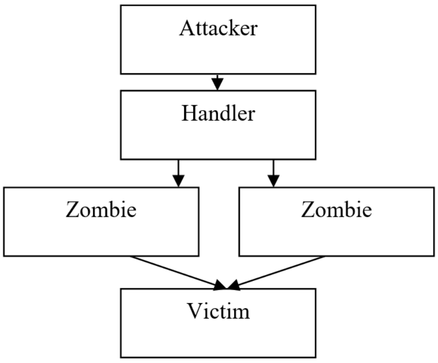


Figure 2.3: Structure of a regular DoS attack

the server and the user [100].

### 2.6.2 Modification Attacks

Modification attacks function through a specific technique where the attacker tends to alter, add, or eliminate data in the victim's dataset. Data can also be mixed, a process called tampering. This attack also includes another commonly used technique that involves the injection of false and incorrect data, in a process known as fuzzing, which allows the attacker to freely falsify information while staying under the radar of detection the whole time. Attackers normally do this by disguising a machine program to roam into the dataset [167]. Authors also highlight that the main objective of any modification attack is to take advantage of any form of communication that takes place between the attacker and the victim, to win the privilege to alter data packets [70]. Our work deals with this issue by means of an appropriate access control.

### 2.6.3 Linking Attacks

Linking attacks are quite unique in that they do not require any programming skills or heavy-duty coding. All the attacker needs to do to perform this attack is to gather fragments of data related to an anonymous user to link them into one body of information, which no longer keeps the user anonymous. It is simply a matter of searching for the data and linking it all together [47].

Accordingly, this relates to data that is normally published online to the public. However, the same data might be scattered all over the internet, or in different datasets.

Therefore, it will take attackers considerable effort to link this data so that they can eventually form a unified picture of sensitive information about users [158].

### 2.6.4 Inference Attacks

The purpose of launching an inference attack is to gather useful information about system users. This information is usually not revealed by the users themselves. However, they might add fragments of their personal details to different systems. For example, an inference attack involves obtaining data on the user's activities relating to his/her daily routine and habits. This data, while it seems trivial, is highly beneficial for the attacker, to figure out other important details such as room temperature data that could be hacked and altered [147]. We consider air conditioning in the smart home context, therefore the recording of temperature data is obviously relevant. The exact condition of the home, and how the occupier wants it to be, enables an attacker to draw inferences about where the occupier is.

Therefore, researchers have indicated that inference attacks are mainly based on the practices and techniques of data mining, so that the attacker can obtain and fully reveal valuable information about the victim, using bits and pieces of other trivial information that might not otherwise prove monumentally critical for the user [161].

## 2.7 Summary

Based on the above review, which has surveyed an extensive body of work conducted by a number of experts whose works' core ideas were extracted by the researcher, we obtain a thorough and cohesive comprehension of fortifying smart home systems with a decentralized blockchain-based architecture that uses certain privacy-preservation techniques to protect residents from possible threats and attacks that seek to reveal and compromise their personal information.

The researcher realizes that even the most effective sets of architecture in the field of home networking might cause the smart home master system to be vulnerable to data leakage, data overflow, or even identity theft. The stolen data might be used and tampered with to blackmail the resident, or simply to render one or more of their smart devices or household items ineffective.

Creating smart home models programmed via centralized architecture systems defeats the purpose, as these architecture systems can be swiftly taken down and manipulated.

Blockchain technology applications, on the other hand, are more effective than other systems, providing more privacy and security options for smart home residents and users, thanks to their decentralized architecture and resilient characteristics, which allow them to have other security and privacy-preservation techniques integrated into their coding scheme to produce a much more powerful system that detects, withstands, and even eliminates possible threats and attacks.

One should be mindful of the fact that blockchain technology applications and solutions might have some technical difficulties and malfunctions while operating smart homes, such as scalability, a poor ability to handle increased number of nodes or amounts of data, and decreased processing power when the amount of data increases. Blockchain technology applications and solutions also happen to be slightly more time-consuming and require specialized skills and knowledge.

These issues can easily be overcome by integrating these applications and solutions with cloud computing technology that utilizes the aforementioned privacy-preservation techniques such as differential privacy, where data can be stored in the cloud over the internet, which decreases data overload and helps the blockchain to process the flowing streams of data at a consistently steady level.

CHAPTER 3

# ETHEREUM-BASED SMART HOME ARCHITECTURE

This chapter is organized as follows: Section 3.2 presents the existing work on blockchain, Section 3.2.1 summarizes traditional smart home architecture, and Section 3.2.2 briefly explains the security functions in blockchain. We develop a prototype implementation of a few smart home components with the help of IoT devices and we demonstrate our scheme in Section 3.3. Prototype evaluation is discussed in Section 3.4. Finally, Section 3.5 summarizes the chapter.

## 3.1 Introduction

In the modern world, Internet of Things (IoT) devices such as sensors and actuators are considered valuable resources for data. This data, which is collected by "things", represents private personal and organizational information and raises privacy, security, and ethical challenges. To overcome the potential issues, well defined and flexible protection mechanisms are required. Many security and privacy approaches have been examined in the IoT environment but tend to be inapplicable and may present limitations because of the nature of decentralized topology and the resource constraints of common devices [45]. Therefore, one proposed solution is to use blockchain-based approaches, which could provide decentralized, secure peer-to-peer networks. Blockchain allows non-trusting members to interact with each other without a trusted intermediate party and without the need for one to give the other write-access. The integration of blockchain- based approaches with IoT devices can produce distributed and trustworthy access control for

IoT [62].

An extensive body of research has been published recently using blockchain as a solution for IoT-based applications, the majority of which only offer proof of concept with possible scenarios. In [118], the authors introduce fair access as a fully decentralized authorization management framework which satisfies the user requirements of controlling and mastering their own privacy. The UTXO model of blockchain was used as a database or policy retrieval point where all access control policies are stored as transactions. Authorization tokens are defined as digital signatures that represent the access rights for specific resource. However, the main limitation of their model is the long-time conformation needed, which is not appropriate for applications requiring high integrity.

To increase the security and privacy of smart home architecture, a lightweight blockchain-based architecture is suggested in [44]. The authors adopt a hybrid approach consisting of three tiers: smart home, overlay network, and cloud storage. IoT devices in the smart home gain the advantages of a private Immutable Ledger (IL), working in a fashion similar to blockchain but managed centrally to reduce the processing overhead. Also, a public blockchain involves higher resource devices joined together to create a distributed trust overlay, which is employed to decrease the process and overhead in the validation of a new block. Different entities communicate in different tiers by transactions which are then grouped into blocks.

In [74], blockchain has been used as database storage. The authors propose a distributed, decentralized publication-subscription-based mechanism that is based on blockchain technology and capabilities, access lists, and access rights control policies. In a user-centric system, different roles can interact and communicate securely in a more private way using scalable messaging services based on a publish-subscribe model and data management protocol stored in blockchain.

In this chapter, the use of blockchain infrastructure is proposed to secure smart home transactions and smart home IoT devices are configured using private Ethereum blockchain. Smart contracts are built to specify the behaviours of IoT devices on the network. As a proof of concept, Raspberry Pis were used to simulate IoT devices in one smart home scenario.

### 3.1.1 Traditional Smart Home Architecture

The concept of a smart home involves the integration of a system and smart devices into the human environment to make people's everyday lives easier. A smart home has an extensive range of solutions such as meters, sensors, and micro-systems that have been built based on a range of technologies, standards, and devices. These solutions can be used to report the required information about the environment on a daily basis. For example, smart devices in a smart home can provide information on the temperature level or energy consumption [21].

Traditional smart homes, as shown in Fig. 3.1, are based on a centralized architecture where home devices are connected to an intermediate hub which provides direct internet connectivity. The communication between these devices and the hub is wireless, using different protocols such as Zigbee or Z-wave. Then, the hub is connected to the home's router to connect the devices to the outside world [62].

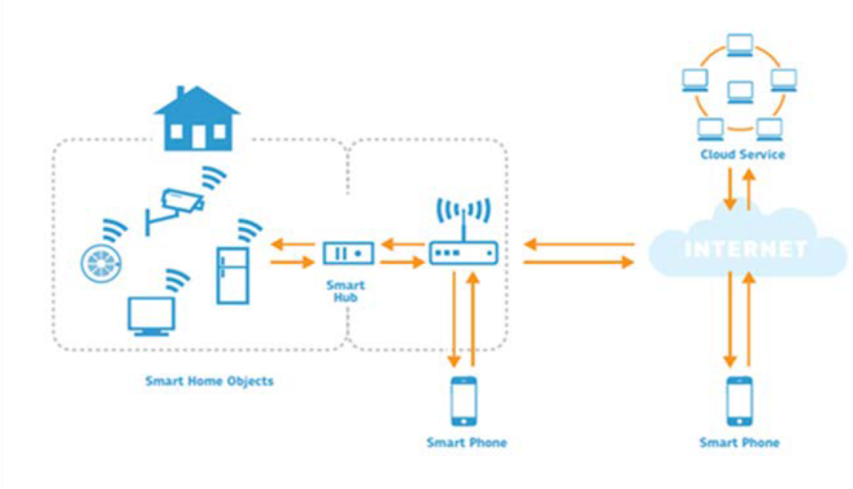


Figure 3.1: Architecture of a traditional smart home [62]

Integration between all the devices results in increased security and privacy issues in the smart home environment [19]. Research and previous work have been performed to identify and understand the potential threats and existing techniques that have been adapted for the smart home environment. For example, [151] proposed a network-centric approach which monitors network activities to detect suspicious behaviour and the use of software defined networking (SDN) technology in the context of the smart house to dynamically block devices based on their network activities. The work in [12] describes a practical traffic-shaping method that effectively protect smart home privacy from a passive network adversary without significantly increasing data cost or reducing network performance. The work in [169] applies a new lightweight encryption/decryption ID authentication method among sensor nodes using a dynamic variable cipher security certificate. However, traditional security approaches are mostly centralized and expensive. Energy consumption and processing overheads are high and there is also a difficulty of scale. Therefore, smart home devices demand a scalable and decentralized approach to overcome this challenge [45].

### 3.1.2 Security and Blockchain

Any security design should address the CIA triad: confidentiality, integrity, and availability functions associated with data and systems. Confidentiality prevents unauthorized users from accessing private data while ensuring it is received only by the correct users. Integrity maintains the consistency and accuracy of the data by making sure the transmitted data is received unaltered. Availability guarantees access to the data when users need it [45]. In blockchain, confidentiality can be addressed by the use of a pair of private and public keys which every node has to own. The sender node uses the private key to sign a digital signature and broadcast the transaction throughout the whole network. The receiver node validates the transaction using the sender node public key. In this way, only valid transactions are stored and added to the blockchain [16]. Although it has been argued that confidentiality and privacy in blockchain are hard to achieve due to the visibility of all the transaction content to every node on the network, many methods have been proposed to tackle this issue [31]. Zero-knowledge proofs and homomorphic encryption are two different methods which have been discussed in the literature [83, 90].

In addition, to ensure data integrity, several cryptographic tools and appropriate data replication strategies are used [57]. In a blockchain-based architecture, the full replications of blockchains exist on a large number of nodes, where all nodes have the same copy of the blocks. Moreover, many cryptographic techniques are used in blockchain including the hash function, digital signatures, and the Merkle tree. A series of SHA-256 hashing functions conduct the mining process to write new transactions, timestamp them, and add them to a block. When a block becomes part of the chain, all miners have to validate and agree on its contents. Hence, it is practically impossible to reverse a transaction because of the one-way nature of the hashing function and the huge computing power that is needed to tamper with the blockchain. The elliptic curve digital signature algorithm (ECDSA) is used in blockchain to generate a digital signature to ensure that all transactions are conducted only by the rightful node. Also, blockchain uses a Merkle tree structure which allows the secure verification of the contents of large data by sending only the hash of the data: the receiver node checks the hash against the root of the Merkle tree. Any change in any transaction at the bottom will result in a change in the hash of the node above and so on up to the root of the tree, which means the hash of the block will be different and will become an invalid block [159].

With regard to availability, blockchain is a fully decentralized architecture which ensures that there is no single point of failure and data is distributed over multiple nodes. Each node in the network has a copy of all transaction history which can be verified and traced back to the first transaction. This results in distributed and fault tolerant architecture [159, 175]. It is assumed in [4] that the blockchain infrastructure is much more resilient to availability threats such as impersonations or DoS than other IoT centralized architecture.

Therefore, the cryptographic hashing in blockchain and its consensus protocol, which verifies whether or not the hash matches its block, make blockchain theoretically tamper-proof. The hash requires a great deal of computing time and energy to generate and serves as proof of work to ensure that each node undertakes computational work to add a new block to the chain without altering the content of the block. Also, hashes link each block with the previous block's unique hash. So, any change in one block will require the calculation of a new hash for that block and also for every subsequent block or else the block will conflict with existing blocks and other nodes will reject the alteration. This is what makes blockchain immutable.

## 3.2 Proposed work

Compared to other blockchain technologies, the Ethereum proposed by Vatalik Buterin in 2013, is a publicly distributed blockchain technology executed by Ethereum Virtual Machine (EVM) [80] which allows users to create their own programs with the desired complexity using a smart contract. This characteristic allows Ethereum to be used by different decentralized applications, not limited to cryptocurrencies. It is suited for applications that require automatic interaction between peers on the network [25, 171].

At its simplest, the transaction time for Ethereum is 12 seconds compared to Bitcoin which has a block time of 10 minutes, given that it is used for a wide variety of applications. Recently, many organizations and industries have tried to build their own use cases for Ethereum [80].

### 3.2.1 Ethereum-based Smart Home Architecture

In the smart home scenario, smart devices may communicate with each other directly to request data to offer certain services. For example, a smart air conditioner (AC) requests the present room temperature from the temperature sensor in order to turn on the AC automatically when the temperature goes up to a certain value or turn the heater on if the temperature falls below a certain value. Both devices can also send alerts or notifications to the user about their state.

We may begin by considering current smart solutions and looking at how they function via conventional methods as shown in Figure Fig. 3.2 If one wishes to remotely control a smart air conditioner, then one would usually require a (hopefully) secure web service which allows access only after one enters the login and password. It is then possible to send a command, and the web service will instruct the hardware to activate or deactivate the AC.

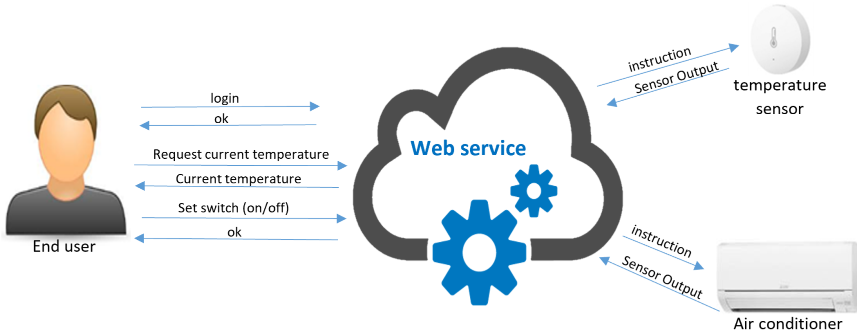


Figure 3.2: Traditional way to control a connected smart device

The issue here is that the web services provide what is termed "write access". The instructions sent are translated into hardware instructions by the web service and then executed by the attached device, in this case the AC. By definition, it is not secure to allow write access. No web service is impenetrable. Granting access to a device from "the outside" always raises the possibility that someone will be able to hack this access for ulterior ends. This vulnerability is a consequence of a single point of failure, the point at which the authentication is verified and where incoming instructions are accepted.

Therefore, we propose a prototype built on blockchain and using the smart contract to controls the permissions for changing the (AC) state.

The architecture of our proposed prototype is based on Ethereum smart contract and consists of a smart home miner connected to private blockchain, temperature sensor, and air conditioner (AC). We use Raspberry Pi to simulate IoT devices. . Fig. 3.3 shows the experimental prototype for smart home IoT devices and shows the interaction between the devices through monitor transactions and access transactions.

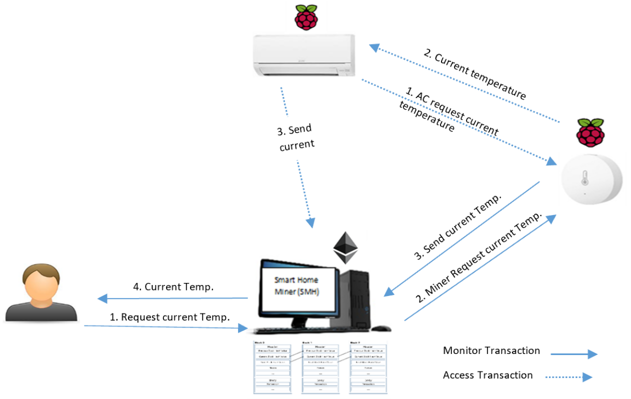


Figure 3.3: Experimental prototype

### 3.2.2 Smart Contract Creation process

For our experiment, we create two smart contracts. The first is the monitor contract to check the temperature sensor reading deployed in the first Raspberry Pi. The second smart contract is deployed in the second Raspberry Pi's access contract, which allows the AC to request a temperature reading value from the monitor contract.

**Monitor Contract.** This contract allows the homeowner to check the current value of the temperature. Only the owner can set and change the temperature value, by specifying the address of the owner who has the permissions to set the value in the contract body. The contract can send alerts to the owner at certain times to show the current room temperature.

**Access Contract.** This contract can request temperature reading values by calling the value from the monitor contract. Then, based on the value, the contract will either turn the A/C on or turn it off, and send notifications to the owner about its current state.

#### 3.2.2.1 Hardware and Software

We build a case using one laptop (Dell XPS) and two single-board computer (Raspberry Pi 3 model B). On each device, we install a geth client (command line interface implementing in Go-Ethereum) transfer devices to Ethereum nodes [170]. For each node, we create an Ethereum account and configure these nodes to form a private blockchain network, where the laptop plays the role of two miners because it's large computing and storage capability. The Raspberry Pi functions as a lightweight Ethereum node to deploy the monitor contract and access contract.

For the writing and compiling of our contracts we used the Remix integrated development (IDE) [134].This is a browser- based IDE for Solidity, which is the language used to write smart contracts. For deploying and compiling the contract and also monitoring the contract state, Web3.js (Ethereum JavaScript API) is adapted to interact with the corresponding geth client through an HTTP connection [125]. A simple HTML web page is built to facilitate the interaction between the homeowner and the devices.

#### 3.2.2.2 Implementation

Based on the guidelines discussed in the Ethereum white paper [28], we next configured our private blockchain with some modifications:

1. A compatible version of the Ethereum client for each device is chosen for download and installation.

2. Windows power shell is used to start geth by executing "geth" command..

3. In our private blockchain, each node has to fulfil the requirements to be able to join the same blockchain; these requirements include:

    a) the same genesis file (Test.json) has to be initialized by every node. The initialization creates the genesis block, which is the first block of the blockchain and does not refer to any block.

    b) the same network ID has to be used by each node to connect to the same blockchain. Any ID can be assigned except 1, 2, and 3 since they are reserved for the main chain. For our configuration we assign network id 4224 as follow:

    ```
    {
      "config": {
        "chainId": 4224,
    ```

```
    "homesteadBlock": 1,
    "eip150Block": 2,
    "eip150Hash": "0x0000000000000000000000000000000000000",
    "eip155Block": 3,
    "eip158Block": 3,
    "byzantiumBlock": 4,
    "ethash": {}
  },
  "nonce": "0x0",
  "timestamp": "0x5b41b451",
  "extraData": "0x00000000000000000000000000000000000000000",
  "gasLimit": "0x47b760",
  "difficulty": "0x80000",
  "mixHash": "0x0000000000000000000000000000000000000000000",
  "coinbase": "0x0000000000000000000000000000000000000000",
  "alloc": {
    "0000000000000000000000000000000000000000": {
      "balance": "0x1"
    },
    "0000000000000000000000000000000000000001": {
      "balance": "0x1"
        }
  },
  "number": "0x0",
  "gasUsed": "0x0",
  "parentHash": "0x00000000000000000000000000000000000000000"
}
```

4. to initialize our private blockchain, the following geth command is executed

   ```
   geth \   - datadir / user/Amjad/Test/miner1 init Test.json
   ```

5. Then we create an account for each node where every account has a private and public key and is indexed by its address, which is derived from the last 20 bytes of the public key.

   ```
   geth  -datadir / user/Amjad/Test/miner1 account new
   ```

6. To start geth on each node, the following command is executed, which includes different flags for different functionalities. For more information on each flag, refer to the Ethereum white paper [28].

   ```
   geth --networkid 4224 --mine --minerthreads 2 --datadir "." --nodiscover
   --rpc --rpcport "8543" --port "30304" --ipcdisable --rpccorsdomain "*"
   --nat "any"--rpcapi admin,eth,web3,personal,net
   --unlock 0 --password ./password.sec
   ```

7. Due to the limited number of nodes used in our framework, there is no need to use discovery mechanism to pair nodes. Static-nodes.json file is used to pair the nodes. We obtained the enode ID using the following command.

   ```
   >admin.addPeer("enode://5f1d23c79a9bd7505469ed524047d276ad3a5964db76
   3ae4e5c13a53326b9f492e7a02367f8e5c350a960e08bed1604e6860262b9013cf7c
   0c70aad9f91c1094\$@[::]:30303?discport=0")
   ```

8. The last step is repeated to add the two Raspberry Pis as nodes to give a private blockchain with fully synchronized nodes.

#### 3.2.2.3 Smart Contract Development and Deployment

A remix browser is utilized. For the monitoring contract, two main functions are defined: setValue() and getValue(). Only the homeowner can set the temperature value, so a modifier is used to restrict the use of a set function to the address of the homeowner. Any other nodes can request the temperature value by calling the get function, which will return the temperature value.

```
pragma solidity ^0.4.18;
contract Test {
string public Sensor;
address owner;
function Test() public{
owner = msg.sender;}
modifier onlyOwner{
require(msg.sender == owner);
_;
```

```
}
event Value(string sensor);
function setValue(string _Sensor) onlyOwner public {
Sensor = _Sensor;
Value(_Sensor); }
function getValue() public constant returns (string){
return(Sensor); }
}
```

The access contract is developed to allow any node to read the current temperature value. It has only one function, which calls the getValue() function from the monitor contract based on its address. Therefore, it impossible to alter the value because it will only be read from the specific address of the monitor contract.

```
pragma solidity ^0.4.18;
contract Access {
function getSensorValue(address addr) returns (string){
Test T = Test(addr);
return T.getValue(); }
}
contract Test {
function getValue()returns(string);
}
```

Finally, a simple HTML user interface (UI) is built to interact with the smart contract using web3.js. The first UI consists retrieves the temperature value from the getValue() function, and forms one input field for the value which will be set via jQuery from the input text field.

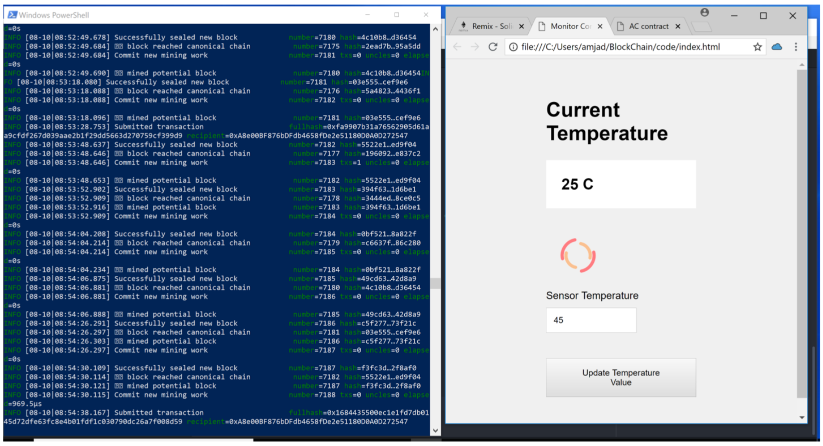
In the head tag, the Web3.js library is imported to connect to our private blockchain nodes. Then, in the script tag the code is written to work with the smart contract. The web provider is set to our localhost 8543. The web3.eth.contract() method is used to create the contract, accepting the Application Binary Interface (ABI) parameter, which allows us to call functions and receive data from our smart contract. The ABI is copied from the Remix browser where our smart contract is written. Then, the actual contract address is defined based on the associated contract address in Remix.

The second UI is built to simulate the AC state. It retrieves the current temperature from the access contract that calls the getValue() from the monitor contract. Every 5 seconds, the temperature reading is updated by calling the new temperature value. Based on the value, a notification will be sent about the current temperature and the AC state (on/off).

## 3.3 Prototype evaluation

### 3.3.1 Snapshot Examples for the UI

1. The monitor contract shows the current value and the minimum to allow the homeowner to set the value. Once the Update Temperature value button is hit, the miner as shown on the left side receives the transaction and commences mining as Fig. 3.4 shows.



```
d=0s
INFO [08-10|08:52:49.678] Successfully sealed new block           number=7180 hash=4c10b8…d36454
INFO [08-10|08:52:49.684] ▯▯ block reached canonical chain          number=7175 hash=2ead7b…95a5dd
INFO [08-10|08:52:49.684] Commit new mining work                  number=7181 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:52:49.690] ▯▯ mined potential block                  number=7180 hash=4c10b8…d36454IN
FO [08-10|08:53:18.080] Successfully sealed new block           number=7181 hash=03e555…cef9e6
INFO [08-10|08:53:18.088] ▯▯ block reached canonical chain          number=7176 hash=5a4823…4436f1
INFO [08-10|08:53:18.088] Commit new mining work                  number=7182 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:53:18.096] ▯▯ mined potential block                  number=7181 hash=03e555…cef9e6
INFO [08-10|08:53:28.753] Submitted transaction                   fullhash=0xfa9907b31a76562905d61a
a9cfdf267d039aae2b1f29dd5663d270759cf399d9 recipient=0xA8e00BF876bDFdb4658fDe2e51180D0A0D272547
INFO [08-10|08:53:48.637] Successfully sealed new block           number=7182 hash=5522e1…ed9f04
INFO [08-10|08:53:48.646] ▯▯ block reached canonical chain          number=7177 hash=196092…e837c2
INFO [08-10|08:53:48.646] Commit new mining work                  number=7183 txs=1 uncles=0 elapse
d=0s
INFO [08-10|08:53:48.653] ▯▯ mined potential block                  number=7182 hash=5522e1…ed9f04
INFO [08-10|08:53:52.902] Successfully sealed new block           number=7183 hash=394f63…1d6be1
INFO [08-10|08:53:52.909] ▯▯ block reached canonical chain          number=7178 hash=3444ed…8ce0c5
INFO [08-10|08:53:52.916] ▯▯ mined potential block                  number=7183 hash=394f63…1d6be1
INFO [08-10|08:53:52.909] Commit new mining work                  number=7184 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:54:04.208] Successfully sealed new block           number=7184 hash=0bf521…8a822f
INFO [08-10|08:54:04.214] ▯▯ block reached canonical chain          number=7179 hash=c6637f…86c280
INFO [08-10|08:54:04.214] Commit new mining work                  number=7185 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:54:04.234] ▯▯ mined potential block                  number=7184 hash=0bf521…8a822f
INFO [08-10|08:54:06.875] Successfully sealed new block           number=7185 hash=49cd63…42d8a9
INFO [08-10|08:54:06.881] ▯▯ block reached canonical chain          number=7180 hash=4c10b8…d36454
INFO [08-10|08:54:06.881] Commit new mining work                  number=7186 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:54:06.888] ▯▯ mined potential block                  number=7185 hash=49cd63…42d8a9
INFO [08-10|08:54:26.291] Successfully sealed new block           number=7186 hash=c5f277…73f21c
INFO [08-10|08:54:26.297] ▯▯ block reached canonical chain          number=7181 hash=03e555…cef9e6
INFO [08-10|08:54:26.303] ▯▯ mined potential block                  number=7186 hash=c5f277…73f21c
INFO [08-10|08:54:26.297] Commit new mining work                  number=7187 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:54:30.109] Successfully sealed new block           number=7187 hash=f3fc3d…2f8af0
INFO [08-10|08:54:30.114] ▯▯ block reached canonical chain          number=7182 hash=5522e1…ed9f04
INFO [08-10|08:54:30.121] ▯▯ mined potential block                  number=7187 hash=f3fc3d…2f8af0
INFO [08-10|08:54:30.115] Commit new mining work                  number=7188 txs=0 uncles=0 elapse
d=969.5µs
INFO [08-10|08:54:38.167] Submitted transaction                   fullhash=0x1684435500ec1e1fd7db01
45d72dfe63fc8e4b01fdf1c030790dc26a7f008d59 recipient=0xA8e00BF876bDFdb4658fDe2e51180D0A0D272547
```

Current Temperature

25 C

Sensor Temperature

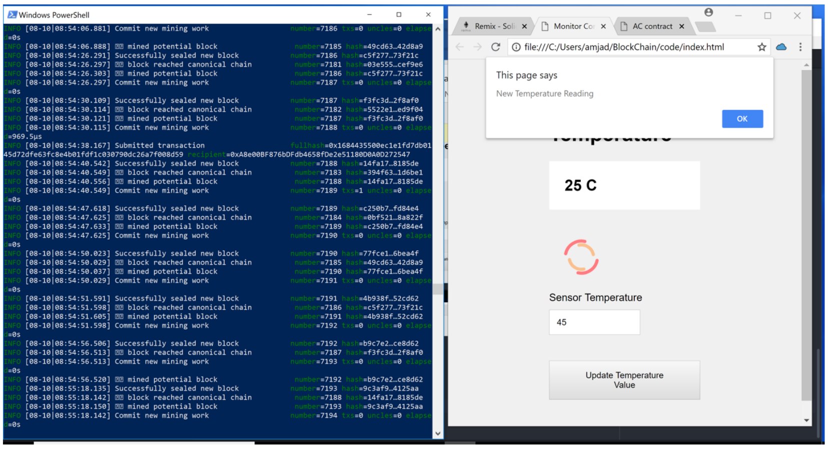
45

Update Temperature Value

Figure 3.4: Owner sets a new value for the temperature

2. Once the transaction is mined, an alert appears as shown in Fig. 3.5 to show the temperature value has been changed and that there is a new temperature reading.

3. Access contract UI, as shown in Fig. 3.6, shows the different present room temperatures and the AC state. It is updated every 5 seconds. Three different notifications are set, based on the temperature reading. If the temperature reading is more than 30°C (Fig. 3.6.a), the AC will turn on. If it is less than 20°C, the AC will be turned on in heater mode, (Fig. 3.6.c). Otherwise, the temperature will be normal and the AC will be set to off (Fig. 3.6.b).



```
Windows PowerShell
INFO [08-10|08:54:06.881] Commit new mining work                   number=7186 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:54:06.888] □□ mined potential block                   number=7185 hash=49cd63…42d8a9
INFO [08-10|08:54:26.291] Successfully sealed new block            number=7186 hash=c5f277…73f21c
INFO [08-10|08:54:26.297] □□ block reached canonical chain           number=7181 hash=03e555…cef9e6
INFO [08-10|08:54:26.303] □□ mined potential block                   number=7186 hash=c5f277…73f21c
INFO [08-10|08:54:26.297] Commit new mining work                   number=7187 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:54:30.109] Successfully sealed new block            number=7187 hash=f3fc3d…2f8af0
INFO [08-10|08:54:30.114] □□ block reached canonical chain           number=7182 hash=5522e1…ed9f04
INFO [08-10|08:54:30.121] □□ mined potential block                   number=7187 hash=f3fc3d…2f8af0
INFO [08-10|08:54:30.115] Commit new mining work                   number=7188 txs=0 uncles=0 elapse
d=969.5µs
INFO [08-10|08:54:38.167] Submitted transaction                    fullhash=0x1684435500ec1e1fd7db01
45d72dfe63fc8e4b01fdf1c030790dc26a7f008d59 recipient=0xA8e00BF876bDFdb4658fDe2e51180D0A0D272547
INFO [08-10|08:54:40.542] Successfully sealed new block            number=7188 hash=14fa17…8185de
INFO [08-10|08:54:40.549] □□ block reached canonical chain           number=7183 hash=394f63…1d6be1
INFO [08-10|08:54:40.556] □□ mined potential block                   number=7188 hash=14fa17…8185de
INFO [08-10|08:54:40.549] Commit new mining work                   number=7189 txs=1 uncles=0 elapse
d=0s
INFO [08-10|08:54:47.618] Successfully sealed new block            number=7189 hash=c250b7…fd84e4
INFO [08-10|08:54:47.625] □□ block reached canonical chain           number=7184 hash=0bf521…8a822f
INFO [08-10|08:54:47.633] □□ mined potential block                   number=7189 hash=c250b7…fd84e4
INFO [08-10|08:54:47.625] Commit new mining work                   number=7190 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:54:50.023] Successfully sealed new block            number=7190 hash=77fce1…6bea4f
INFO [08-10|08:54:50.029] □□ block reached canonical chain           number=7185 hash=49cd63…42d8a9
INFO [08-10|08:54:50.037] □□ mined potential block                   number=7190 hash=77fce1…6bea4f
INFO [08-10|08:54:50.029] Commit new mining work                   number=7191 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:54:51.591] Successfully sealed new block            number=7191 hash=4b938f…52cd62
INFO [08-10|08:54:51.598] □□ block reached canonical chain           number=7186 hash=c5f277…73f21c
INFO [08-10|08:54:51.605] □□ mined potential block                   number=7191 hash=4b938f…52cd62
INFO [08-10|08:54:51.598] Commit new mining work                   number=7192 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:54:56.506] Successfully sealed new block            number=7192 hash=b9c7e2…ce8d62
INFO [08-10|08:54:56.513] □□ block reached canonical chain           number=7187 hash=f3fc3d…2f8af0
INFO [08-10|08:54:56.513] Commit new mining work                   number=7193 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:54:56.520] □□ mined potential block                   number=7192 hash=b9c7e2…ce8d62
INFO [08-10|08:55:18.135] Successfully sealed new block            number=7193 hash=9c3af9…4125aa
INFO [08-10|08:55:18.142] □□ block reached canonical chain           number=7188 hash=14fa17…8185de
INFO [08-10|08:55:18.150] □□ mined potential block                   number=7193 hash=9c3af9…4125aa
INFO [08-10|08:55:18.142] Commit new mining work                   number=7194 txs=0 uncles=0 elapse
d=0s
```

file:///C:/Users/amjad/BlockChain/code/index.html

This page says

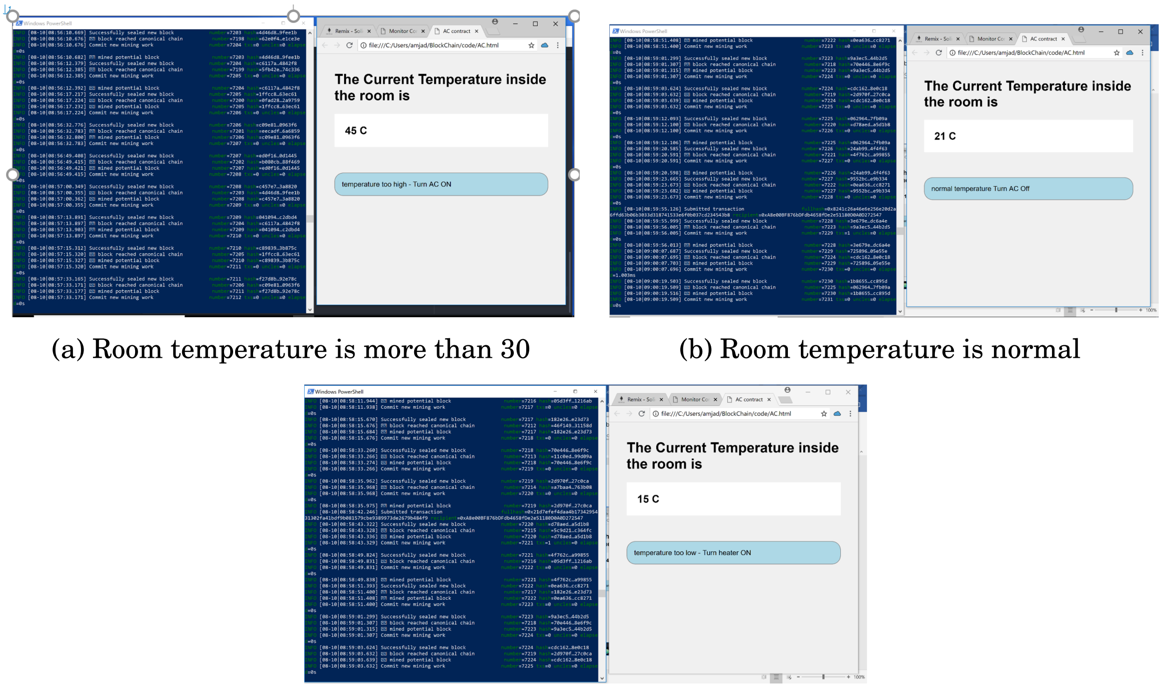
New Temperature Reading

25 C

Sensor Temperature

45

Figure 3.5: New temperature alert



```
Windows PowerShell
INFO [08-10|08:56:10.669] Successfully sealed new block            number=7203 hash=4d46d8…9fee1b
INFO [08-10|08:56:10.676] □□ block reached canonical chain           number=7198 hash=62e0f4…e1ce3e
INFO [08-10|08:56:10.676] Commit new mining work                   number=7204 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:56:10.682] □□ mined potential block                   number=7203 hash=4d46d8…9fee1b
INFO [08-10|08:56:12.379] Successfully sealed new block            number=7204 hash=c6117a…4842f8
INFO [08-10|08:56:12.385] □□ block reached canonical chain           number=7199 hash=5fb42e…74c336
INFO [08-10|08:56:12.385] Commit new mining work                   number=7205 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:56:12.392] □□ mined potential block                   number=7204 hash=c6117a…4842f8
INFO [08-10|08:56:17.217] Successfully sealed new block            number=7205 hash=1ffcc8…63ec61
INFO [08-10|08:56:17.224] □□ block reached canonical chain           number=7200 hash=0fad28…2a9759
INFO [08-10|08:56:17.232] □□ mined potential block                   number=7205 hash=1ffcc8…63ec61
INFO [08-10|08:56:17.224] Commit new mining work                   number=7206 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:56:32.776] Successfully sealed new block            number=7206 hash=c09e81…0963f6
INFO [08-10|08:56:32.783] □□ block reached canonical chain           number=7201 hash=eecadf…6a6859
INFO [08-10|08:56:32.800] □□ mined potential block                   number=7206 hash=c09e81…0963f6
INFO [08-10|08:56:32.783] Commit new mining work                   number=7207 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:56:49.408] Successfully sealed new block            number=7207 hash=ed0f16…0d1445
INFO [08-10|08:56:49.415] □□ block reached canonical chain           number=7202 hash=b080cb…88f469
INFO [08-10|08:56:49.421] □□ mined potential block                   number=7207 hash=ed0f16…0d1445
INFO [08-10|08:56:49.415] Commit new mining work                   number=7208 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:57:00.349] Successfully sealed new block            number=7208 hash=c457e7…3a8820
INFO [08-10|08:57:00.355] □□ block reached canonical chain           number=7203 hash=4d46d8…9fee1b
INFO [08-10|08:57:00.362] □□ mined potential block                   number=7208 hash=c457e7…3a8820
INFO [08-10|08:57:00.355] Commit new mining work                   number=7209 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:57:13.891] Successfully sealed new block            number=7209 hash=041094…c2dbd4
INFO [08-10|08:57:13.897] □□ block reached canonical chain           number=7204 hash=c6117a…4842f8
INFO [08-10|08:57:13.903] □□ mined potential block                   number=7209 hash=041094…c2dbd4
INFO [08-10|08:57:13.897] Commit new mining work                   number=7210 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:57:15.312] Successfully sealed new block            number=7210 hash=c89839…3b875c
INFO [08-10|08:57:15.320] □□ block reached canonical chain           number=7205 hash=1ffcc8…63ec61
INFO [08-10|08:57:15.327] □□ mined potential block                   number=7210 hash=c89839…3b875c
INFO [08-10|08:57:15.320] Commit new mining work                   number=7211 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:57:33.165] Successfully sealed new block            number=7211 hash=f27d8b…92e78c
INFO [08-10|08:57:33.171] □□ block reached canonical chain           number=7206 hash=c09e81…0963f6
INFO [08-10|08:57:33.177] □□ mined potential block                   number=7211 hash=f27d8b…92e78c
INFO [08-10|08:57:33.171] Commit new mining work                   number=7212 txs=0 uncles=0 elapse
d=0s
```

file:///C:/Users/amjad/BlockChain/code/AC.html

The Current Temperature inside the room is

45 C

temperature too high - Turn AC ON

(a) Room temperature is more than 30

```
Windows PowerShell
INFO [08-10|08:58:51.408] □□ mined potential block                   number=7222 hash=0ea636…cc8271
INFO [08-10|08:58:51.400] Commit new mining work                   number=7223 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:59:01.299] Successfully sealed new block            number=7223 hash=9a3ec5…44b2d5
INFO [08-10|08:59:01.307] □□ block reached canonical chain           number=7218 hash=70e446…8e6f9c
INFO [08-10|08:59:01.315] □□ mined potential block                   number=7223 hash=9a3ec5…44b2d5
INFO [08-10|08:59:01.307] Commit new mining work                   number=7224 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:59:03.624] Successfully sealed new block            number=7224 hash=cdc162…8e0c18
INFO [08-10|08:59:03.632] □□ block reached canonical chain           number=7219 hash=2d970f…27c0ca
INFO [08-10|08:59:03.639] □□ mined potential block                   number=7224 hash=cdc162…8e0c18
INFO [08-10|08:59:03.632] Commit new mining work                   number=7225 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:59:12.093] Successfully sealed new block            number=7225 hash=062964…7fb09a
INFO [08-10|08:59:12.100] □□ block reached canonical chain           number=7220 hash=d78aed…a5d1b8
INFO [08-10|08:59:12.100] Commit new mining work                   number=7226 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:59:12.106] □□ mined potential block                   number=7225 hash=062964…7fb09a
INFO [08-10|08:59:20.585] Successfully sealed new block            number=7226 hash=24ab99…4f4f63
INFO [08-10|08:59:20.591] □□ block reached canonical chain           number=7221 hash=4f762c…a99855
INFO [08-10|08:59:20.591] Commit new mining work                   number=7227 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:59:20.598] □□ mined potential block                   number=7226 hash=24ab99…4f4f63
INFO [08-10|08:59:23.665] Successfully sealed new block            number=7227 hash=9552bc…e9b334
INFO [08-10|08:59:23.673] □□ block reached canonical chain           number=7222 hash=0ea636…cc8271
INFO [08-10|08:59:23.682] □□ mined potential block                   number=7227 hash=9552bc…e9b334
INFO [08-10|08:59:23.673] Commit new mining work                   number=7228 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:59:55.126] Submitted transaction                    fullhash=0x8241c26a46e6e256e20d2a
6ffd63b06b3033d318741533e6f0b037cd234543b8 recipient=0xA8e00BF876bDFdb4658fDe2e51180D0A0D272547
INFO [08-10|08:59:55.999] Successfully sealed new block            number=7228 hash=3e679e…dc6a4e
INFO [08-10|08:59:56.005] □□ block reached canonical chain           number=7223 hash=9a3ec5…44b2d5
INFO [08-10|08:59:56.005] Commit new mining work                   number=7229 txs=1 uncles=0 elapse
d=0s
INFO [08-10|08:59:56.013] □□ mined potential block                   number=7228 hash=3e679e…dc6a4e
INFO [08-10|09:00:07.687] Successfully sealed new block            number=7229 hash=725896…05e55e
INFO [08-10|09:00:07.695] □□ block reached canonical chain           number=7224 hash=cdc162…8e0c18
INFO [08-10|09:00:07.703] □□ mined potential block                   number=7229 hash=725896…05e55e
INFO [08-10|09:00:07.696] Commit new mining work                   number=7230 txs=0 uncles=0 elapse
d=1.003ms
INFO [08-10|09:00:19.503] Successfully sealed new block            number=7230 hash=1b8655…cc895d
INFO [08-10|09:00:19.509] □□ block reached canonical chain           number=7225 hash=062964…7fb09a
INFO [08-10|09:00:19.516] □□ mined potential block                   number=7230 hash=1b8655…cc895d
INFO [08-10|09:00:19.509] Commit new mining work                   number=7231 txs=0 uncles=0 elapse
d=0s
```

file:///C:/Users/amjad/BlockChain/code/AC.html

The Current Temperature inside the room is

21 C

normal temperature Turn AC Off

(b) Room temperature is normal

```
Windows PowerShell
INFO [08-10|08:58:11.944] □□ mined potential block                   number=7216 hash=05d3ff…1216ab
INFO [08-10|08:58:11.938] Commit new mining work                   number=7217 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:58:15.670] Successfully sealed new block            number=7217 hash=182e26…e23d73
INFO [08-10|08:58:15.676] □□ block reached canonical chain           number=7212 hash=46f149…31158d
INFO [08-10|08:58:15.684] □□ mined potential block                   number=7217 hash=182e26…e23d73
INFO [08-10|08:58:15.676] Commit new mining work                   number=7218 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:58:33.260] Successfully sealed new block            number=7218 hash=70e446…8e6f9c
INFO [08-10|08:58:33.266] □□ block reached canonical chain           number=7213 hash=11c0ed…99d09a
INFO [08-10|08:58:33.274] □□ mined potential block                   number=7218 hash=70e446…8e6f9c
INFO [08-10|08:58:33.266] Commit new mining work                   number=7219 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:58:35.962] Successfully sealed new block            number=7219 hash=2d970f…27c0ca
INFO [08-10|08:58:35.968] □□ block reached canonical chain           number=7214 hash=a7baa4…763b08
INFO [08-10|08:58:35.968] Commit new mining work                   number=7220 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:58:35.975] □□ mined potential block                   number=7219 hash=2d970f…27c0ca
INFO [08-10|08:58:42.246] Submitted transaction                    fullhash=0x21d7efef4daa4b17342954
31302fa41bdf9b081579cbe9389973de2679b484f9 recipient=0xA8e00BF876bDFdb4658fDe2e51180D0A0D272547
INFO [08-10|08:58:43.322] Successfully sealed new block            number=7220 hash=d78aed…a5d1b8
INFO [08-10|08:58:43.328] □□ block reached canonical chain           number=7215 hash=5c9d21…c364fc
INFO [08-10|08:58:43.336] □□ mined potential block                   number=7220 hash=d78aed…a5d1b8
INFO [08-10|08:58:43.329] Commit new mining work                   number=7221 txs=1 uncles=0 elapse
d=0s
INFO [08-10|08:58:49.824] Successfully sealed new block            number=7221 hash=4f762c…a99855
INFO [08-10|08:58:49.831] □□ block reached canonical chain           number=7216 hash=05d3ff…1216ab
INFO [08-10|08:58:49.831] Commit new mining work                   number=7222 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:58:49.838] □□ mined potential block                   number=7221 hash=4f762c…a99855
INFO [08-10|08:58:51.393] Successfully sealed new block            number=7222 hash=0ea636…cc8271
INFO [08-10|08:58:51.400] □□ block reached canonical chain           number=7217 hash=182e26…e23d73
INFO [08-10|08:58:51.408] □□ mined potential block                   number=7222 hash=0ea636…cc8271
INFO [08-10|08:58:51.400] Commit new mining work                   number=7223 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:59:01.299] Successfully sealed new block            number=7223 hash=9a3ec5…44b2d5
INFO [08-10|08:59:01.307] □□ block reached canonical chain           number=7218 hash=70e446…8e6f9c
INFO [08-10|08:59:01.315] □□ mined potential block                   number=7223 hash=9a3ec5…44b2d5
INFO [08-10|08:59:01.307] Commit new mining work                   number=7224 txs=0 uncles=0 elapse
d=0s
INFO [08-10|08:59:03.624] Successfully sealed new block            number=7224 hash=cdc162…8e0c18
INFO [08-10|08:59:03.632] □□ block reached canonical chain           number=7219 hash=2d970f…27c0ca
INFO [08-10|08:59:03.639] □□ mined potential block                   number=7224 hash=cdc162…8e0c18
INFO [08-10|08:59:03.632] Commit new mining work                   number=7225 txs=0 uncles=0 elapse
d=0s
```

file:///C:/Users/amjad/BlockChain/code/AC.html

The Current Temperature inside the room is

15 C

temperature too low - Turn heater ON

(c) Room temperature is less than 20

Figure 3.6: Current room temperature

### 3.3.2 Security evaluation

Table 3.1 summarizes how our framework achieves the security requirements discussed in Section 3.2.2. Our framework relies on the Ethereum blockchain. Validated transactions are taken to be tamper-free and it is assumed the user keeps his private key securely. Therefore, only the home owner has control over the blockchain data. The

Table 3.1: Security evaluation achievements

| Requirement | Defense |
|---|---|
| Confidentiality | The use of a pair of private and public keys |
| Integrity | Hash function, Curve Digital Signature Algorithm, and Merkle tree |
| Availability | Only validated transactions by the miner are accepted |

signed digital transaction and the decentralized nature of the blockchain guarantee that attackers cannot access the network or impersonate a real user. Attackers have to gain control over the majority of the network resources or fake the owner's digital signature to control the nodes. In addition, the miner in the framework accepts transactions only from those nodes that have been given both a private and a public key when they are added to the private network.

Blockchains such as Ethereum typically require a set number of "confirmations" from other nodes. This makes sure that the transaction has been mined and correctly embedded into the blockchain, creating multiple third-party "witnesses" to ensure the transaction's authenticity. Even if a hacker somehow hijacks a single node or intercepts instructions, it will prove impossible to hijack all of these simultaneously and trick the network into believing that the device is off when it should be on.

Due to the smart contract acting as the vehicle for the command, and doing so only after multiple third party confirmations, the AC being controlled in this example no longer requires any external accessibility, it can be a read-only node with only outbound connection and that syncs with other blockchain nodes. Security is achieved by proven cryptographic methods, whereby the private key makes it possible to create an unforgeable signature, one which can then be verified as genuine by any third party without any need for access to the private key. Thus, blockchain is in effect a secure database joined with a range of programming options, called smart contracts. A smart contract is simply an uncomplicated computer program. These computer programs can be set up to only trust instructions from an authorized node. This node authenticates them self by a signature from their private key, something which can be kept completely secret. Thus, sending instructions to the blockchain cannot compromise security. There is at present no proven way for an attacker to interfere with a message being sent to the smart contract, as against a client-server system with a central database and a multitude of security layers, all necessary to fend off attackers.

One additional security benefit is the separation of writing and reading from the hardware switch. An old-fashioned client-server system must always protect its database using layers of security, it is where all the important information sits. A blockchain, on the contrary, spreads its information around a network. In a client-server system one can attack a single node to alter a system's state. To affect the same in a blockchain system, it would be necessary to attack and subvert each and every node, practical impossibility.

Thus, IoT home devices are protected from malicious requests and DDOS attacks. In fact, DDOS attacks are one of the critical types of attack that are generally relevant for smart homes [45], a point which is effectively addressed by our framework.

## 3.4 Summary

In this chapter, our objective has been to offer an implementation of secure smart home transactions using Ethereum smart contracts. Each IoT device in a smart home is assigned a private and public key by the miner to ensure data integrity. Each transaction is validated by the miner and only valid transactions with a valid address are mined. This increases smart home availability by limiting the accepted transaction and eliminating the risk of DDOS attacks.

The analysis offered in this chapter is valuable in tackling the first research question of this thesis, by presenting a prototype implementation whereby, Ethereum smart contracts is being used to secure access to smart home devices. This demonstrates the benefits of using a blockchain-based scheme and the relevance of blockchain in data security. In the course of the development of our prototype, the main weakness of Ethereum blockchain encountered was its inability to perform in real time. The transaction time was around 12 second which is high if an immediate response is necessary. Furthermore, the resource constraints of IoT devices are a key challenge when they are integrated with blockchain technology. A large amount of storage is required to record the entire blockchain. Therefore, in order to address this research challenge addressing scalabilty, Chapter-4 presents the integration between blockchain and edge computing.

# Part II

# Security and Privacy Implementation in Smart Homes: Attribute-Based Access Control and Smart Contracts

CHAPTER 4

# Attribute-Based Access Control and Smart Contracts

This chapter is organized as follows: Section 4.2 presents the research background. Section 4.2.1 summarizes the access control scheme; Section 4.2.2 introduces the ERC-20 Token and Section 4.2.3 briefly explains edge computing. In Section 4.3, we review the previous works on blockchain. We propose attribute-based access control and demonstrate our scheme in Section 4.4. The architecture is validated, and an analysis of the results is presented in Section 4.5. Finally, Section 4.6 summarizes the chapter.

## 4.1 Introduction

Constructing a residence with an integrated Internet of Things (IoT) network provides homeowners with outcomes such as increased comfort, security, and quality of life. A smart home network is underpinned by the IoT infrastructure, which connects heterogeneous smart devices (e.g. smartphones, smart meters, wearable devices, etc.). Smart home systems can both enable and enhance people's ability to live independently. They include a suite of invaluable technologies including those to monitor and assess health (a point that we mention only in passing, though we will allude to it briefly below), thus making them attractive to users and device designers. Not surprisingly, it is predicted that by 2022 the value of the global smart home market will exceed $53 billion. This prediction is based on an almost 21 percent annual rate of growth forecast for the market from 2018 to 2022. Although the benefits of smart homes to homeowners and stakeholders are well documented, several risks must also be considered, including cyber-attacks and threats to the data security and privacy of users [110].

Traditional approaches to the resolution of such risks rely on centralised frameworks which are susceptible to cyberattacks. Hence, the access control function is important for preventing access to unauthorised users via explicit or implied specifications and only permitting access to resources for authorised parties. Access controls have traditionally been supported by a centralised system which is relatively simple to manage [116]. This means that a central server is used to process all access controls: namely, assigning access rights, managing access (e.g., updates, revocations), and access verifications. However, there are risks around the server being the point of failure due to 'natural' (functional) or external (cyberattack) forces and potentially compromising the access control system. Furthermore, the large scale and distributed nature of IoT systems means there are difficulties related to controlling requests by centralised schemes to access the desired resource [116].

Distributed access control networks can counter some of the aforementioned limitations of centralised networks. These networks perform the processes related to access control using multiple nodes rather than a single server. The nodes 'agree' on the rights to be assigned, the policies to provide access and the verification results to provide solid and reliable access controls that can resist malicious attacks. As a result, there is growing interest in utilising emergent blockchain technology for distributed and reliable access control.

The emergence of distributed and tamper-resistant ledger-based blockchain techniques to protect data has opened up new possibilities for smart home data privacy, security, and integrity challenges. Blockchain is made up of a digital ledger that records and shares transaction information in the network. Each user has access to secure cryptographic public and private keys to interact with the system. One user can initiate the transaction with his keys, and the other users in the network can accept it with their own keys. Once the nodes agree that the originating user possesses the data he claims, the transaction is accepted; else, it is rejected [184].

Blockchain technology achieves strong performance across a range of smart home applications including control over access to the home, data sharing, and so forth. The implementation of blockchain in smart home networks is also justified on the basis that it exists independently of the current heterogeneous protocols often applied in smart homes (e.g., Z-Wave, Zigbee, Bluetooth and Thread)[103]. Nonetheless, due to the high level of resources consumed during the mining and consensus procedures and the limitations of the node resources in smart home devices, it is challenging to use blockchain directly in a smart home.

Blockchain works as the backbone of the proposed architecture. The end user accesses the device as a node of the blockchain. The access control functions mentioned in the smart contracts are used to authenticate the nodes. All of these constitute parts of our blockchain. The level of security support through the implementation of blockchain is presented in Chapter 3 of the thesis.

Edge computing offers an alternative and complementary method for managing PoW puzzles and supporting the blockchain applications in a smart home. Edge computing takes place at the network extremes (edges) by extending the distribution of cloud-based resources and services. It supports a multi-access system for users to access cloud-like services for enhanced computing, applications, and storage. Resource-constrained smart home appliances can consequently increase their computing capabilities by farming out the mining and storage jobs to edge servers. The incorporation of blockchain and edge computing sets up a decentralised system for computation outsourcing and storage security related to scalable and safety-proof operations [178].

To address the concerns discussed above and motivated by the advantages of integrated blockchain technology and edge computing, we present a novel lightweight Ethereum blockchain-based multi-tier edge-smart home architecture. This moves beyond and improves upon the prototype presented in the previous chapter, using the integration of edge computing to solve storage issues and those around access control to provide an extra level of authentication. In our framework, every single home has multiple edge servers as local blockchain miners and the smart contracts are used to enforce the rules and policies in an automated manner to regulate the smart home's IoT devices based on the Attribute Based Access ABAC. In particular, we present an architecture involving authentication rules and logic based on Ethereum smart contract integrated edge computing.

- We propose ERC-20 token generation and an attribute-based access control mechanism that utilizes Ethereum smart contracts integrated with edge computing(servers) to authenticate user access to IoT smart home devices. The access tokens are issued by the smart contracts with no intermediary or trusted third party.

- We portray the overall system details including the design architecture, workflow scenario, and interactions among entities with the smart contracts including the attribute access control scheme designed to provide protection from illegal data access in a smart home system.

- We include the complete design of the Ethereum smart contract including the implementation and the testing scenarios.
- We discuss the performance evaluation of the proposed scheme and compare this with existing models with respect to various performance metrics.
- We provide security analysis of our proposed authentication scheme and we examine how the scheme achieves security goals (confidentiality, integrity, and availability), and is able to overcome modification and DoS attacks.

## 4.2 Chapter Background

This section provides the background information needed to understand the proposed framework. It discusses the key concepts, access control, ERC 20 tokens and edge computing which sets the stage for the rest of this chapter.

### 4.2.1 Access control scheme

Access control systems are usually based on access control lists (ACLs), which provide users' access permissions. When there is an increase in the number of users seeking resources, ACLs become more difficult to govern. As a solution to this limitation of ACL systems, designers have created role-based access control (RBAC) systems, [9] which add an intermediate layer to the process of distributing role permissions rather than giving them directly to users and then assigning them their roles. This strategy can considerably reduce the time and effort required to monitor access control rules. This is true even when the number of subject roles and resources are increased, or when the system contains many administrative fields. ABAC systems attempt to address the issues associated with an increase in the number of roles by allowing users to apply the subject's attributes directly, as well as the resource and environmental properties. This can be done to describe the access policies and, as a result, reduce the number of rules or rule updates. On the other hand, ABAC still needs to access a consistent description of the field attribute and the definition of attributes across many fields [71].

Research has demonstrated the applicability of attribute-based encryption to share audit-log information and broadcast encryption [66]. In this scenario, the data are stored on the server in an encrypted form while different users are still allowed to decrypt different pieces of data according to their security policy. This effectively eliminates the need to rely on the storage server to prevent unauthorized data access. Moreover, others have published a guide to attribute access control with a definition of ABAC and 149 descriptions of the functional components of ABAC [79]. Also, the guide provides planning, design, implementation, and operational considerations for employing ABAC within a large enterprise with the goals of improving information sharing while maintaining control of that information. Furthermore, attribute-based access control has been used in blockchain architecture. Authors have presented a new digital asset management platform, called DAM-Chain, with transaction-based access control (TBAC) which integrates the distribution ABAC model and blockchain technology [186]. They take transactions as a bridge to integrate ABAC and blockchain into a new platform for resource distribution and sharing. They claim that their proposed platform supports flexible and diverse permission management as well as a verifiable and transparent access authorization process in blockchain-based architecture.

Others propose a distributed ABAC system based on blockchain to enable the trusted auditing of access attempts [138]. In addition to auditability, this system presents a level of transparency that both access requesters and resource owners can benefit from. They present a system architecture with an implementation based on Hyperledger Fabric, achieving high efficiency and low computational overhead. They have validated their solution through a decentralized access control management application in digital libraries.

This chapter examines attribute-based access control in particular because it is deemed to be an appropriate decentralised model for an IoT setup and provides scalability, flexibility, and strong dynamics. Our access control scheme is different from other reported work, in which the authors used three types of access control procedures; device-to-device (D2D) access control, device-to-user (D2U) access control, and device-to-fog server (D2FS) access control to authenticate users in the Internet of Everything (IoE) [20]. Our access control is based on different policies which combine a set of subjects (users), a set of objects (IoT devices), and a set of actions to state that such and such user can perform a given action in the IoT device. The policy is invoked whenever there is an access request from any user or device in the network using the smart contract. Moreover, we integrate the token mechanism to further finalize permissions to access the IoT devices. The smart contract checks the policies and then tracks the token amount to ’who owns’ a particular token and ’how much’ of it to access a certain IoT device.

### 4.2.2 ERC-20 Token

ERC-20 stands for Ethereum Request for Comments, and the number 20 serves as a unique identifier to differentiate it from the other standards. It is a protocol that defines a set of standards and rules for token issues on the Ethereum network and is used to create blueprints for smart contracts based on Ethereum. As a technical standard, ERC-20 has become one of the most important and widely used tokens for all smart contracts on the Ethereum blockchain [26]. ERC-20 defines a set of six functionalities within the Ethereum system for the benefit of other tokens.

1. *totalSupply ()*: to figure out how many tokens have been created and exist in the system.

2. *balanceOf (address owner)*: to return the number of tokens in the account for a given address.

3. *allowance (address tokenowner, address spender)*: the user's balance is one of the most critical data needed to complete a transaction. To carry out a transaction, the user must have a certain number of tokens. If the user does not have the required number of tokens, the allowance () function is used to cancel the transaction.

4. *approve (address spender, unit tokens)*: the contract owner allows the required quantity of tokens to be collected from the contract's address once the user has the required number of tokens for a transaction and the balance has been checked. By comparing the transaction against the total token supply, this function ensures that there are no additional or missing tokens.

5. *transfer (address to, unit tokens)*: this transfer() function enables the contract owner to send tokens. It enables the contract owner to transfer a number of the tokens to other addresses. It also enables a specific number of tokens to transfer between the total supply and a user account.

6. *transferFrom (address from, address to, uint256 tokenId)*: the contract owner can transmit tokens using the transfer() function. This function allows the contract owner to send token amounts to different addresses. Also, it allows a certain number of tokens to be transferred from the overall supply to a user account.

### 4.2.3 Edge computing

The ability of cloud computing to provide limitless processing, data storage, and system administration resources has led to the development of many cloud-based apps and the rapid expansion of Internet-based corporations, such as Amazon, in recent years. The trend recently has been to move cloud functions to network edges [142]. This is dependent on delay-sensitive applications (for example, virtual reality) with strict delay requirements. Edge computing has put more pressure on cloud resources and services to provide mobility, location detection, and lower latency. As a result of these benefits, network edge technology is critical to realise the future of the IoT [180].

The edge computing structure has three levels: end device (front-end), edge server (near-end), and core cloud (far-end). The three-level hierarchy depicts the elements' computing capacity as well as their edge computing characteristics. Sensors and actuators on the front-end provide additional and improved user responsiveness. The resource requirements have to be dispatched to the server, however, given their restricted capacity, near-end edge servers handle most network traffic and a variety of resource needs (such as real-time data processing and computation offloading). As a result of deploying edge servers, end users benefit from improved computation performance at the cost of increased latency. Far-end cloud servers provide greater processing power (e.g., big data analytics) and additional data storage space. The objective of this system architecture is to enable the edge network to support computation-intensive and time-critical applications. Furthermore, certain edge server apps offer data synchronisation via cloud communications.

## 4.3 Blockchain based architecture

Data security and privacy with IoT devices in a smart home is one of the major challenges as connected IoT devices are vulnerable to various attacks and they lack basic security features. To address these issues, numerous centralized solutions have been proposed [149]. Researchers propose an information-centric network-based system for smart home services with a three-layered architecture namely remote cloud, fog layer with smart home servers and end devices [10]. The platform enables real-time systems to be deployed, including smart monitoring and control applications. Another framework proposes integrated existing IoT architecture components [154]. These authors have looked at IoT smart home challenges and solutions to bridge the gap between current state-of-the-art smart home applications and the possibility of integrating them into an IoT-enabled world. Others promote the vision of smart and connected communities (SCC) [156]. They integrate IoT with cyber-physical cloud computing and big data for smart tourism to enhance a community's preservation, liveability, revitalisation, attainability, and security. However, all this work is based on central architecture, where communication and processing overhead, access control, and the single point of failure are major challenges. Therefore, various researchers have turned their attention to distributed frameworks and proposed popular blockchain-based solutions for various IoT use cases.

### 4.3.1 Blockchain Authentication, Access control and edge computing in smart home applications

Researchers have looked at the concerns surrounding gateways or connections between IoT devices, claiming that such centralised arrangements present several security risks such as integrity, certification, and availability [94]. The authors respond by proposing a blockchain-based smart home gateway network that can protect against potential gateway attacks. The blockchain technology network, which is made up of three layers: device, gateway, and cloud, is utilised at the gateway layer to facilitate decentralisation by storing and exchanging data blocks. This maintains data integrity both inside and outside the smart home and availability through authentication and communication between network users. On the other hand, their architecture has some limitations in terms of the computing complexity imposed by blockchain operations at the gateways.

The benefits of using Ganache, Remix, and web3.js architecture for smart home-based IoT blockchain (SHIB) to overcome the difficulties of data privacy, trust access control, and the ability to extend the system were advocated by other authors [36]. They present an IoT gateway for connecting a smart home's cluster of IoT devices to a blockchain network. Their work is complicated by the fact that each user and IoT device must be assigned to one and only one subject-object pair due to the fact that, the gateway may not have enough computer power to handle large transactions.

Some authors have presented a private blockchain-based access control (PBAC) approach to solve data security and privacy issues while using smart devices in smart home systems [174]. Within the IoT system, the proposed PBAC provides "an unforgeable and auditable foundation" that can prevent unauthorised data access, protect data security from threats, and enable accurate, robust, and instant access to information. They only recommend one internet server as an administrator. However, the entire system fails if the administrator is inactive.

Other works proposes utilising a blockchain-based approach based on Proof-of-Authority to develop a consensus mechanism to better manage home appliances in a decentralised framework [149]. When compared to a standard Proof-of-Work based system, the authors demonstrate additional features to improve the effectiveness of a blockchain method using proof-of-authority as the consensus mechanism to address security concerns.

The implementation of IoT and blockchain-based multi-sensory frameworks in the context of in-home quality of life (QOL) for recently diagnosed cancer patients has been studied [131]. Multiple medical and ambient intelligent IoT sensors can capture QOL data from the smart home environment and securely share it with a specified community of interest using the authors' suggested blockchain and off-chain-based framework. The in-home secure monitoring system captures QOL data, such as transactional records and multimedia-based big data (e.g. physiological and mental state data), which the authors can manage using blockchain-based data analytics.

Another author proposes a blockchain-based IoT architecture that reduces the impact on IoT devices while maintaining the most of traditional blockchain's security and privacy advantages [47]. An overlay network can be created using high-resource devices to employ a public distributed blockchain that assures privacy and security at all stages of the transaction process. Furthermore, it employs distributed trust to provide excellent security and privacy for IoT applications, and it minimizes the time necessary to execute block validation. However, no information on the establishment of this scalable blockchain or the security certificates is provided.

One author has implemented IoT-based architecture in tandem with BC (Hyperledger Fabric) to assess the validity of the communicating devices to identify whether they are normal or malicious [8]. They have tested their scheme in a smart home-based scenario. However, the transaction size in Fabric is larger than other blockchain platforms because they also carry the certificate information for approval. Therefore, the latency becomes worse with increase in block size in their scenario.

Other authors have integrated both blockchain and a group signature to anonymously authenticate group members, as well as a message authentication code to efficiently authenticate home gateways without leaking information in a smart home scenario [96]. In HomeChain, all the request records from group members (or revocation requests from the group manager) will be chained into the blockchain. Due to the immutability of blockchain and the traceability of a group signature, it is not easy to tamper with or delete these records and hence they may provide reliable auditing. However, as there was no access control policy, they have adopted a revocation list to revoke the authorities of malicious users.

Others authors propose an ABAC framework for IoT systems through the use of Ethereum smart contract technology [181]. The system is made up of four smart contracts that manage ABAC rules, subject and object attributes, and access control. However, the main drawback of their framework is that, the average time for access control is high due to the complex interactions between the access control contract and other smart contracts to retrieve attributes and policies.

Researchers have developed a smart contract structure for distributed and trustworthy access control IoT systems [182]. The proposed structure includes numerous access control contracts (ACCs), one judge contract (JC), and one register contract (RC). However, only one subject-object combination is handled by a single ACC. A larger implementation cost is implied by an increasing linear relationship between gas costs and system subject-object pair counts.

It must be acknowledged that the work presented by other researchers are found to be encouraging, however, there are certain limitations on computational complexity covering parameter such as, computing cost, time requirement, etc .. In this thesis we develop and integrate a novel architecture which would integrate the access control scheme within two smart contracts deployed in multi-edge servers to achieve a secure distributed blockchain to serve smart home IoT devices. The use of multi-edge servers as an admin provides a complementary way to overcome the computation cost and single point of failure. We also investigate one of the popular blockchain technologies, Ethereum smart contract and ERC-20 token generation, for the implementation of a simulation smart home devices.

## 4.4 Proposed Attribute-Based Access Control Scheme

The core architectural and design features of our proposed blockchain-based system are described in this section, which uses Ethereum smart contracts to register and manage the home user, IoT smart home devices, and the edge server.

### 4.4.1 System Architecture

The proposed system architecture is shown in Fig 4.1. The architecture is made up of four primary participants, each of which has Internet access to Ethereum smart contracts: end users (home users, services accessors), IoT smart home devices, edge servers, and the cloud servers hosting IoT data. Each IoT smart home equipment has its own Ethereum address (with public and private keys). All other parties have their own Ethereum Addresses (EA) and interact with the smart contract directly, either through an Ethereum client for edge and cloud nodes or a front-end application/wallet for end users.

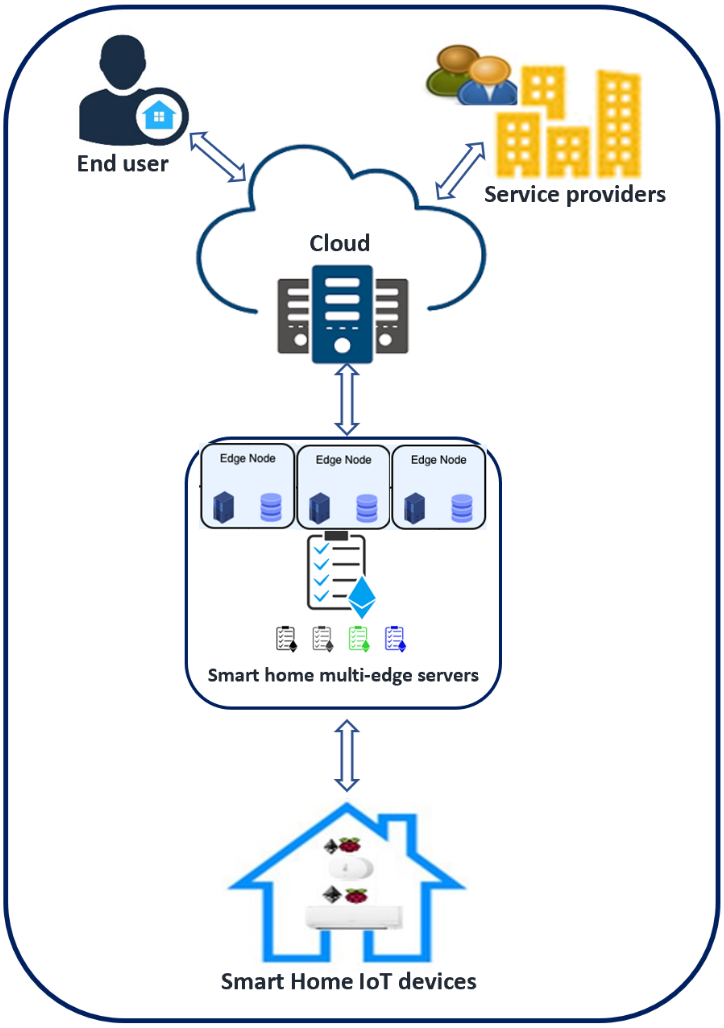


Figure 4.1: System architecture

The key roles of the various system participants are summarised below:

1. End user: Requests access permission through the smart contract to access certain smart home devices.

   a: Home user: A user device (e.g., PCs, laptops, smartphones) is a device that allows users to access services supplied by servers (e.g., monitoring the current temperature of his or her home).

   b: Service accessors: Any service providers such as health care, police or other parties who need to access the smart home data to provide any type of service. We offer these as examples of why service accessors might exist; the implementation details are not within the scope of the present research.

2. IoT devices: Sensors and actuators are the two main types of IoT devices in the system. Sensors can collect environmental data (such as temperature) and send it to edge servers or storage devices for later use. On the other hand, actuators can perform operations (such as turning on the air conditioner) in response to a user's command.

3. Smart home multi-edge servers (Admin Edge): An edge node is a device or a set of devices that can communicate with IoT and storage devices to provide a variety of services. Examples of interactions between servers and other peers include collecting environmental data from sensors, issuing commands to actuators to conduct certain activities, and accessing or writing data to storage devices. Edge nodes process all incoming and outgoing transactions and use a shared key for local communications with IoT devices and local storage. This maintains the smart contracts that manage the registration of the end users and IoT devices and authenticates the end users to access the IoT devices. Because IoT devices lack sufficient processing power, only edge servers can perform the mining operation.

4. Cloud: Provide long-term data analytics and storage. The resources in the cloud can also be configured as nodes on blockchain to ensure the privacy and integrity of the data in the system.

### 4.4.2 Attribute-based access control and Smart contracts

To avoid complexity of in a single smart contract, the proposed framework consists of two Ethereum smart contracts, namely the register contract and the access contract. The first contract is responsible for storing and managing (e.g., updating, adding, deleting) the subject and object attributes, as well as policies. The access contracts are responsible for controlling the access of the IoT devices by generating ERC-20 tokens and finalising the permission to access the IoT devices. The smart contracts are introduced as follows:

1. Register contract: The policy is deployed on the blockchain to register and manage the attributes of users and IoT devices as shown in following code. Only the administrator has the permission to execute this contract. A relevant code is posted on GitHub (https://github.com/jklepatch/eattheblocks/blob/master/dapp-30/day5-crud/smart-contract/Crud.sol).

```
pragma solidity ^0.5.0;
contract Add{
struct User {
    uint256 id;
    string name;
    bool set; // This boolean is used to
    differentiate between unset and zero struct values
}
 address owner;
 modifier onlyOwner() {
       require(owner == msg.sender);
            _;
}
mapping(address => User) public users;
function createUser(address _userAddress, uint256 _userId,
                    string memory _userName) public onlyOwner
    {
    User storage user = users[_userAddress];
    // Check that the user did not already exist:
    require(!user.set);
    //Store the user
    users[_userAddress] = User({
        id: _userId,
        name: _userName,
        set: true
    });
}
}
function deleteUser(address user) public onlyOwner
    {
    if (user.length<2)
    throw;
    else {
            unit i=0;
            while (i<user.length){
                  if(user[i] == user){
                     delete user[i];
                     userDeleted(user, msg.sender);
                  }
                  i++;
                }
        }
    }
```

Each user and IoT device has a unique identifier (Ethereum account address) and multiple attributes associated with its ID. This contract has the function of managing the subject and object attributes such as adding, deleting and updating.

A user struct is created to save the address of the new user. Mapping is introduced; namely: users where the users link a user's address to a set of users it can access. The deletion functions used within the contract are restricted modifiers for owner users to execute. deleteUser function allows the deletion of users from the system. However, The contract would reject the delete operation if less than two users remain in the system and throws an exception. In our system, we use one laptop device to simulate the edge server running two miners; therefore, we need at least two users to keep running the miners.

Also, this contract specifies the policy associated with each user and IoT device based on user type, as described below:

```
contract Request Access{
     function checkAttrebute (addressOf User)
      attribute my_at = attribute (addressOf User);
     function GetPolicy (addressOf User)
      Policy my_po = Policy (addressof user);
```

```
       if (my_at.checkAttrebute() == true & my_po.GetPolicy ()== true)
            return my. sendToken()
       return FAILURE;
}
```

A policy is a statement that combines a set of subjects (users), a set of objects (IoT devices), and a set of actions to state that this user can perform such and such an action on the IoT devices. An example of a policy is shown in Table 4.1.

Table 4.1: Example of user attributes, IoT attributes and permissions.

| User attributes | IoT Device attributes | Action |
|---|---|---|
| UserAddress | IoTAddress | Execute |
| UserType | IoTName | Read |
| UserName | IoTFun | write |

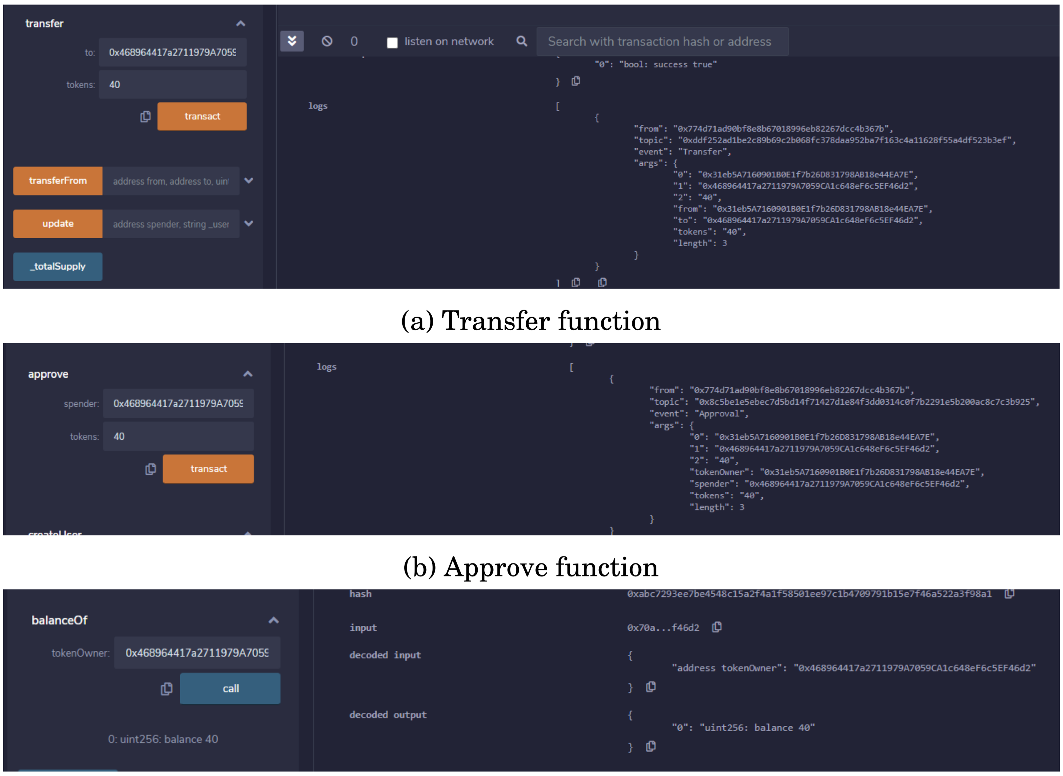
2. Access Contract: This contract governs access requests from users (subject) to IoT devices (object). As following code shown , the user executes this contract to request a token to communicate with an object.

```
contract Attribute is ERC20Interface, Owned(
    string public Symbol;
    string public decimals;
    mapping (address => unit) balance;
    mapping (unit256 => AttributeData) checkAttribute;
    mapping (unit256 => Policy) GetPolicy;

    event Sendtoken (address from, address to, unit tokens)
    struct AttributeData{
         unit256 AttributeID;
         string Attribute;
         string approve,
     }
     struct Policy{
         unit256 PolicyID;
         string Policy;
         string approve;
     }
 function AttributeToken () public {
         balances [msg. sender]= 100 ;
         totalSupply = 100;
         name = "ACoin";
         decimals = 0;
         symbol = "A";
     }
 function checkAttribute (unit256 AttributeD, string Attribute,
 string approve) public
 returns (bool success)

 {checkAttribute[AttributeID]= AttributeData(AttributeID,
 Attribute, approve);
 return true;
 }
 function GetPolicy (unit256 PolicyID, string Policy,
 string approve) public
 returns (bool success)
        {GetPolicy[PolicyID] = Policy (PolicyID , Policy, approve);
 return true;
 }
function sendToken (address to, unit tokens) public
returns (bool success)
        {require (! frozenAccount[to]);
        emit sendtoken (msg. sender, to, tokens);
 return true;
```

This contract includes functions for validating subject attributes and checking policy; the access contract (AC) assesses whether the subject has the right to perform an action on the object based on the policy received, and then sends a token to the subject. The main functions in the contract are Check attribute(), Get policy() and TransferToken(). This contract is also in charge of generating ERC-20 tokens. Fig 4.2 illustrates how to use some access contract functionalities. To prevent a valid user from flooding the network with access control requests, each user has a specific number of valid tokens at a time dependent on user type.



to: 0x468964417a2711979A7059

tokens: 40

transferFrom: address from, address to, uint

update: address spender, string _user

```
                                        "0": "bool: success true"
                                    }
logs                                [
                                        {
                                            "from": "0x774d71ad90bf8e8b67018996eb82267dcc4b367b",
                                            "topic": "0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef",
                                            "event": "Transfer",
                                            "args": {
                                                "0": "0x31eb5A7160901B0E1f7b26D831798AB18e44EA7E",
                                                "1": "0x468964417a2711979A7059CA1c648eF6c5EF46d2",
                                                "2": "40",
                                                "from": "0x31eb5A7160901B0E1f7b26D831798AB18e44EA7E",
                                                "to": "0x468964417a2711979A7059CA1c648eF6c5EF46d2",
                                                "tokens": "40",
                                                "length": 3
                                            }
                                        }
                                    ]
```

(a) Transfer function

spender: 0x468964417a2711979A7059

tokens: 40

```
logs                                [
                                        {
                                            "from": "0x774d71ad90bf8e8b67018996eb82267dcc4b367b",
                                            "topic": "0x8c5be1e5ebec7d5bd14f71427d1e84f3dd0314c0f7b2291e5b200ac8c7c3b925",
                                            "event": "Approval",
                                            "args": {
                                                "0": "0x31eb5A7160901B0E1f7b26D831798AB18e44EA7E",
                                                "1": "0x468964417a2711979A7059CA1c648eF6c5EF46d2",
                                                "2": "40",
                                                "tokenOwner": "0x31eb5A7160901B0E1f7b26D831798AB18e44EA7E",
                                                "spender": "0x468964417a2711979A7059CA1c648eF6c5EF46d2",
                                                "tokens": "40",
                                                "length": 3
                                            }
                                        }
```

(b) Approve function

tokenOwner: 0x468964417a2711979A7059

0: uint256: balance 40

```
hash                    0xabc7293ee7be4548c15a2f4a1f58501ee97c1b4709791b15e7f46a522a3f98a1

input                   0x70a...f46d2

decoded input           {
                              "address tokenOwner": "0x468964417a2711979A7059CA1c648eF6c5EF46d2"
                        }

decoded output          {
                              "0": "uint256: balance 40"
                        }
```

(c) Token balance

Figure 4.2: Example of access contract functions execution

### 4.4.3 System design

The proposed system provides authentication for users using an attribute access contract and token distribution. Fig 4.3 illustrates typical attribute-based access contract transactions with this authentication mechanism. Users can remotely access or control home devices using the freshly generated token so that only the requester is able to receive the response from the legitimate home admin. We now describe four phases in our system: Initialization, Request Control, State Delivery and Chain Transaction.

1. Initialization: For illustration purposes, we assume that family members constitute a group of users from amongst whom a group admin is chosen. An admin invokes the Register Contracts to add other users and IoT devices. Users allocate their Ethereum address (EA) and individual private keys for signing transactions. Correspondingly, each home admin holds the group public key for transaction verification. The admin is run in different miners in multiple edge nodes to avoid a single point of failure.

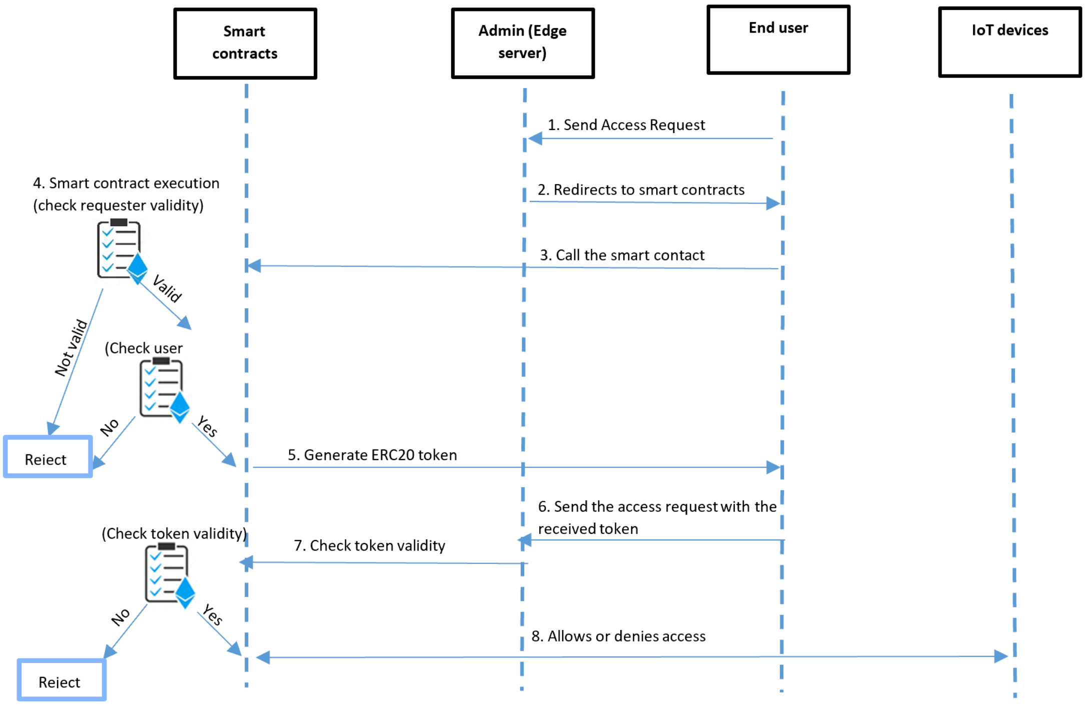
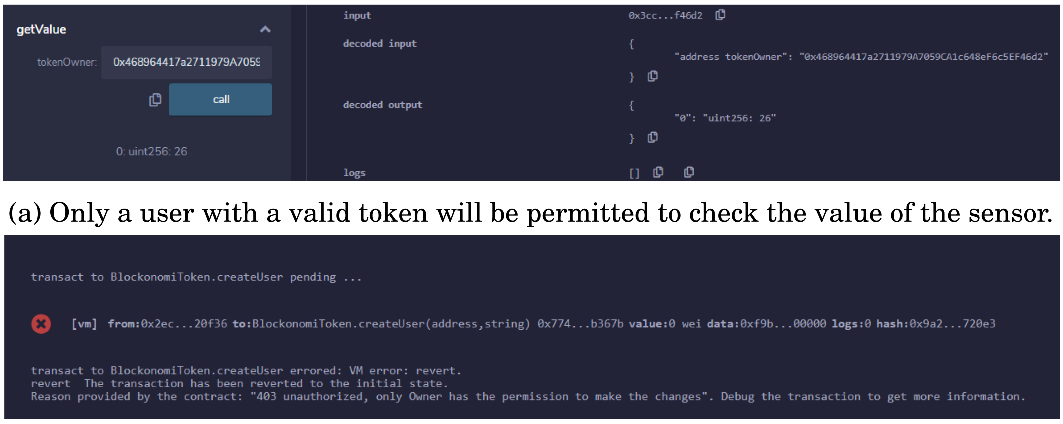


Figure 4.3: Typical transactions in proposed scheme

2. Request Access: When a user wishes to publish an access or control request with the home admin, a token is generated for a certain duration and exact access time. This is the suggested approach to avoid replay attacks and profiling. After obtaining the token by invoking TransferToken () from the access contract, the user constructs the transaction based on his/her requirements. For example, the user requests the room temperature, the transaction is computed after the user is redirected to the smart contract and request token, and three main functions are invoked in that contract: Check attribute(), Get policy(), TransferToken(). The user sends the received valid token with the request access to the admin, and if the user has a valid token, then the access will be granted. The output of valid and invalid user requests for accessing data on room temperature is shown through the screen shots in Fig 4.4.

3. State Delivery: The home admin monitors the smart contract for new requests once a user requests a new access or service. If the transaction passes the verification, then the home admin checks the token validity and grants or denies access to the IoT device.



```
tokenOwner: 0x468964417a2711979A7059
0: uint256: 26

input            0x3cc...f46d2
decoded input    {
                         "address tokenOwner": "0x468964417a2711979A7059CA1c648eF6c5EF46d2"
                 }
decoded output   {
                         "0": "uint256: 26"
                 }
logs             []
```

(a) Only a user with a valid token will be permitted to check the value of the sensor.

```
transact to BlockonomiToken.createUser pending ...

 [vm]  from:0x2ec...20f36 to:BlockonomiToken.createUser(address,string) 0x774...b367b value:0 wei data:0xf9b...00000 logs:0 hash:0x9a2...720e3

transact to BlockonomiToken.createUser errored: VM error: revert.
revert  The transaction has been reverted to the initial state.
Reason provided by the contract: "403 unauthorized, only Owner has the permission to make the changes". Debug the transaction to get more information.
```

(b) User without enough tokens or if an unregistered user requests to check the temperature.

Figure 4.4: User request for room temperature data

4. Chain Transaction: Admin nodes (miners) are responsible for retrieving transactions in the smart contract and compete with each other to be the first to successfully solve the PoW for chaining the block to the blockchain. In order to reach consensus, the miner publishes his solution to the blockchain network once he has solved the PoW problem. Those who mine the first block to reach consensus are awarded the mining reward.

### 4.4.4 Implementation

The detailed hardware and software configurations are as follows. Our system is developed on a private network. The model runs on a Private Ethereum network which consists of one laptop device (Dell XPS ) to simulate the edge server running two miners connected to two single-board computers (Raspberry Pi 3 Model B) which are simulating the temperature sensor and LED and one home user laptop. The edge server is equipped with 4 independent CPU cores and 16 GB RAM. One processor core is dedicated to the mining environment, while the remaining processor cores are dedicated to the edge computing service. The miner can boost up to 3.5 GHz CPU, 8 GB RAM, and 1 TB storage. As the IoT devices, two Raspberry Pis have 1.2 GHz CPU, 1 GB RAM, and 32 GB storage with accessory modules including a temperature sensor and LED sensor. The laptop as home user has 2.2 GHz CPU, 16 GB RAM, and 256 GB storage.

In the edge server, the blockchain running framework is Go-Ethereum, and Solidity is used as the development language for smart contracts. Remix integrated development (IDE) is used to write and compile the contracts (Remix 2020). This uses Solidity as the language to write smart contracts. Web3.js (Ethereum JavaScript API) is also used in the model to deploy and compile the contracts and to monitor the contract state. JavaScript is used to interact with the corresponding geth client via the HTTP connection. A simple HTML web page is built to support the interaction between the home users and the devices. A light version of the Raspbian operating system and Go-Ethereum have been loaded on the Raspberry Pis, which disables block mining function. Windows 10 home (64bit) version is used in the home user laptop.

In the testbed, the first laptop's operation supports two edge service providers and a block miner solving a PoW puzzle. The Raspberry Pis and the second laptop act as blockchain clients generating and sending transactions of resource requests to the edge server. With the above setup, the edge server functions as a full blockchain node, storing all transactions, running predefined smart contracts, and creating new blocks. Only transaction data is stored on the IoT devices, which function as light nodes in the blockchain.

The private blockchain is configured following a series of steps, including the selection of a compatible version of Ethereum, the use of the Windows power shell to initiate geth, and the requirement for each node to satisfy multiple requirements before being able to join the blockchain. This includes (1) initialisation of the genesis file (Test.json) to create the first block, (2) use of the network ID to connect to the same blockchain, and (3) initialisation of the private blockchain using a geth command. An account with a private and public key is created by the miner for each node and indexed according to its address whereby it can interact with other nodes and smart contracts. The geth on each node is then started using a command which includes different flags for different functions. In order to prevent external attackers from gaining access to the nodes, the "no discovery" flag has been set on all nodes. A specific command is then used to retrieve the node ID to allow syncing to occur. This last step is repeated to add the two Raspberry Pis as nodes and the home user laptop to create a private blockchain with fully synchronized nodes.

It is important to note that while the edge server has full rights to access all features, other users and IoT devices are only given authorisation to use a subset of those features under the smart contracts. In the event that a user or vulnerable devices are compromised and used to carry out malicious operations, this configuration reduces the harm.

## 4.5 Evaluation and Analysis

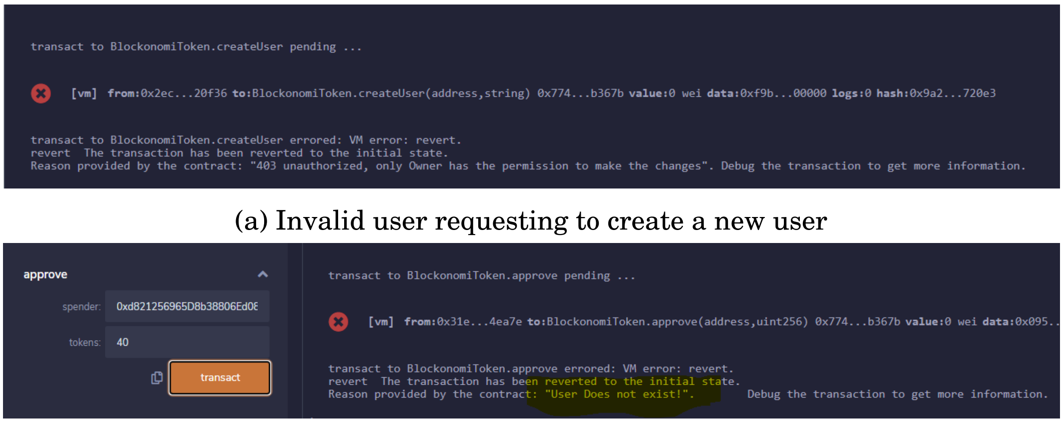
This section provides a complete discussion of the security and performance of the attribute-smart contract-based edge scheme. In this section, we briefly define the possible threats and attacks and then discuss the handling techniques to ensure satisfaction of the security goals of the CIA triad, namely that, confidentiality, integrity and availability are satisfied. Authentication and access control are provided in our architecture to address these goals.

### 4.5.1 Security analysis

Confidentiality aims to ensure that unauthorised users are prevented from gaining access to IoT devices and their data and to make sure that private data is delivered only to the intended users. One approach to achieving confidentiality is message encryption using an SSL session after authenticating the user successfully [20]. As a powerful feature of blockchain, our framework assigns a unique 20-byte Ethereum addresses (EA) directly to authorised node (including IoT devices) with almost no collision. EA has asymmetric public key pairs that can be used to establish a secure SSL session for communication between any authenticated nodes such as an authenticated user or IoT device. During the private network formation,the miner distributes private and public keys associated with the EA for each node. The temperature sensor or the LED, as the sender node, utilises the private key to provide a digital signature, allowing the requested transaction to be broadcast across the entire network.

In terms of availability, our architecture leverages the inherent properties of blockchain technology, which offer reliability and robustness. Because of the decentralised structure of blockchain and the ledger replication in multiple locations, there is no possibility of a single point of failure and all data is circulated via multiple nodes. A copy of the transaction history is stored in each admin node, enabling it to be verified and linked back to the initial transaction. Moreover, to increase smart home availability, IoT devices are protected from malicious requests by limiting the accepted transactions to those with users who have a valid token. So, every transaction received is authorised by the admins before forwarding it to the IoT devices.

Furthermore, the use of a valid token increases the level of security in our architecture. This can be observed as only the admins can issue a valid token and only the intended user can use the token. Fig 4.5 shows the revert error when anyone other than the admin tries to create a user or issue a token. Also, the token's owner cannot transfer the token to any other users, so if the public key of a user is compromised, the smart contract construction prevents token transfer. The admin will only allow transactions that have a valid token associated with a valid user to be accepted in the network.



```
transact to BlockonomiToken.createUser pending ...

[vm] from:0x2ec...20f36 to:BlockonomiToken.createUser(address,string) 0x774...b367b value:0 wei data:0xf9b...00000 logs:0 hash:0x9a2...720e3

transact to BlockonomiToken.createUser errored: VM error: revert.
revert   The transaction has been reverted to the initial state.
Reason provided by the contract: "403 unauthorized, only Owner has the permission to make the changes". Debug the transaction to get more information.
```

(a) Invalid user requesting to create a new user

approve

spender: 0xd821256965D8b38806Ed0E

tokens: 40

```
transact to BlockonomiToken.approve pending ...

[vm] from:0x31e...4ea7e to:BlockonomiToken.approve(address,uint256) 0x774...b367b value:0 wei data:0x095..

transact to BlockonomiToken.approve errored: VM error: revert.
revert   The transaction has been reverted to the initial state.
Reason provided by the contract: "User Does not exist!".        Debug the transaction to get more information.
```

(b) Invalid user requesting a token

Figure 4.5: Revert transaction

1. Denial of service (DoS) Attack: In this type of attack, the attacker sends a large number of transactions to the target in order to disrupt its availability. The use of attribute-based access control smart contracts in our architecture reduces the effect of this attack since only authorized transactions will be accepted. The admin has to examine the address and policy for each user and device to issue a valid token to send a transaction. If the admin receives several unsuccessful access requests from an unauthorized entity, it can block that transaction and reject it. Furthermore, the policy is enforced automatically by the smart contracts. If malevolent outsiders compromise and control the IoT devices for malicious activities, such as making continuous resource requests or initiating DoS attacks, the smart contracts will execute automatically based on the preprogrammed policies of the total token supply, the access time, and the duration. For example, in our scenario we specify the total token supply as 100 from each user and if users or devices request access, the request contract will issue one valid token at a time; if the requests exceed the number of tokens supplied, the transaction will be rejected.

   In our experiment we built a prototype to simulate only two smart devices in a smart home scenario with one admin for the sake of testing the architecture. However, in a large-scale scenario, the network will have more than one admin and many nodes to mine and the architecture will use the advantages of the distributed and immutable ledger of blockchain. So, by design, it would appear that blockchain is equipped to face up to and withstand a DDoS attack. In the first place, it eliminates the risk of having a single point of failure. It can maintain a list of compromised IPs in its ledger, and this would be resistant to disruption attempts. As soon as a server with the list is compromised, a user can switch to any other node on the network to access a safe copy.

2. Modification attack: In this form of attack, the attacker may try to alter or delete the stored data of a particular user or device. To launch this attack, the attacker has to compromise the local storage security. Different cases of modification attacks have been discussed in blockchain-based information sharing frameworks. It is claimed that the implementation of a smart contract protocol prevents the adversary from breaking the security of their proposed scheme [50][49]. Similarly, in our scheme only the admin has the right to store, delete, or update the data, based on the policy in the smart contracts. All the information about users, devices, and policy, is shared between the edge nodes and the cloud; assuming the adversary wants to change or modify the ID of a user or any device, the change will be detected by the edge nodes since every block contains its previous hash block and a change in one block will result in a break in the chain.

   The authentication and access control threat is the next type of threat. According to published work, an attacker may be able to take control of a smart home device or install a fake device into a home network [55]. To protect against these threats, we have implemented a hierarchical defense mechanism in our design. In order to prevent smart home devices from being directly accessed via the Internet, there is an admin node that manages all incoming and outgoing transactions. A transaction is dropped if the admin notices that it does not adhere to the contract's policies.

   Secondly, every device in the home must have a unique address and follow the same genesis transaction in the local blockchain that allows each to communicate with the admin and other devices. A device is isolated from the network if it lacks a unique address and genesis transaction. As a result, an attacker will be unable to connect to the network and install malicious devices.

### 4.5.2 Performance analysis

To evaluate the performance of the proposed model, we are conducting experiments in a private Ethereum network where the edge server represents the home admin to add a home user, as well as the two sensors (temperature and LED). The home user requests the room temperature to turn on/off the AC (change the state of LED) based on temperature. The admin checks the user's validity and then gives access to the user as described previously in the system design section. We simulate two types of transactions in a smart-home setting i.e. store and access. Here, we investigate the store transaction (adding a new user or IoT device using the register contract) and the request access transaction to invoke some data (using the access contract). We evaluate the block size, gas cost and time cost by comparing our scheme with other works [8, 96, 181, 182].

1. Block size: The block size in Ethereum is based on the contracts being run and the associated number of transactions known as a Gas limit per block, and the maximum can vary slightly from block to block. Depending on how much gas each transaction spends, transactions are combined in the form of blocks. There are 280 storage transactions and 300 access transactions in a 1MB block. The storage size is 2.80KB and the access transaction size is 4.00KB. More than 200 registrations can be stored with an average block size of 130KB.

   Since the size of the block is the key factor that impacts overall latency, in our experiment, we find the block size varies between 118 KB to 145 kB based on the contract being executed. We evaluate the interaction delay of register contract and access contract which are important to ensure system effectiveness.

   Fig 4.6 shows that the time for one transaction to be completed is less than 30ms for the register contract and 50ms for the access contract. Such a delay should satisfy the latency requirement of real-time applications.

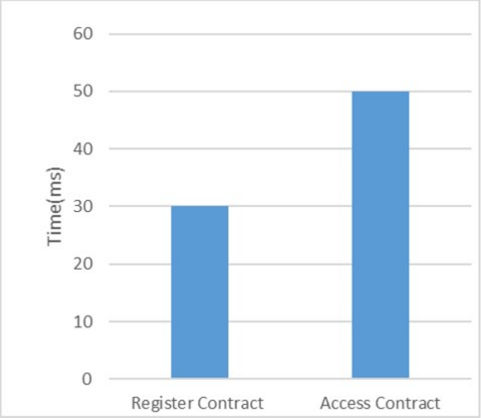


Figure 4.6: Time to complete one transaction

However, latency worsens for the register contract as the block size is increased. The latency increases due to the increased time needed to include the transaction in the block and the increased bandwidth required to propagate a bigger block through network. However, the completion of new block validation and transmission is faster since the edge server has more computing and bandwidth resources. On the other hand, when compared with other work [8], IoT-BC is based on Fabric architecture, which in general has a larger transaction size because it carries the certificate information for approval. As a result, the total increase in transaction latency in IoT-BC is 22.45% while in our scheme it measures 20.23%.

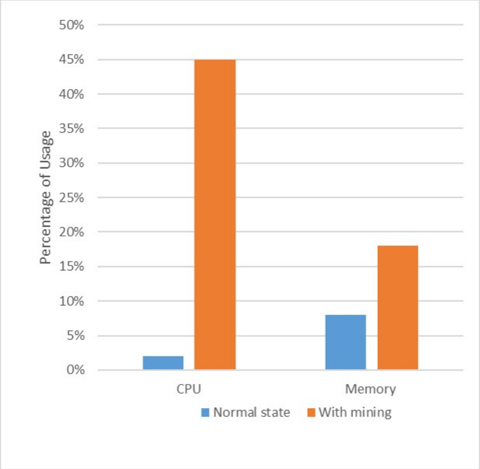


Figure 4.7: Resource usage for single transaction

The CPU and memory usage are also explored, as illustrated in Fig 4.7. We realize that a very low percentage of CPU resources are taken by regular transactions, while memory usage is slightly greater since the blockchain client uses 8% even when in a normal state. However, we note that in a real smart home environment, the number of IoT devices connected will be increased and this will have a possible impact on blockchain overhead. Since the miner is located on the edge server, mining, verifying, and storing new blocks will increase the computing resources used. Therefore, specifying the number of IoT devices to be managed by one edge server, or launching more VMs as the miners to share the load of the computation required is recommended.

2. Gas cost: In the deployment of smart contracts on the blockchain and execution of these contracts, Application Binary Interfaces (ABIs) require a fee to be paid to the miner which mines the block. The amount of gas needed to execute an operation,

such as implementing a smart contract or executing an ABI, is measured by Ethereum using a unit called gas. In general, with a more difficult task, more gas is used. The price of gas varies depending on the time. A task's fee is determined by the amount of gas used and its cost. Table 4.2 lists the amount of gas paid for some functions, such as adding a subject or policy, or deploying or executing the ACC.

Table 4.2: Calculated gas cost

| | Proposed Scheme | Scheme in [182] | Scheme in[181] |
|---|---|---|---|
| AddUser | 85,662 | - | 152,863 |
| AddPolicy | 360,273 | 128,777 | 363,964 |
| DeployACC | 1,377,071 | 1,706,290 | 1,301,972 |

In our proposed scheme, the gas amount required to deploy the access contract is 1,377,071, which is more than in the existing schemes compared here. We can observe from the table that when we compare with the proposed ABAC framework in reported work [181], it consumes less gas than our scheme. This increased value is due to the relatively complex interactions in our scheme for retrieving attributes and policies between the access contract and admin policy smart contract and authority contract.

However, in another piece of work, , one ACC is deployed for only one subject-object pair [182]. The gas cost increases linearly as the number of subject-object pairs of the system increases, while in our proposed system, there is no need to deploy a new access contract when the subject and object increase. This results in less gas consumed and hence, less cost. Moreover, when comparing the gas cost for performing functions such as add user or add policy, our proposed scheme consumes less gas for the same functions in the scheme [181].

3. Time cost: The approximate time cost for executing the access contract is 40 seconds in our proposal which is more than the 36 seconds average time for ABAC presented elsewhere [181]. This is due to the time it takes to invoke the token in our proposed scheme and the extra time needed to check token validity and call other smart contracts. However, the fresh onetime token generated during each access request is used for securing the session, and this ensures data confidentiality, which is worth the difference of a few seconds. Furthermore, the time to deploy our access smart contract is around 185.83 seconds compared to the framework deployed in

other research [96]. This is due to smart contract invocations (i.e., getRequest, getRL, uploadResponse, and getResult).

Note that the ABI's execution time varies based on a variety of factors such as the system's CPU capacity, network design, mining schedule, and so on, and therefore the execution time may change amongst Ethereum networks.

## 4.6 Summary

This chapter evaluates a real-time interaction model between home users and a fully validating private blockchain node through the use of attribute-based access control to authenticate smart home users and IoT devices. By combining blockchain technology with attribute based access control and edge computing, this model solves the problem of the traditional access control method which is based on centralized design to meet the access control requirements in IoT.

In this chapter, we have described the development of Ethereum blockchain with multiple smart contracts, and the implementation is detailed to demonstrate the feasibility of the framework. Compared with existing schemes, our proposed scheme achieves more fine-grained access control with fresh token generation and less computing cost with edge computing. Our framework also achieves the desired security goals and is resilient against modification and DoS attacks.

The analysis offered in this chapter is valuable in tackling the second research question of this thesis, by comparing our result with existing work and shows that we archive better time and gas cost than others. We also answer the fourth question by showing how our scheme is effective against current threats such as Denial of service (DoS) Attack and modification attack.

# Part III

# Privacy-Preserving Mechanism in Smart Homes using Blockchain

CHAPTER 5

# Privacy-Preserving Mechanism in Smart Homes using Blockchain

The chapter is organized as follows. Section 5.2 introduces the chapter. Section 5.2.1 summarises the privacy issues in blockchain-based IoT. In Section 5.2.2 we review the strategies to preserve blockchain privacy. In Section 5.3, we discuss differential privacy. We propose a layer architecture and demonstrate our scheme in Section 5.4. The privacy evaluation is discussed in Section 5.5. Finally, Section 5.6 summarizes the chapter.

## 5.1 Introduction

While blockchain is regarded as the future of data storage due to its decentralized structure, several issues are yet to be resolved before it is implemented in daily life scenarios. A significant parameter in blockchain applications that needs further development is data preservation and transaction privacy. Blockchain user identification across the decentralized network is supported by the public key. As a result, identities do not all remain private or anonymous. An adversary in the role of a third-party may analyse the transactions on the network and potentially infer the identities of other users. In addition, blockchain's decentralized structure allows unprotected blockchain scenarios to be observed. Moreover, additional privacy features are needed to better protect personal data on the blockchain nodes. With financial blockchain systems for instance, the transaction details are broadcast across the decentralized network whenever a transaction takes place [56]. This broadcasting occurs to safeguard each blockchain node with up-to-date information. Furthermore, the ledger recording the transaction remains uniform across the network. An adversary may use this information to monitor an individual and go back through the transaction details to discover transaction information. Moreover, with regard to blockchain-based IoT devices, an adversary may compromise the information exchange between devices for illegal purposes.

Furthermore, there are also privacy risks associated with applying blockchain in other sectors such as financial, real estate, and asset management [55]. That is, blockchain's distributed nature means that the individual's identity or personal information may be leaked during transactions. To date, the literature in the field on how to preserve the individual's privacy in blockchain has mostly focused on anonymization strategies and their derivatives [18]. However, studies show that anonymization cannot ensure total privacy because of the potential to combine anonymized data with similar datasets to discover personal information [38].

To overcome the aforementioned issues and provide privacy protections, it may be useful to integrate differential privacy based on machine learning with the use of the latest blockchain technology. Differential privacy is efficient at preserving privacy in statistical databases and real-time settings [76]. Differential privacy is an approach to preserve the confidentiality of data without risking its leakage by adding noise to data without influencing the correct output of the data analysis result. Therefore, in this chapter, we aim to prove that the machine learning scheme has the same or better accuracy when differential privacy is employed and that it should therefore be utilised for the sake of increased privacy. This chapter therefore compares the accuracy of machine learning with and without the differential privacy as an initial test of the concept, rather than taking the machine learning algorithm as a topic for discussion for its own sake.

The use of differential privacy can create a level of indistinguishability in statistical blockchain data, leaving the analyst unable to predict with any certainty the accessibility of individual blockchain nodes. All the nodes are taken to be part of the blockchain network. In the present chapter (and building upon these matters with larger datasets in the following chapter) we investigate the advantages of differential privacy in the attempt to optimise privacy between the smart home devices and both the home user and the cloud. Differential privacy is a good fit for use in blockchain technology to preserve the individual's identity during a broadcast. While ensuring that the information remains useful for completing transactions, differential privacy can still perturb appearance of the person's identity to the network such that an adversary will not be able to determine the sender's or the receiver's actual identity. Thus, differential privacy can help to keep sensitive/personal information private in a dataset. Differential privacy in blockchain applications may thus prove to be beneficial to protect data privacy [76].

## 5.2 Summarises the Privacy Issues in Blockchain-based IoT

Blockchain technology relies on authentication and encryption services to preserve data security (i.e., secure transactions). Cryptography and the use of public key encryption are linked to such blockchain services. This means that users must have access to both the public and private keys to manage their transactions.

Two types of keys are used in public key cryptography: distributed network keys, which are also known as public keys, and individual personal keys, which are also known as private keys. The public key infrastructure (PKI) is the most frequent technique providing key management functions for cryptography in the blockchain. PKI techniques based on blockchain are decentralised, which eliminates the need for a centralised access point or a trusted third party [76]. Furthermore, these methods do not require trustworthiness to be established via nodes or system users to make the public system more visible. Instant Karma PKI, Blockstack, and Certcoin are only a few of the blockchain approaches that have been mentioned in the literature to enable PKI encryption and transaction security on blockchain nodes. Blockchain privacy and security, on the other hand, are only now beginning to be fully addressed. As explained in published research any exposure of the private key owner's identity can lead to the disclosure of additional transactions by that owner using linking techniques [115]. Furthermore, when exposed to certain types of attacks, the anonymity of blockchain users may be compromised [78].

Moreover, as a means of ensuring privacy, Ethereum uses cryptographic hash functions and transactions are secured using cryptographic mechanism-based privacy. However, since Ethereum is a public ledger, all users may access the decentralized ledger. The transaction data is available online but the inclusion of these cryptographic frameworks does not guarantee full privacy. Deanonymization attack is the most well-known privacy attack on Ethereum, in which data from a distributed ledger is deanonymized by tracing and linking features with other databases [28].

Hence, the methods for preserving privacy in blockchain applications constitute an important research issue. Some researchers have sought to improve blockchain privacy through the use of different strategies such as the use of two-level anonymity. Additionally, some have focused on resolving confidentiality issues based on public blockchain transactions to enhance blockchain trustworthiness [31]. Another potential solution is the use of a differential privacy preserving strategy that utilises data perturbation methods for the protection of private data in the blockchain, as alluded to above. Adding noise to the stored distributed ledger records is a feature provided by differential privacy to overcome this issue. Non-trusted users or those who lack a specific role in the network can benefit from the unpredictability noise of differential privacy. It may be possible to only allow query evaluation in the public ledger to analyse any record or previous transaction and add noise to this query evaluation to protect privacy. Also, Ethereum's smart contract gives developers the ability to add differential privacy to their transaction [28]. The flexibility of choosing a suitable way to add noise based on privacy and utility requirements makes the use of differential privacy optimal to overcome the privacy issues in blockchain-based architecture.

### 5.2.1 Privacy preserving mechanism in Blockchain

Blockchain-based IoT systems do not have privacy features embedded in their design. As a result, the private data of users may be uncovered by adversaries using targeted attack strategies. To address this issue, researchers have suggested several strategies to preserve user privacy for different blockchain-based IoT system applications [75].

**Encryption:** Encryption is a technique to preserve users' privacy by scrambling their critical data and sensitive information or details, and coding them in an unreadable format that is nearly impossible to break. Once the programmer reaches a higher understanding of how to appropriately implement this technique, the users' data is then completely secure and thus kept private [122].

**Anonymization:** Anonymization is another technique of privacy-preservation that creates anonymous protocols for users so that they do not appear online with their real personal details; this is supposed to protect their identity and allows them to control the amount of information they communicate [122].

**Differential privacy**: Differential privacy is a useful privacy preservation strategy for maintaining data confidentiality without the risk of leakage. The concept was initially introduced by Dwork who developed a mechanism for protecting database privacy via the addition of noise during query evaluation [23]. As the approach was strengthened, researchers began to apply different differential privacy variations in everyday applications. Researchers continue to develop different differential privacy-based IoT systems which apply data perturbation concepts in real-time and dynamic settings [136]. However, the application of differential privacy in smart home applications still needs research. This chapter investigates the integration of differential privacy based on machine learning with blockchain technology in the smart home scenario.

## 5.3 Differential privacy

Differential privacy is described as a probabilistic mechanism used for the provision of an information-theoretic security guarantee. The following definition is provided by Dwork [51].

In the case of two adjacent datasets D and D' which differ by a single record, M preserves ($\epsilon$;$\delta$)-differential privacy if

$$Pr[M(D) \in S] \leq Pr[M(D') \in S] \times e^{\epsilon} + \delta \tag{5.1}$$

where $\epsilon$ is the privacy budget and $\delta$ is the failure probability. In this equation, $\epsilon$ functions as a privacy budget controlling how much noise is added. In addition, $\delta$ is the sensitivity value typically established based on the dataset. S is the scope query output for query function M.

In addition to the standard definitions, researchers have sought to propose privacy variations to address different privacy requirements [77]. The variants are classified into four data perturbation processes: Laplace, Gaussian, Uniform, and Geometric; all of these support differential privacy. The Gaussian process is used in this work.

One method to improve utility in a privacy budget is to broaden how deferential privacy is defined. Some broader definitions have been offered that provide enhanced utility, including small $\epsilon$ values [77]. Three broader differential privacy definitions which are regularly applied include zero-concentrated deferential privacy [24], advanced composition deferential privacy [52], and Rényi deferential privacy [106]. All three provide an enhanced evaluation of cumulative privacy loss by exploiting the fact that privacy loss random variables are based on an anticipated privacy loss. The cumulative privacy budget gained from the evaluation ties the composition mechanism's greatest privacy loss with all except the $\delta$ failure probability. As a result, there is a reduction in the amount of noise needed and therefore an enhancement utility across multiple compositions. This work applies Rényi differential privacy (RDP).

### 5.3.1 Rényi Differential Privacy

Rényi differential Privacy (RDP) is defined as [106]: a randomised process M which has $\epsilon$-RDP of order $\alpha$; abbreviated as $(\alpha;\epsilon)$-RDP. In adjacent datasets D, D' it holds that

$$D_\alpha(M(D) \parallel M(D')) \leq \epsilon \tag{5.2}$$

RDP bounds single privacy loss moments at a stated time, permitting better accuracy in privacy loss analysis [183]. When M is a $(\alpha;\epsilon)$-RDP process, it then satisfies $(\epsilon + \frac{log1/\delta}{\alpha-1}, \delta)$-DF for all $0 < \delta < 1$.

Through the concept of moment accounting, RDP allows for a better composition result [1]. Several studies [107, 168, 187] have shown that analysing the RDP of subsampled techniques provides a tighter bound on total privacy loss than utilising typical strong composition theorems [64].

## 5.4 Proposed Architecture

### 5.4.1 Privacy preserving Differential Privacy Mechanism

We implement privacy-preserving classification using edge computing and a blockchain scenario. The proposed mechanism trains the machine learning model accurately to suit all IoT smart home data. The model also classifies a given packet to an IoT device in the smart home scenario as shown below

```
dict_labels = {'Pc': 0, 'Temperature sensor': 1, 'LED sensor': 2}
for i in range(y_train.shape[0]):
  y_train[i] = dict_labels[y_train[i]]
  y_train = y_train.astype('float')
```

The aim is to provide a privacy-preserving data aggregation method in the context of smart homes that agrees to provide their data to a cloud server, so that the cloud can learn privately from the data produced by IoT devices inside the home and then deliver these data to an external entity to provide better services for home users.

As Fig 5.1 shows, we consider that a number of edge nodes have private data from the IoT devices in the smart home and collaborate with each other to return the results to the cloud. These edge nodes assist the smart home in sharing their data with the cloud by learning the model and training the data before sending the final result to the cloud. The edge nodes first calculate the gradients based on the current model while attempting to limit privacy leakage. They employ a differential privacy scheme to perturb their data.

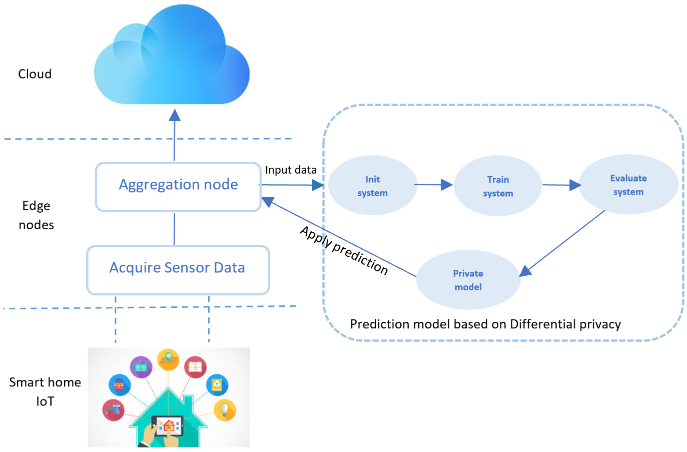


Figure 5.1: Edge node functions for data privacy scheme

The cloud collects the gradients broadcast by the edge nodes and performs the desired scheme to analyse the data. For the proposed model, we consider two different methods to train the model using a machine learning algorithm on the prepared data. First, without considering privacy, we train a one-layer neural network on the data and analyse the accuracy of the proposed scheme. We call this approach a "plain algorithm". Second, we train the same one-layer neural network on our data based on the scenario previously explained. We use stochastic gradient descent (SGD), one of the most popular optimization algorithms [29]. SGD algorithms have received significant attention recently because they are simple and satisfy the same asymptotic guarantees as more computationally intensive learning methods [152]. We call the second algorithm a "private algorithm".

## 5.4.2 Algorithms and dataset

### 5.4.2.1 Plain Algorithm

As previously mentioned, this algorithm does not consider any matters of privacy and the data is handed over in the full to the cloud server through the following algorithm:

```
kfold = KFold(n_splits=10, shuffle=True)
fold_no = 1
for train, test in kfold.split(X, y):
 model = Sequential()
```

```
model.add(Dense(no_classes, activation='softmax'))
 # Compile the model
 model.compile(loss=loss_function,
      optimizer=optimizer,
      metrics=['accuracy']
```

The algorithm specifications are as follows:

- Model: K-fold-one-layer neural network
- Loss function: categorical cross entropy
- Optimizer: adam (adaptive moment estimation)
- Number of epochs (training rounds): 10

#### 5.4.2.2 Private Algorithm

The basic idea of this approach is presented in following algorithm:

```
models.append( tf.keras.Sequential([tf.keras.layers.Dense(3)]))
       optimizers.append( DPGradientDescentGaussianOptimizer(
       l2_norm_clip=l2_norm_clip,
       noise_multiplier=noise_multiplier,
       num_microbatches=num_microbatches,
       learning_rate=learning_rate))
       losses.append(tf.keras.losses.CategoricalCrossentropy(
       from_logits=True, reduction=tf.losses.Reduction.NONE))
models[i].compile(optimizer=optimizers[i],
                  loss=losses[i],
                   metrics=['accuracy'])
```

The scheme called differential private stochastic gradient descent (DP-SGD) modifies the gradients used in SGD, which lies at the core of almost all deep learning algorithms. Models trained with DP-SGD provide demonstrable differential privacy guarantees for their input data. We have made the following two modifications to the SGD algorithm in to accommodate privacy aspects with the data [121]:

- "First, the sensitivity of each gradient needs to be bounded. In other words, we need to limit how much each individual training point sampled in a mini batch can influence gradient computations and the resulting updates applied to model parameters. This is done by clipping each gradient computed on each training point" [128].

- "Random noise is sampled and added to the clipped gradients to make it statistically impossible to know whether or not a particular data point was included in the training dataset by comparing the updates which SGD applies when it operates with or without this particular data point in the training dataset" [128].
- We select the following parameters and specifications in the design of our algorithm:
  - Model: k-fold-one-layer neural network
  - Loss function: categorical cross entropy
  - Optimizer: DP-SGD (differentially private stochastic gradient descent)
  - Number of epochs (training rounds): 10
  - l2-norm-clip: 1.5
  - Noise multiplier: 2

#### 5.4.2.3 Dataset

The experiment was conducted to detect and classify a type of device in a private blockchain in a smart home; the ultimate purpose of this is to prove whether machine learning has the same or better accuracy when using differential privacy and that for the sake of increase the privacy, as alluded to above. One way of doing this is to observe how machine learning techniques on captured packets (stored in files such as pcap files) are applied to distinguish different devices in the network. The dataset was produced by generating a pcap file using Wireshark to capture the network packets in our private network. Our synthetic dataset consists of n = 11,000 samples. Using Tshark, we then filtered the captured packets and extracted the headers of each packet. Then, the dataset was created and processed using the Python script. We selected our dataset based on network traffic generated by our private Ethereum network, thus providing accurate representations of the devices we use in the experiment.

#### 5.4.2.4 Threat model

Our goal is to collect smart home data from the edge nodes and analyse the efficiency of our proposed scheme using different threat models. Since all data stored in blocks will be available to all blockchain users, we assume that the adversary in our model may have full access to the data. We focus on side channel attacks where adversaries use machine learning algorithms to infer information on smart home IoT devices by monitoring the incoming/outgoing network traffic from/to smart homes. We would like to emphasize that traffic patterns extracted from IoT data may enable adversaries to correlate their inside information on some residents, thus giving adversaries prior information to assist in launching an inference attack on the system. As a result, the adversaries can create a profile on smart home residents and launch subsequent sophisticated attacks such as the linkage attack.

In addition, we also consider other threats associated with the malicious user where adversaries steal identity information such as geographical data about the edge node and allowing the adversary to steal specific tasks that the edge node executes. Also, another assumption is that the adversary can legally communicate with the edge node and as a result, leak geographical information. Attackers can easily measure the communication time and estimate the physical distance from measuring/ comparing latency.

We assume that the cloud server deployed is secure, as it is one element of the architecture described in Chapter 4 in Fig.4.1. The classification model is trained on different edge nodes with a tailored machine learning algorithm to classify a given packet to one of the IoT devices in the smart home.

## 5.5 Evaluation and Analysis

### 5.5.1 Privacy analysis

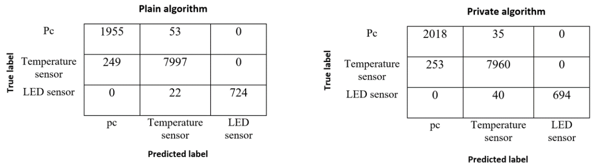
In our proposed model, we assume that all participants have a verified identity that is managed and issued by the access control scheme in the smart contract in a private blockchain. Therefore, identity privacy in our framework is out of the scope of our work. We only consider privacy leakage from data when a learning process runs.

We present a security analysis on the proposed differential privacy-based blockchain system, which is associated with the pre-defined threat environment discussed in the threat model section. Based on the threat assumption, adversaries have full access to all data stored in the blocks. In our model, for the first type of threat, without adding noise, adversaries can easily obtain the real identities and behaviour of users by mining information or launching a linkage attack. Fortunately, our model uses a differential privacy protection method (Gaussian Distribution Mechanism) to add noise into the real data, such that a distortion is effected to protect the target set. We observe that using the Gaussian mechanism it is possible to successfully screen and classify the IoT devices while ensuring the privacy of all data.

In data mining-based attacks, from the adversary's perspective, adding the noise can escalate the complexity of feature extraction and information retrieval. Moreover, added noise is also essential to defend users' and IoT devices' identities to prevent the second type of threat, as matching data rarely occurs between blockchain data and other supportive databases for the processed data. Thus, our model can efficiently improve the privacy-preserving capability.

### 5.5.2 Experiment results

We compare the performance of both machine learning techniques provided in the previous section using the confusion matrix. The alternative outputs of a classification, which in our case are '0' for the PC, '1' for the temperature sensor, and '2' for the LED sensor, are compared to the actual values of the class feature already available in the evaluation (testing) dataset, as illustrated in Fig 5.2.



Plain algorithm

| True label | pc | Temperature sensor | LED sensor |
|---|---|---|---|
| Pc | 1955 | 53 | 0 |
| Temperature sensor | 249 | 7997 | 0 |
| LED sensor | 0 | 22 | 724 |

Predicted label

Private algorithm

| True label | pc | Temperature sensor | LED sensor |
|---|---|---|---|
| Pc | 2018 | 35 | 0 |
| Temperature sensor | 253 | 7960 | 0 |
| LED sensor | 0 | 40 | 694 |

Predicted label

Figure 5.2: The confusion matrix of device classification

There are four parameters presented in the confusion matrix, True Positive (TP), where the classifier has correctly measured the number of packets that are correctly classified to a device type, True Negative (TN), similar to TP but the value of the class feature is negative, False Positive (FP), where the classifier measures the number of packets that are incorrectly classified as a device type and False Negative (FN), which measures the number of packets that are incorrectly not classified as a device type. One metric is created by combining the TP, TN, FP, and FN values, namely accuracy which we can use to evaluate the classifiers. Accuracy represents the probability that a record is correctly identified as one of the device types.

The accuracy (overall success rate) is calculated using the following equation:

$$OSR = (TN + TP)/(TP + FP + TN + FN) \tag{5.3}$$

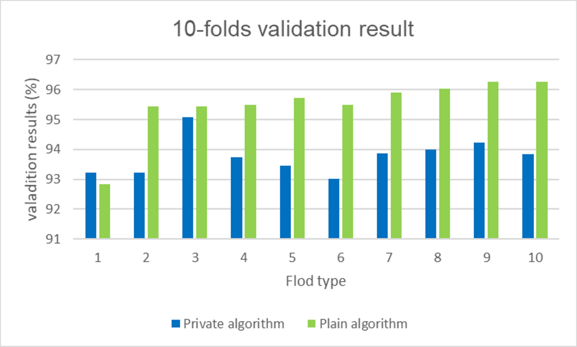


Figure 5.3: 10-fold validation results

For the classification stage, we use Python in the Google Colab environment to apply a well-known machine learning algorithm. We illustrate the approach using k-fold cross-validation on the neural network model to ascertain the efficiency of our proposed scheme.

Fig 5.3 shows the accuracy of the model before (plain algorithm ) and after (private algorithm) adding the noise.

Table 5.1: Calculated accuracy

| Classifier | Accuracy |
|---|---|
| Plain algorithm | 0.95 |
| Private algorithm | 0.93 |

As shown in Table 5.1, the plain model has an average accuracy close to 0.95 (95%) while for the private model, accuracy is close to 0.93 (93%).

Our experiment shows that the accuracy of our private model is very close to that of the plain one when the privacy budget is 0.7 because the private method with noise disturbance is relatively small. Therefore, the accuracy of this classification method is close to that of the plain classification method. It is shown in the experiment that the private model has the same accuracy as the plain model in classifying the device type. Thus, our results demonstrate the feasibility of differential privacy guarantees without significant loss in terms of accuracy; edge nodes aggregate noisy data to the cloud while preserving smart home privacy and providing accurate data for further analysis.

Nonetheless, there is a trade-off between accuracy and privacy that directly links to adding noise to the scheme. To increase the level of privacy, we increase the amount of noise. However, on the other hand, this may result in a loss of data accuracy. Therefore, efficient measurements are required to achieve the best result. However, it is outside of the scope of the present chapter and we leave it for our next chapter, where we conduct further analysis to measure the differential privacy guarantee to reach improved privacy protection without any loss in accuracy.

## 5.6 Summary

In this chapter, we have extend our work published in [127] and expand the functional capabilities of our architecture by adding differential privacy as a scheme to preserve the privacy of users.

The basic characteristics of blockchain technology are a trustless environment, immutability, and transparency, all of which come at the expense of data privacy. From a privacy-preserving perspective, information recorded on blockchain may raise privacy concern. The blockchain itself cannot solve the privacy issue caused by data sharing [173]. Thus, a formal, mathematical model for data privacy is required to address the issue of privacy. There is a need to build a design mechanism for blockchain usage that does not compromise data privacy while gaining the benefits of the technology.

We integrate differential privacy based on machine learning with the use of our blockchain model. The model classifies a given packet to an IoT device in the smart home scenario. The dataset was created using Wireshark to record network packets in our private Ethereum network and create a pcap file. The experiment shows that the differential privacy model has the same accuracy as the plain model, and it guarantees privacy without sacrificing accuracy.

The analysis presented in this chapter is useful in addressing the thesis's third research question. Also, this chapter is the foundation for the following chapter, where we test our proposed model with Rényi differential privacy on a wider scale with different classifier algorithms as a proof of concept. Moreover, we conduct further research to achieve better privacy guarantee to strongly protect smart home data with better accuracy.

CHAPTER 6

# DIFFERENTIAL PRIVACY MEASURES ON DIFFERENT DATASETS

The chapter is organized as follows: Section 6.1 introduces the chapter. Section 6.2 describes the threat model. In Section 6.3, we propose our differential privacy model. In Section 6.4, we discuss the experiment setup. The experiment results are discussed in Section 6.5. Finally, Section 6.6 summarizes the chapter.

## 6.1 Introduction

The ever-increasing gathering and transfer of personal information in smart home networks (along with the possible communication of this data to other wired or wireless networks) has implications for user security and privacy. In turn, these implications must be addressed if the full range of affordances and benefits of smart home networks are to be exploited effectively [130].

The current application of blockchain technology in IoT and smart cities is of interest to both academia and industry. Blockchain is a disruptive technology emerging from the digital currency domain which is becoming more widespread in various other areas. Bitcoin was the first digital currency to successfully apply blockchain technology to support a tamper-free transaction recording ledger. Ethereum developed the smart contract via the Ethereum Virtual Machine (EVM) on its blockchain. Smart contracts enable people to use a trusted computing machine in the blockchain, thus facilitating the future use and success of decentralised apps (DApps) [177].

Blockchain is increasingly being identified as the pathway to addressing concerns about IoT security, reliability, trustworthiness, and scalability [14]. Blockchain adoption in smart homes lessens security worries around authentication, authorisation, confidentiality, and single point of attack. Blockchain technology is built upon cryptography that supports digital ledger decentralisation. Rather than using centralised networks, it utilises a distributed database that retains a chain of blocks. The blocks in the blockchain are connected by keeping the previous block's hash to safeguard the blocks from interference [75].

However, a lack of user privacy linked to blockchain's widespread adoption and implementation remains a major concern [48] [76]. Data confidentiality has subsequently emerged as an issue of primary importance as smart-home-generated data contain sensitive content including user health information and location details. The main concerns about the integrity of blockchain are around attacks related to user privacy such as linking attacks [59]. Such attacks utilise accessible data recorded in blocks to gain access to private information by tying the information to alternative datasets or relevant background knowledge. Attackers may have a greater chance to work out how to target smart home privacy data. Launching such an attack may include the use of data mining algorithms, with the process made easier when it involves unencrypted raw data [48]. Thus, there is a developing sense of urgency for blockchain-based smart home framework designs to include mechanisms for the preservation of user privacy. Privacy protections of smart home data aim to permit data usage without disclosure as well as to reduce data loss when providing published data to service providers.

To date, anonymization and its strategic derivatives have only been identified in the literature in relation to the preservation of user privacy in blockchain [76]. However, several studies have demonstrated that anonymization does not offer complete privacy because anonymized data may be blended with similar datasets to disclose personal information. To satisfactorily address this problem, the combination of differential privacy and modern blockchain technology may offer a sustainable solution due to its dynamism and robust theoretical base [76]. Differential privacy can be applied in blockchain to access private databases through queries that achieve data aggregation, as well as to receive user data with statistical differences from sources while maintaining the user's chosen privacy levels.

This chapter presents a secure privacy-preserving layered architecture for smart home based-blockchain. To maintain security among IoT devices, users and the cloud, an access control smart contracts in private Ethereum blockchain is designed as we presented in chapter 4. Furthermore, to achieve privacy preservation, the architecture employs the differential privacy machine learning algorithm to send a private smart home data to the cloud.

In the context of smart homes that agree to submit their data to a cloud server, our proposal intends to provide a privacy-preserving data aggregation mechanism, so this cloud can learn privately from data generated by IoT devices in people's homes and then transfer that information to an external entity to improve services for home users.

This chapter is extended further to prove the advantages of using differential privacy based on our scheme we presented in Chapter 5 of the thesis. In Chapter 5 we used our own dataset for the experimental part. In Chapter 6 we use differential privacy techniques with three different IoT datasets consisting large amounts of data and different machine learning algorithms. Also, we investigate various metrics over and above accuracy in this chapter. Using differential privacy in our small dataset produced promising results. We also use three benchmarked datasets and demonstrate how they perform under different conditions to validate our proposed scheme.

The main objective of the model in this chapter is to protect privacy with complete accuracy in relation to the model's predictions when aggregating the data from traffic analysis attacks, linking and mining attacks by adding Gaussian noise. The implementation of our model ensures the accuracy of the calculation is reliable and the model utility is elevated. The accuracy requires that noisy data will not influence the correct output of any data analysis. Furthermore, correct data are retrievable for further analysis and investigation while implementing our model.

Undoubtedly, between utility and privacy, there is a trade-off. As a result, while increasing the privacy budget improves model utility, it also improves inference attack success rates. Therefore, we are seeking a set of privacy budget $\epsilon$ values that achieve a balance between utility and privacy, as well as a way to evaluate the exact privacy leakage that occurs when an inference attack is implemented. We study neural network models on three different datasets.

The main contributions are:

1. This chapter presents a secure privacy-preserving layer architecture for smart home-based blockchain. To maintain security among IoT devices, users and the cloud, an access control smart contract in private Ethereum blockchain is designed to authenticate access to IoT smart home devices. Furthermore, to achieve privacy

preservation, the architecture employs a differential privacy machine learning algorithm to send private smart home data to the cloud.

2. The proposed model's effectiveness is defined in terms of accuracy, utility and privacy leakage. Experiments are performed using three publicly available datasets, the UNSW-NB15 dataset, NSL-KDD dataset and ToN-IoT.

3. We compare the performance on two metrics, accuracy and utility, of our private algorithm using three IoT-based datasets which have recently been used in recently related smart-home-based blockchain frameworks.

4. We determine how much privacy leakage our proposed algorithm will allow in adversarial scenarios such as inference attacks using all three datasets. Through the evaluation of the private model, we choose different values of privacy budget $\epsilon$ quantifying privacy leakage, attack accuracy and F1-score.

The remainder of the chapter is structured as follows. In Section 2, we present a review of the related literature. We describe the threat model in Section 3. In Section 4, we detail the proposed layered architecture, algorithms and datasets used. The experiment's setups are described in Section 5 and the experiment results are described in Section 6. Finally, in Section 7, we conclude our work and discuss the potential future scope for additional work.

## 6.2 Threat model

The threat assumptions are indicated in this chapter through an elaboration of what the cloud is required to access from the data in the smart home. First, the blockchain platform is a trusted party because its features preserve the integrity and availability of data but may lack confidentiality. Smart contracts are tamper-free and will function as specified. After the deployment, the code can be seen and checked by anyone. Furthermore, submitted and stored contract data may be read directly by any parties with access to the blockchain.

As a threat, consideration is given to data consumers such as service providers who are interested in aggregate statistics generated from smart home sensors. They try to infer individual data to achieve added benefits such as improving their services via arbitrary inference methods.

This work examines membership inference attacks specifically to demonstrate the usefulness of its metrics for the evaluation of privacy leakage.

A membership inference attack aims to deduce that a given record exists within the training set. Such attacks can reveal information from the training data as highly sensitive. Membership inference attacks may be fully black-box whereby the attacker has query access only to the targeted model [148] or they may assume full white-box access to the targeted model by the hacker, in addition to auxiliary information [179].

Shokri et al. [148] first proposed membership inference attacks on machine learning by considering attackers with the capability to deploy black-box methods to query the target model to gain confidence scores related to the queried input. Attackers attempted to manipulate the confidence score to establish the presence of the query input within the training data. The attack method initially requires the labelled dataset on training shadow models to then be produced by black-box queries or via assumptions regarding the underlying training set distribution. Attackers then develop an attack model utilising shadow models to determine if an input record is present in the shadow training set. Lastly, attackers then conduct API calls to obtain confidence scores for the stated to target model input record and deduce if the input is part of the training set. Hence, inference models distinguish between the target model's training set input predictions and the predictions that are not trained on. This is based on the idea that the confidence score in the target model will be higher in incidences of training compared to arbitrary instances outside the training set. The cause of this may be the generalisation gap prominent in models which overfit training data.

The objective of differential privacy is to obscure the existence or absence of dataset records. The objective of membership inference attack, conversely, is to reveal the existence or absence of dataset records. Intuitively, the two ideas are in opposition, with some researchers pointing to the direct relation between differential privacy and membership inference attacks [183].

To measure privacy leakage in differential private machine learning implementations, this work evaluates the extent to which an attacker can infer information from a model. It is to be noted that we measured privacy leakage via the use of membership inference.

## 6.3 Differential privacy model

In this section, we present the layered architecture with differential privacy. The architecture demonstrates a layered access scheme to secure sensitive data as shown in

Fig. 6.1.

Each layer obtains input queries from the layer above it and invokes appropriate access policy based on the role of the user. Our main objective is to demonstrate the functions associated with Layer-3, the differential privacy layer. Our previous work describes in detail the functions of Layer-1 and Layer-2 [127]. However, we briefly present the use of these layers as follows:

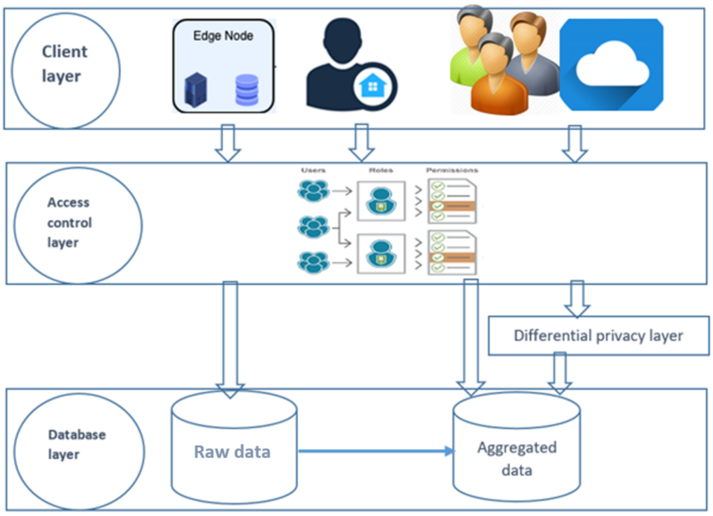


Figure 6.1: Use of Differential Privacy in a layered architecture

- Client Layer: The client layer accepts queries from different users and sends them to the access control layer along with their credentials.

- Access Control Layer: The access control blockchain layer is responsible for granting data access requests. After the client layer executes a smart contract, the transaction is initiated. The user is granted the necessary access credentials to perform the transaction based on the inputs submitted to the smart contract. The smart contract is executed on all nodes attempting to access the data tables. Following this, the block is disseminated over the blockchain network. All network nodes validate the block, agree on the consensus algorithm, and add the block to the blockchain. Any user can not modify the smart contract code, and the logic is always run once a user tries to access data. To allow secure access, the access control system uses blockchain technology and smart contracts, returning the key required to conduct the queries. The fundamental benefit of smart contracts is that any complicated access permission logic may be easily implemented.

- Differential Privacy Layer: To provide additional protection to sensitive information, the differential privacy layer uses differential privacy approaches. Differential privacy is a perturbation-based computing technique. To maintain privacy, it is implemented by introducing noise to the data. By doing this, only the owners of the data have access to the real data. The differential privacy layer is invoked after the access control layer compels all users to provide all of their queries. Actual results are adjusted based on the user type to protect individual privacy and operational privacy. This layer is responsible for moving the data from edge nodes to the cloud server to make it available for further use and analysis.

As seen in Chapter 4, attribute-based access control is implemented on the Ethereum blockchain with the user's roles stored in the access control smart contract. Users can send queries to the system using a public address. Upon receiving a user's public address, the client layer calls the smart contract. If access is allowed, the smart contract returns the user's authorisation; otherwise, access is refused. The client layer request is either transmitted directly to the data repository (for the administrator) or passed through the differential privacy module if the access control layer so authorises (service accessor). Users who are not legitimate are, of course, refused access.

### 6.3.1 Differential Privacy mechanism

In this scenario, we suppose that there are many smart home edge nodes and that there is one cloud server. The edge nodes have access to the data of IoT devices inside the smart home. The cloud server can be accessed by service accessors based on their credential. The service accessors are interested in having access to the data of IoT edge nodes to undertake analysis by implementing machine learning algorithms. For example, service accessors can learn how customers interact with their IoT devices and appliances using specific machine learning algorithms and also learn about customer usage insights, which guides the development of their future products and services. Also, health providers can apply machine learning algorithms to user behavioural patterns learned from medical sensors to detect anomalous behaviour which could be a sign of user health or safety risk.

Suppose that each edge node has gathered hundreds of IoT packets that have been passed through it and between different IoT devices inside the smart home. Edge nodes have access to the label of those packets, which in our scenario shows whether the packet is malicious or not. Assuming that the number of packets gathered by each edge node is not large enough to run a machine learning algorithm on them, the cloud server will gather all the packets and make it available for further analysis which compromises the privacy of smart home users. Therefore, we need to consider a way to aggregate the data to the cloud while ensuring privacy. We propose a private algorithm which adds noise to the data and results in sending noisy data to the cloud without compromising privacy.

Our algorithm trains the model on private data using differential privacy. Learning with differential privacy gives verifiable privacy assurances and decreases the danger of disclosing sensitive training data. A model trained with differential privacy could, intuitively, be unaffected by any single or small group of training samples in its data set. The edge nodes actively participate, applying the machine learning algorithm in a differential private setting. There are a number of alternative approaches that may be considered. First, edge nodes can add noise (Laplace or Gaussian noise) and make their data noisy. The noisy data can be sent to the cloud server. The variance of the noise is related to the level of privacy. The more noise is added, the higher the level of privacy is provided. On the other hand, more noise results in lower utility.

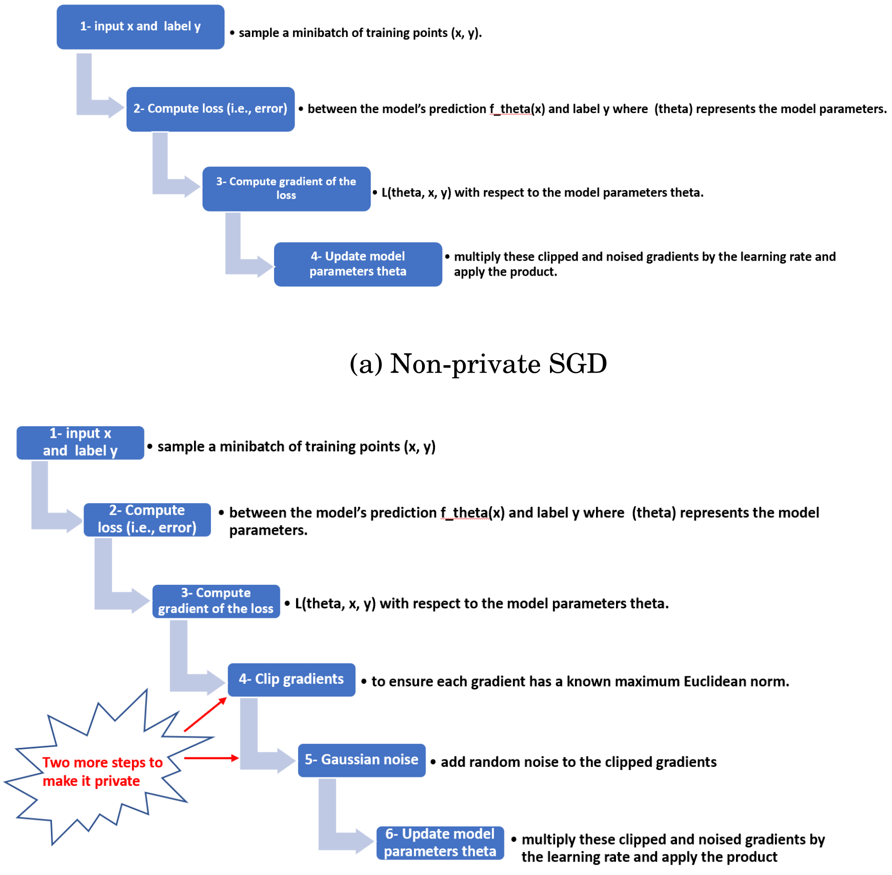
Second, the edge nodes can be involved in the computation phases of running the algorithm in the cloud. They can do the computation element of their own data separately (if applicable) and return the results to the cloud without handing over their data to the cloud. They can also add noise to the outcome of computations to enact the privacy of their data. The challenge is to find the level of noise that should be added without compromising privacy while improving utility. In our scenario, we investigate different noise levels to attain better privacy with acceptable utility.

### 6.3.2 Privacy preserving Differential Privacy Model

This section demonstrates how we apply differential privacy in machine learning systems, with an emphasis on the privacy budget variations that have been made to achieve acceptable utility. The choice of privacy budget ($\epsilon$) is critical to the effective privacy given by differential privacy mechanisms. While higher privacy budgets provide better utility, a lower privacy budget provides better privacy.

On the prepared data, we explore two distinct methods for training the machine learning algorithm. Non-private algorithms without considering privacy and private algorithms are investigated. Fig. 6.2 shows the two steps that distinguish the two algorithms.

For both algorithms, we train a three-layer neural network on the data and analyze the method outputs including the confusion matrix based on the scenario previously


1- input x and label y
• sample a minibatch of training points (x, y).
2- Compute loss (i.e., error)
• between the model's prediction f_theta(x) and label y where (theta) represents the model parameters.
3- Compute gradient of the loss
• L(theta, x, y) with respect to the model parameters theta.
4- Update model parameters theta
• multiply these clipped and noised gradients by the learning rate and apply the product.
1- input x and label y
• sample a minibatch of training points (x, y)
2- Compute loss (i.e., error)
• between the model's prediction f_theta(x) and label y where (theta) represents the model parameters.
3- Compute gradient of the loss
• L(theta, x, y) with respect to the model parameters theta.
4- Clip gradients
• to ensure each gradient has a known maximum Euclidean norm.
Two more steps to make it private
5- Gaussian noise
• add random noise to the clipped gradients
6- Update model parameters theta
• multiply these clipped and noised gradients by the learning rate and apply the product

(a) Non-private SGD

(b) Private SGD

Figure 6.2: Non-private SGD and private SGD algorithms' steps

explained. For this, the activation function is used to determine the output of each node on the neural network. ReLU is the most used activation function while working with fully connected deep neural networks [13]. The function returns 0 if it receives any negative input, but for any positive value x, it returns that value. The vanishing gradient problem is solved using ReLU, which allows the model to learn faster and perform better.

The optimizer uses the algorithm to change the attributes of neural networks such as weight and learning rate to reduce the losses and, hence, provide more accurate results. We use stochastic gradient descent (SGD) as one of the most common and successful optimisation methods [84]. SGD is an iterative procedure. Each iteration takes a random sample of data from the training set (this is where stochasticity comes from). The error between the model's prediction and the training labels is then calculated. The loss is then discriminated in relation to the model's parameters. These derivatives (or gradients) indicate how each parameter should be updated to bring the model closer to correctly predicting the label. Iteratively recomputing gradients and using them to update the model's parameters is referred to as the descent [65].

### 6.3.3 Non-private Algorithm

As previously mentioned, this algorithm has no regard for privacy and the data is sent directly to the cloud server. The algorithm specification is as follows

- Model: three-layer neural network
- Loss function: categorical cross entropy
- Optimizer: Stochastic Gradient Descent
- Number of epochs (training rounds): 10

```
model = tf keras.Sequential([
  tf.keras.layers.Dense(64, activation='relu'),
  tf.keras.layers.Dense(32, activation='relu'),
  tf.keras.layers.Dense(2, activation='softmax' ])
model.compile(optimizer-'SGD',
  loss-tf.keras.losses.SparseCategoricalCrossentropy(fromlogits-True)
  ,metrics=-['accuracy'])
model.fit(X train, y train, epochs=10)
```

### 6.3.4 Private Algorithm

The concept underlying this method, known as differential private stochastic gradient descent (DP-SGD), is to adjust the gradients used in SGD, which is at the core of nearly all deep learning algorithms by injecting calibrated noise into the training gradients. As shown in Fig. 6.3, the model trained with DP-SGD provides verifiable differential privacy guarantees for their input data. Using the Gaussian mechanism, the framework can successfully screen and classify the normal packets while ensuring and guaranteeing the privacy of all data.

The major process for training a model with privacy-specific hyperparameters is summarized as the follows:

1. Compute the gradient for a random subset of examples.

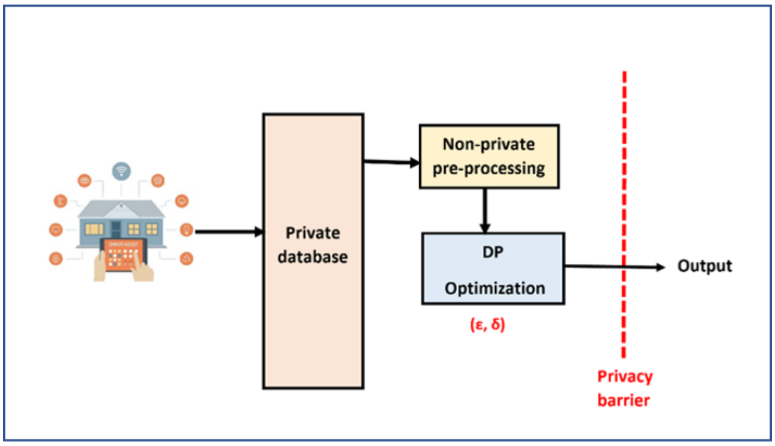


Figure 6.3: Learning using DP optimizer

2. Clip the L2 norm of each gradient.
3. Compute the average gradients.
4. Add some noise to protect privacy.
5. Take a step in the opposite direction of this average noisy gradient.
6. Obtain model output.

"DP-SGD has three privacy-specific hyperparameters and one existing hypermeter that must be tuned to obtain fine-grain results.

- l2_norm_clip (float) - The maximum Euclidean (L2) norm of each gradient that is applied to update model parameters. The optimizer's sensitivity to individual training points is limited by this hyperparameter.
- Noise_multiplier (float)- The amount of noise sampled and added to gradients during training. Generally, more noise results in better privacy (frequently, but not always, at the cost of reduced utility).
- Microbatches (int) - Each batch of data is split in smaller units called microbatches. By default, each microbatch should contain a single training example. This allows us to clip gradients on a per-example basis rather than after they have been averaged across the minibatch. This, in turn, decreases the (negative) effect of clipping on a signal found in the gradient and typically maximizes utility. However, increasing the size of microbatches to include more than one training example can

reduce computational cost. The average gradient across these multiple training examples is then clipped. The total number of examples consumed in a batch, i.e., one step of gradient descent, remains the same. The number of microbatches should evenly divide the batch size.

- Learning_rate (float) - This hyper parameter already exists in vanilla SGD. Each update becomes more important as the learning rate increases. A low learning rate may assist the training procedure in converging if the updates are noisy (for example, when the additive noise is considerable compared to the clipping threshold)" [128].

Table 6.1 shows the parameters and specifications of the algorithm shown below:

Table 6.1: Architecture of ANN AutoEncoder using DP-SGD

| | |
|---|---|
| Number of layers | 3 |
| Activation function in the hidden layers | ReLU |
| Optimizer | DPGradientDescentGaussian |
| Number of epochs | 10 |
| Number of micro batches | 100 |
| Number of sample data for each edge node | 100 |
| l2_norm_clip | 1.5 |
| noise multiplier | 1.5, 2.5 |

```
model = tf.keras. Sequential([
   tf.keras.layers.Dense(64, activation='relu'),
   tf.keras.layers.Dense(32, activation='relu').
   tf.keras.layers.Dense(2, activation-'softmax!)])

optimizer = DPGradientDescentGaussianOptimizer(
            12 norm clip=12 norm clip,
            noise multiplier=noise_multiplie,
            num microbatches=num_microbatches,
            learning rate=learning_rate)
```

```
  loss = tf.keras.losses. CategoricalCrossentropy(
       from logits-True, reduction tf.losses.Reduction.NONE)

model compile(optimizer=optimizer, loss=loss, metrics= ['accuracy'])
model.fit(X train, y train,
          epochs=epochs,
          validation data=(X test, y test),
          batch size-batch size)
```

### 6.3.5 Selection and description of the smart home dataset

To evaluate and test our proposed model, we used three different IoT datasets.

Firstly, we use the UNSW-NB15 dataset, which is a new generation of Internet of Things (IoT) and Industrial IoT (IIoT) datasets to evaluate the fidelity and efficiency of different cybersecurity applications based on artificial intelligence (AI). It was created by UNSW Canberra's Cyber Range Lab, which is part of the Australian Center of Cyber Security. We chose UNSW-NB 15 since it is one of the newest datasets and in widespread usage today, giving realistic representations of both normal (non-malicious) network traffic and a variety of botnet network attacks [89].

Secondly, NSL-KDD is an effective benchmark dataset to help researchers compare different intrusion detection methods. This dataset is an improvement over KDD'99 data set, from which duplicate instances were removed to remove biased classification results. This dataset contains a standard set of data to be audited, which includes a wide variety of intrusions and has been the most widely used dataset for the evaluation of anomaly detection methods [87].

Thirdly, the ToN-IoT dataset was created by UNSW Canberra Cyber IoT Lab, School of Engineering and Information Technology (SEIT), UNSW Canberra @ The Australian Defense Force Academy (ADFA) from a practical and large-scale network. A variety of normal and cyber-attack events from IoT networks were compiled in parallel processing for the dataset [92]. Prior to analyzing data and training ML algorithms on the datasets, we processed them and prepared them for the application of the algorithms. We performed the following steps as data cleaning and preprocessing:

1. Drop the unnecessary columns of data.

    - Drop rows with missing values.
    - Drop columns with text explanations.

  - Drop duplicate rows.

2. Separate labels and data.
   - Split X and y (label - actual values) for test dataset.

3. Normalize data.
   - Transform or convert the dataset into a normal distribution.
   - Split data frame into training and testing data.

## 6.4 Experimental Setup

This work uses a fully connected deep neural network and creates a classification model to train and then build multiple classification models which can classify attacks based on the type of network traffic versus the normal type of network traffic. We used 3 layers for the autoencoder and DP-SGD (differential private stochastic gradient descent) as an optimizer function.

We evaluated the neural network's differential private algorithms using gradient perturbation. We considered Rényi differential privacy (RDP) [106] with a different privacy budget and compared their accuracy, utility, and privacy leakage. We evaluated the models on three main metrics:

- Firstly: accuracy, the model's accuracy on the test set in comparison to the non-private baseline.
- Secondly: model utility based on accuracy loss as defined by [84]. This is determined by the differences between the accuracy of the non-private model and the private model.
- Thirdly: privacy leakage is defined by [179] the calculation of the differences between the adversary's true and false positive rates for membership inference attacks. We also use the attack accuracy and F1-score to quantify the model's vulnerability.

We use the confusion matrix [89] to compare the performance of both ML algorithms provided in the preceding section to evaluate accuracy. As shown in Table 6.2, the two possible outputs either "1" (attack identified) or "0" (normal network traffic), are already in the evaluation (test) dataset when we check the classifications against actual values.

The metric accuracy is created by combining the TP, TN, FP, FN values, which we can use to evaluate the classifiers.

Accuracy refers to the probability of a record being correctly identified as an attack or normal traffic. The calculation of accuracy (overall success rate) is based on Equation 5.3.

Table 6.2: Classification outcomes

| | Actual negative | Actual positive |
|---|---|---|
| Predicted Negative | TN | FP |
| Predicted positive | FN | TP |

To clearly represent the model utility, the accuracy loss of non-private and private models is normalised with regard to the accuracy loss of private models. An accuracy loss value of 1 indicates that the model loses 100% of its accuracy and hence has no benefit, whereas a value of 0 indicates that the model reaches the same accuracy as the non-private baseline.

An inference attack is evaluated using a 20,000-record set, 10,000 of which come from the training set and another ten thousand come from the testing set[84]. The records in the training set are referred to as members, while the records in the other sets are referred to as non-members. The attacker is unaware of these labels. The attacker's goal is to determine if a particular input record belongs in the training set. The privacy leakage metric is computed by subtracting the inference attack's true positive rate (TPR) from its false positive rate (FPR). As a result, the privacy leakage metric is always in the range of 0 to 1, with 0 signifying no leakage.

The attacker accuracy reports the proportion of instances properly predicted to be members of the target model's training dataset. The F1-score is a statistic that combines the precision and recall measures into a single value. In our case, precision refers to the ratio of proper anticipation of input being members of the target model's training dataset, whereas recall refers to the proportion of input projected to be members of the target model's training dataset that are actually members.

### 6.4.1 Model Scenario

In this scenario, we suppose that there are numerous smart home edge nodes and there is one cloud server. The edge nodes have access to the data of IoT devices inside the smart home. The cloud server can be accessed by service accessors based on their credentials. The service accessors are interested in having access to the data of IoT edge nodes to undertake analysis by implementing machine learning algorithms. For example, service accessors can learn, using specific machine learning algorithms, how customers interact with their IoT devices and appliances and can also learn about customer usage insights to guide the development of their future products and services. Also, health providers can apply machine learning algorithms on user behavioural patterns learned from medical sensors to detect anomalous behaviour which could be a sign of user health or a safety risk.

Suppose that each edge node has gathered hundreds of IoT packets that have been passed through it and between different IoT devices inside the smart home. Edge nodes have access to the label of these packets, which in our scenario shows whether the packet is malicious or not. Assuming that the number of packets gathered by each edge node is not large enough to run a machine learning algorithm on them, the cloud server will gather all the packets and make them available for further analysis, which compromises the privacy of smart home users. Therefore, we need to consider a way to aggregate the data to the cloud while ensuring privacy. We propose a private algorithm which adds noise to the data and results in sending noisy data to the cloud without compromising the privacy.

To reiterate, our algorithm trains the model on private data using differential privacy. Learning with differential privacy gives verifiable privacy assurances and decreases the danger of disclosing sensitive training data. A model trained with differential privacy could, intuitively, be unaffected by any single or small group of training samples in its data set. The edge nodes actively participate in applying the machine learning algorithm in a differential private setting. A number of alternative approaches may be considered. First, edge nodes can add noise (Laplace or Gaussian noise) and make the data noisy. The noisy data can be sent to the cloud server. The variance of the noise is related to the level of privacy. The more noise is added, the higher the level of privacy that is provided. On the other hand, more noise results in lower utility.

Second, the edge nodes can be involved in the computation phases of the algorithm running in the cloud. They can do the computation part of their own data separately (if applicable) and return the results to the cloud without handing over their data to the cloud. They can also add noise to the outcome of computations to ensure the privacy of their data. The challenge is to find the level of noise that to add without compromising privacy along with improving utility. In our scenario, we investigate different noise levels to attain better privacy with acceptable utility.

### 6.4.2 Training membership attack model

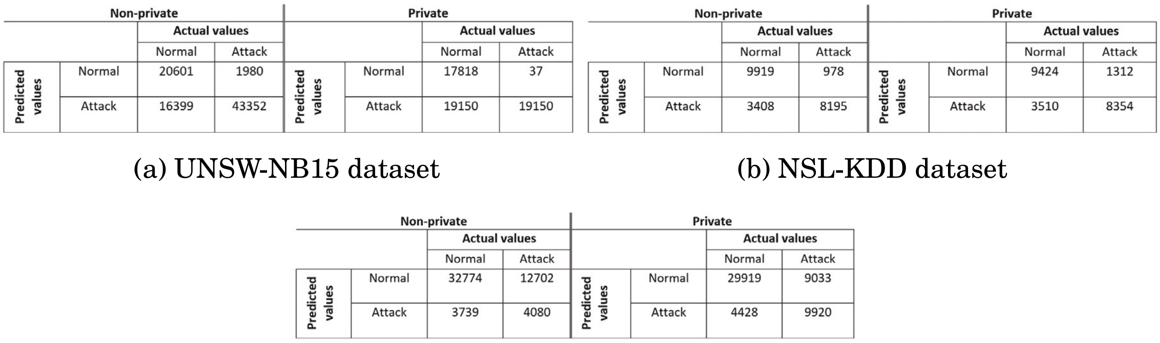
In our attack framework instantiation, we use five shadow models with the same model architecture as the target model. The inference model is made up of two 64-layer hidden layers in a neural network. This setting is consistent with the original work[148].

1. Compute the prediction vector **y** = fI shadow(**x**) for every (**x**, y) ∈ Dtrain shadow i.
2. Set Dtrain attack, to the attack training set and add the record (y,**y**, in).
3. Let Dtest shadow I be a set of records disjoint from the training set of the ith shadow model.
4. Then, ∀($\mathbf{x}, y$)∈ Dtest shadow i, prediction vector **y**= fI shadow(**x**) is computed and the record (y,**y**, out) is added to the attack training set Dtrain attack.
5. Finally, divide the Dtrain attack into ctarget partitions, each with its own class label. Train a separate model for each label y that predicts whether **x** is in or out of the membership set based on **y** [148].

## 6.5 Experimental Results

### 6.5.1 Model Accuracy

The confusion matrices of our classification algorithms are listed in Fig. 6.4 for the three different datasets: the UNSW-NB15 dataset, NSL-KDD and ToN-IoT.



| Non-private | | | | Private | | | |
|---|---|---|---|---|---|---|---|
| | | Actual values | | | | Actual values | |
| | | Normal | Attack | | | Normal | Attack |
| Predicted values | Normal | 20601 | 1980 | Predicted values | Normal | 17818 | 37 |
| | Attack | 16399 | 43352 | | Attack | 19150 | 19150 |

(a) UNSW-NB15 dataset

| Non-private | | | | Private | | | |
|---|---|---|---|---|---|---|---|
| | | Actual values | | | | Actual values | |
| | | Normal | Attack | | | Normal | Attack |
| Predicted values | Normal | 9919 | 978 | Predicted values | Normal | 9424 | 1312 |
| | Attack | 3408 | 8195 | | Attack | 3510 | 8354 |

(b) NSL-KDD dataset

| Non-private | | | | Private | | | |
|---|---|---|---|---|---|---|---|
| | | Actual values | | | | Actual values | |
| | | Normal | Attack | | | Normal | Attack |
| Predicted values | Normal | 32774 | 12702 | Predicted values | Normal | 29919 | 9033 |
| | Attack | 3739 | 4080 | | Attack | 4428 | 9920 |

(c) ToN-IoT dataset

Figure 6.4: The confusion matrices of our classification algorithms

To calculate the accuracy from the confusion Matrix and equation 5.3, the accuracy of the non-private algorithm for UNSW-NB15 dataset will be:

Accuracy (OSR)= (TN+TP)/(TP +FP+TN+FN)

=(20601+43352)/(43352+1980+20601+16399) =0.77

While for the private algorithm:

Accuracy (OSR)= (TN+TP)/(TP +FP +TN+FN)

= (17818+ 45295)/(45295+ 37+ 17818+19150) = 0.76

Therefor, as Table 6.3 and Fig. 6.5 shows, the baseline model for the non-private model in the UNSW-NB15 dataset has a test accuracy of close to 77% while for the private model, the accuracy is 76%. In comparison, ANN achieves an accuracy of 63% in [89]. Using the same equation for the NSL-KDD dataset, the test accuracy of the non-private algorithm is 80% while in the private algorithm it is 79% compared with past results [87] at 80% also, in the ToN-IoT dataset the accuracy is 69% for the non-private algorithm and 74% for the private one.

Table 6.3: Performance evaluation of private algorithms compared with non-private algorithms

| Classifier | UNSW-NB15 | NSL-KDD | ToN-IoT |
|---|---|---|---|
| ANN | 63% [89] | 81%[87] | 77%[92] |
| Non-private | 77% | 80% | 69% |
| Private | 76% | 79% | 74% |
| private-K-Fold | 91% | 95% | 76% |

It is worth noting that the accuracy of the private model increases dramatically when using private-K-fold in all the three datasets. The accuracy is similar in all 10 folds. This means that our algorithm is consistent and we can be confident that training it on any data set and deploying it in production will lead to similar performance.

It should be pointed out that the comparison presented above is based on the three-layer ANN and other cases which have been cited. The use of K-fold is to make full use of all the data in the datasets; we have similar results with all the fold.

In the UNSW-NB15 dataset, when the privacy budget $\epsilon$ is set to $10^{-1}$ , our experiment demonstrates that the accuracy of our private model is extremely similar to that of a non-private model, as well as the model previously presented over the same dataset [89]. In the NSL-KDD dataset, when the privacy budget $\epsilon$ is in $10^{-2}$, our private model's accuracy is very close to that of the non-private model and model in [87], while in the

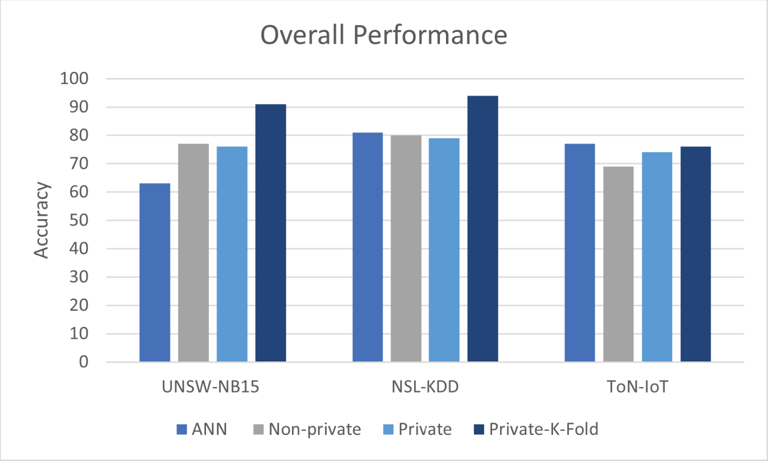


Figure 6.5: Overall performance comparison in terms of accuracy using UNBS-NB 15, NSL-KDD, ToN-IoT datasets

ToN-IoT dataset the accuracy is close to the non-private model when the privacy budget $\epsilon$ is $10^{-1}$.

As a result, this classification method's accuracy is comparable to that of the non-private classification method. It is shown through our experiments that the private model has the same accuracy as that of the non-private model and other models in distinguishing between botnet and normal network traffic. Thus, there is not great performance loss. Both the theoretical analysis and experiment results demonstrate that the private algorithm can efficiently and effectively perform data classification in a private setting without leaking private information and compromising privacy.

### 6.5.2 The impact of different choices of privacy budget on both utility and privacy

The model's utility increases as the privacy budget is increased. However, accuracy falls as the privacy budget rises. As a result, we look for a range of values of $\epsilon$ to balance utility and privacy.

Note that the privacy budget has an inverse relationship to the noise level. Consequently, from a differential privacy point of view, we are searching for a point with the highest noise level that still has good utility compared to the base classifier. We calculate the value of $\epsilon$ for the noise multiplier using the code given below:

```
noise = p.linspace (0.01, 5, 10)
```

```
epsilon = [1]
for noise mltp in noise:
eps, ord = compute_dp_sgd_privacy.compute_dp_sgd _privacy
  (n=N, batch_size=samples of each_node,
  noise multiplier=noise mltp,
  epochs=10, delta=1e-5)
epsilon. append (eps)
```

Table 6.4 shows the value of the privacy budget $\epsilon$ based on the noise multiplier. To evaluate the impact of different privacy budgets $\epsilon$ in terms of accuracy, we evaluate different values of $\epsilon$, ranging from $\epsilon = 10^{-3}$ to $\epsilon = 10^{3}$ as shown in Fig. 6.6.

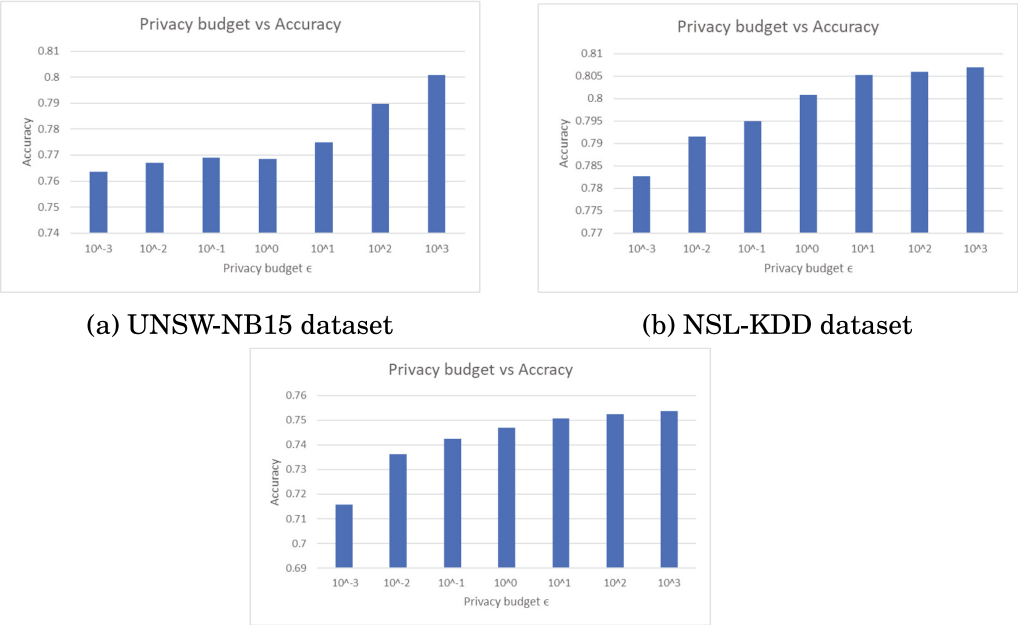
Table 6.4: Privacy budget $\epsilon$ and noise multiplier

| Noise | $\epsilon$ |
|---|---|
| 3 | $\approx 10^{-3}$ |
| 2.5 | $\approx 10^{-2}$ |
| 1.5 | $\approx 10^{-1}$ |
| 0.8 | $\approx 10^{0}$ |
| 0.4 | $\approx 10^{1}$ |
| 0.1 | $\approx 10^{2}$ |
| 0.01 | $\approx 10^{3}$ |

Based on the principle of differentiated privacy, a smaller privacy budget $\epsilon$ gives higher privacy protection. However, we can see that as the privacy budget is reduced, accuracy decreases, indicating that there is a trade-off between the level of privacy and prediction accuracy. The private algorithm achieves model accuracy close to the non-private baseline for $\epsilon = 10^{-1}, 10^{-2}$. As a result, the optimal value to add noise in the UNSW-NB15 dataset and ToN-IoT dataset was 2.5 whereas it was 1.5 in the NSL-KDD dataset which achieves an accuracy closer to the non-private model and increases the model utility.

### 6.5.3 Model utility

As privacy budget $\epsilon$ is varied, Fig. 6.7 compares the accuracy loss for the models trained using Rényi differential privacy's relaxed notions of differential privacy. As depicted in Fig. 6.7, the three datasets achieve an accuracy loss close to 0 when using optimal noise



(a) UNSW-NB15 dataset

(b) NSL-KDD dataset

(c) ToN-IoT dataset

Figure 6.6: Impact of different privacy budgets $\epsilon$ and level of noise on accuracy

values with privacy budget is $\epsilon = 10^{-1}$ for UNSW-NB15 and ToN-IoT datasets and $\epsilon = 10^{-2}$ for the NSL-KDD dataset. This indicates the high utility of the private algorithm in all three datasets.

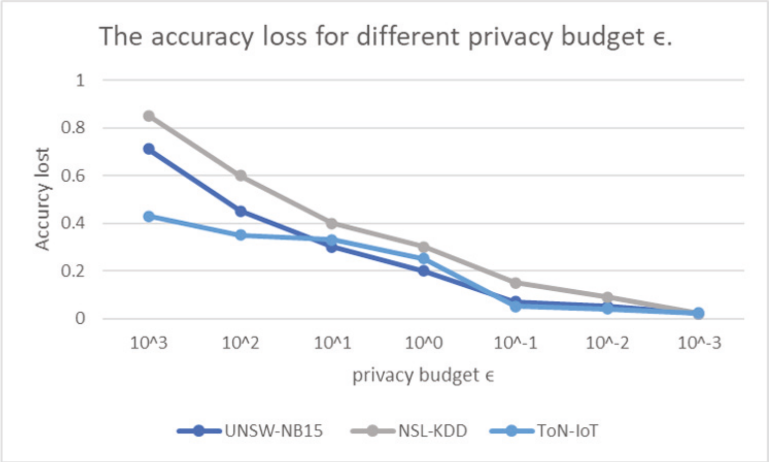


Figure 6.7: Accuracy lost for different privacy budget $\epsilon$

### 6.5.4 Privacy Leakage

In our case, the main attack is membership inference, which involves identifying whether or not a given data item is part of the model's training dataset. When an adversary has complete knowledge of a record, learning that it was used to train a specific model indicates that information is leaking through the model. On the three datasets, UNSW-NB 15, NSL-KDD, and ToN-IoT, our inference attacks can be used as metrics to assess privacy leaks from the model. Fig. 6.8 shows the overall amount of privacy leakage using the non-private algorithm and the private algorithm. Due to model over fitting, the impact of privacy leakage is substantially more severe for the non-private model disclosing over 30%, 60% and 20% of training set members, compared to only 17%, 11% and 15% for the private model for UNBS-NB 15 NSL-KDD and ToN-IoT respectively. The privacy mechanisms provide a substantial reduction in exposure in all three datasets.

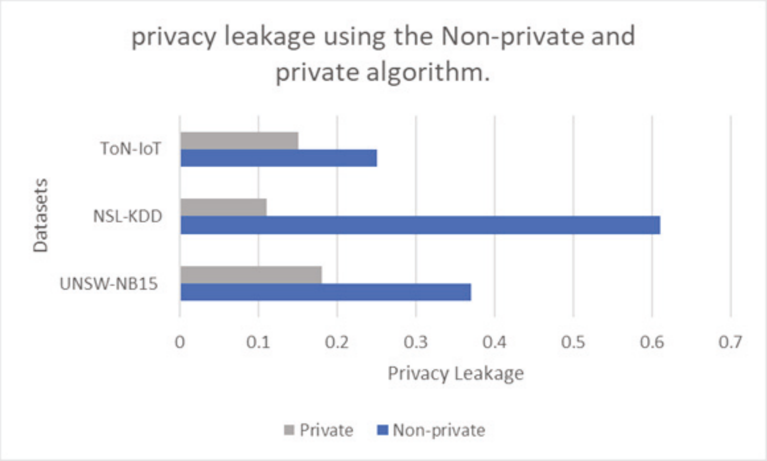


Figure 6.8: Privacy leakage

In adversarial scenarios, we investigate how much privacy leakage the Rényi differential privacy relaxed notions allow. On the three datasets, we empirically evaluate privacy leakage using the relaxed differential privacy notions for various selected privacy budget values $\epsilon$. Our experiment's main objective is to see how actual privacy leaks that an attacker can exploit in practise are influenced by implementation decisions regarding the privacy budget and relaxed notions of differential privacy.

Fig. 6.9 depicts the privacy leakage caused by membership inference attacks on private models trained using Rényi differential privacy. The amount of noise added to the model, the privacy budget $\epsilon$, corresponds to the degree of privacy leakage associated with each variant of differential privacy. The model can withstand an attack for $\epsilon < 10^{-1}$

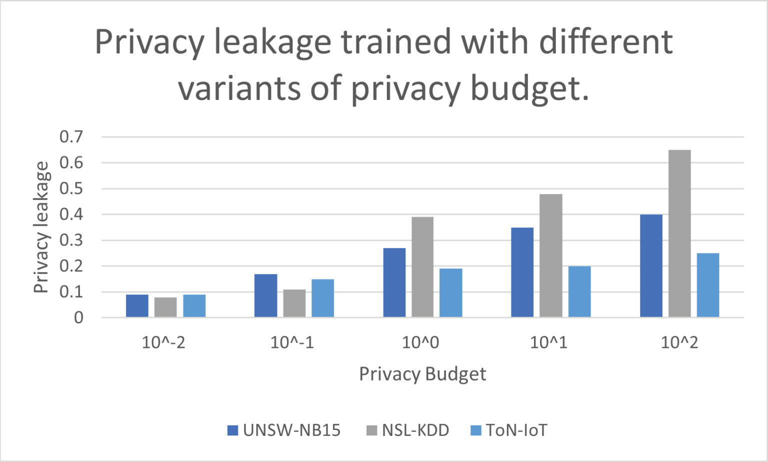


Figure 6.9: Privacy leakage trained with different privacy budget $\epsilon$

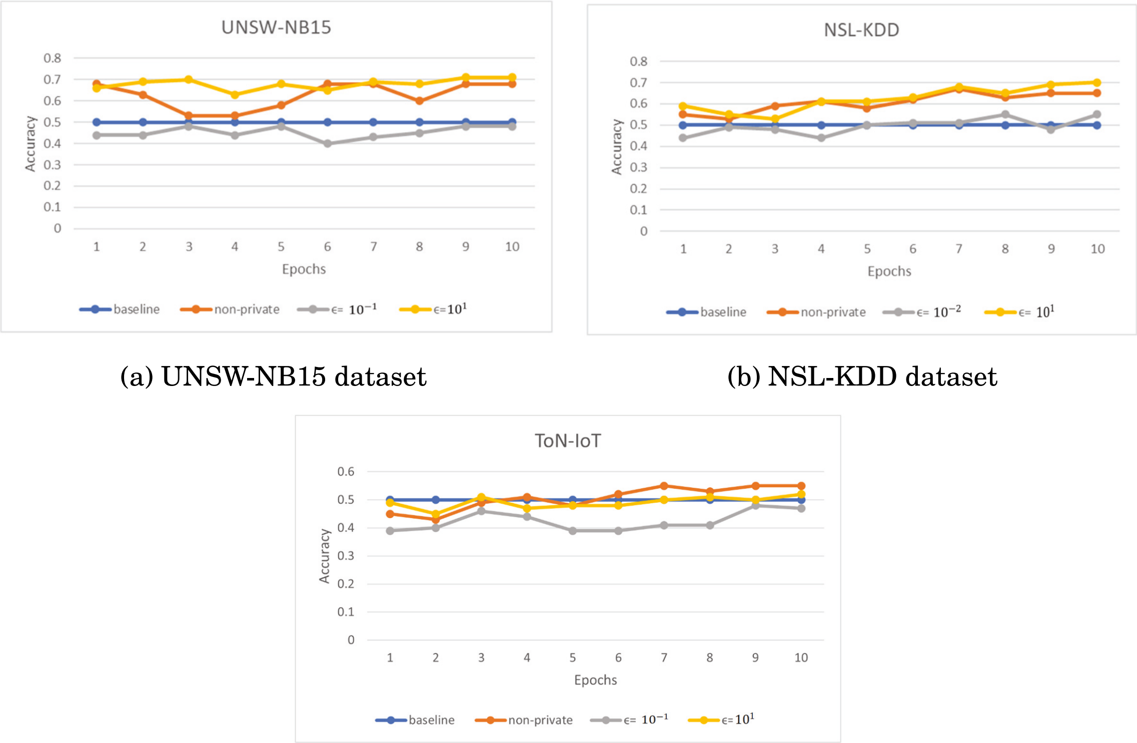
and $10^{-2}$, but the model discloses some privacy leakage for higher $\epsilon$ values. The leakage is significant at higher privacy budget $\epsilon$ values due to model overfitting. For $\epsilon = 10^2$, the model has leakage of more than 25% in all three datasets. Thus, the model provides both acceptable model utility and meaningful privacy when the $\epsilon$ value is less than $10^{-1}$.

To maximise model utility while minimising inference risk, Rényi differential privacy gives tighter cumulative noise limitations that reduces the amount of noise that can be generated. As a result, privacy is not free and lowering the noise requirements by relaxing the Rényi differential privacy definition brings with it additional privacy risks. While these definitions nevertheless satisfy the$(\epsilon;\delta)$-differential privacy claims, the practical utility of these assurances diminishes rapidly as the privacy budget $\epsilon$ rises.

#### 6.5.4.1 Attack Accuracy

On the UNSW-NB15 dataset, Fig. 6.10a displays the membership inference attack accuracy against private and non-private models. As predicted, the non-private model reveals a significant amount of information about its training dataset. As a result, attack accuracy ranges from 53% to 68%, which is immediately above the baseline of 50%. The plots for the private model with privacy budget $\epsilon = 10^{-1}$ demonstrate that the attack accuracy against the related models is not statistically significant and is practically as low as the baseline. This result shows that the private model has the ability to protect against an adversary who knows everything about the training mechanism and model parameters for these specific values of the privacy budget $\epsilon$ for this dataset.

On the other hand, a privacy budget of $\epsilon = 10^1$ achieves higher attack accuracy but



(a) UNSW-NB15 dataset

(b) NSL-KDD dataset

(c) ToN-IoT datasets

Figure 6.10: Attack accuracy

with lower model utility, as discussed earlier regarding privacy lost in Fig. 6.7. Therefore, the private model may be able to withstand strong adversaries while decreasing model utility (e.g., more privacy for less accuracy). Nonetheless, when providing an appropriate utility level for $\epsilon = 10^{-1}$, it demonstrates modest vulnerability to the membership inference attack.

On the NSL-KDD dataset, Fig. 6.10b exhibits the attack accuracy of the membership inference attack against private and non-private models. Compared to the UNSW-NB15 dataset, for different $\epsilon$ values, the membership inference attack on the NSL-KDD dataset is more successful.

With a particular interest in the optimal value of the privacy budget for $\epsilon = 10^{-2}$, this model's attack accuracy is around 50%, which is quite close to the baseline, resulting in significant information leakage regarding the training data. However, referring back to Fig. 6.6b, we can observe that the private model requires a minimum privacy budget of $\epsilon = 10^{-2}$ to achieve a comparable accuracy on the NSL-KDD dataset. Consequently, it is vulnerable to membership inference attacks when giving an acceptable utility level (e.g., $\epsilon = 10-2$), although it may provide security against such attacks by reducing model utility by a minor margin.(e.g., $\epsilon = 10^1$ or above).

In the ToN-IoT dataset, Fig. 6.10c demonstrates similar results to the UNSW-NB15 dataset for the private model with different privacy budget $\epsilon$. The attack accuracy against the related models is insignificant and is nearly as low as the baseline. This result demonstrates that the private model has the potential to provide protection against a strong adversary.

#### 6.5.4.2 F1-Score

Since the attack model's test dataset includes an equal number of members and non-members, our classifier's recall is 1.0 and precision is 0.5, resulting in a baseline F1-score of 0.67.

Table 6.5: F1-score on the three datasets

| | Baseline | Non-private | Optimal $\epsilon$ value | $\epsilon = 10^1$ |
|---|---|---|---|---|
| UNSW-NB15 | 0.67 | 0.53 | 0.44 | 0.75 |
| NSL-KDD | 0.67 | 0.54 | 0.49 | 0.61 |
| ToN-IoT | 0.67 | 0.45 | 0.42 | 0.49 |

Table 6.5 shows an average F1-score for 10 epochs of the membership inference attack against the private models (achieved by varying the privacy parameter $\epsilon$) as well as a non-private model on the three datasets.

The F1-score results in the UNSW-NB15 dataset following a similar pattern with regards to accuracy, with the F1-score for the optimal value being significantly below the baseline. The F1-score with a privacy budget of $\epsilon = 10^1$, on the other hand, clearly shows a high level of vulnerability to membership inference attacks, with the vulnerability approaching that of the totally non-private model. This is consistent with the attack accuracy scores we obtained on this dataset, and therefore, repeats the observation that the private model with $\epsilon = 10^1$ on UNSW-NB15 dataset cannot survive the membership inference attack.

In the NSL-KDD dataset, as before, while the private model with a privacy budget of $\epsilon = 10^{-2}$, is unable to live up to its promise to survive privacy attacks, for these models, the F1-score of the attack is typically lower than the baseline. Nevertheless, the private model with a value of $\epsilon = 10^1$ is vulnerable to a membership inference attack, making it nearly as bad as the totally non-private model. Similarly for attack accuracy, the membership inference attack reaches a higher F1-score on NSL-KDD than on UNSW-NB15.

For ToN-IoT datasets, F1-scores are under the baseline for the different private models as achieved by varying the value of $\epsilon$ as well as for a non-private model. For the optimal $\epsilon$ value the F1-score is the lowest, keeping its promise to survive the privacy attacks. The model also resists the attacks even when using a higher $\epsilon$ value and has a similar F1-score to the non-private model. This is in line with the attack accuracy results we attained on this dataset, therefore, confirming that a private model with varied values of $\epsilon$ on the ToN-IoT dataset can withstand a membership inference attack. On the other hand, it may cause poor model utility. Therefore, the optimal value $\epsilon = 10^{-1}$ is the best privacy budget value to offer an acceptable utility level on this dataset.

In some circumstances, such as the NSL-KDD dataset, our experiment results indicate that private models with higher utility levels are vulnerable to a membership inference attack. This vulnerability is comparable to that displayed by non-private models. Furthermore, when offering an acceptable utility level, it reveals moderate vulnerability to the membership inference attack in both UNSW-NB15 and ToN-IoT datasets.

We use attack accuracy and F1-score as performance metrics to quantify the private model's vulnerability. This analysis also explains the amount of the privacy parameter $\epsilon$ that is recommended, to protect from membership inference attack. Moreover, it suggests optimal values for $\epsilon$ that may offer a good trade-off between utility and privacy for the private model.

## 6.6 Summary

In this chapter, a privacy-preserving secure data aggregation method achieve by integrating blockchain and differential privacy in the context of smart homes is presented. Differential privacy has earned a reputation for providing verified privacy. However, to preserve utility, compromises must be made when it is employed for complex tasks like machine learning. It is important to properly comprehend the privacy consequences of these compromises. Our findings contribute to that understanding by revealing that commonly used deferential privacy relaxations such as Rényi differential privacy (RDP) may result in unacceptable utility-privacy trade-offs. Our techniques use TensorFlow machine learning software and are based on a differential private variant of stochastic gradient descent. The performance of the proposed framework was evaluated with three public datasets UNSW-NB15, NSL-KDD and ToN-IoT.

The experimental results demonstrate that the proposed framework outperforms some of the existing state-of-the-art techniques in terms of accuracy. We further systematically study the impact of membership inference attack against the differential private model. According to our findings, differential private models can only offer privacy protection against adversaries by sacrificing model utility to a substantial extent. As a result, we propose an empirical value of $\epsilon$ that optimally balances utility and privacy for the current datasets' smart home scenario. According to our results, our proposed architecture can ensure increased protection for smart home privacy.

CHAPTER 7

# CONCLUSION

In this chapter, we present all of the thesis questions as well as the evaluation outcomes. We highlight the key contributions of this research to the scientific research community. Towards this end, we discuss potential future works.

## 7.1 Summary of the thesis

The Internet of Things (IoT) has been a major talking point amongst technology enthusiasts in recent years. The IoT has been emerged and evolved rapidly, making the world's fabric around us smarter and more responsive. The smart home uses one such transformation of IoT, which seems to be the wave of the future. However, with the increasingly wide adoption of IoT, data security, and privacy concerns about how our data is collected and shared with others has also risen. To solve these challenges, an approach to data privacy and security in a smart home using blockchain technology is proposed in this thesis.

As discussed in previous chapters, we propose an authentication scheme that combines attribute-based access control with smart contracts and edge computing to create a secure framework for IoT devices in smart home systems. The edge server adds scalability to the system by offloading heavy processing activities and using a differential privacy method to aggregate data to the cloud securely and privately. Furthermore, we present several aspects of testing and implementing smart contracts, the differential private stochastic gradient descent algorithm, and system architecture and design.

We demonstrate the efficacy of our proposed system by thoroughly examining its security and privacy goals in terms of confidentiality, integrity, and availability. Our framework achieves the desired security and privacy goals and is resilient against modification, DoS attacks, data mining and linkage attacks. Finally, we undertake a performance evaluation to demonstrate the proposed scheme's feasibility and efficiency. The proposed work is summarised in the following.

This thesis introduced three major significant parts required to provide security and privacy-preserving mechanisms in the context of smart homes . These parts have been discussed, implemented, and tested in Chapters 3, 4, 5, 6 of this thesis. The research questions posed in Section 1.3 have been addressed in Chapters 3 through 6.

In Chapter 3, research question one is answered by developing a smart home framework that allows IoT devices to communicate securely with each other in the smart home context. We build a prototype of a private Ethereum network powered by smart contracts simulating IoT smart home devices on Raspberry Pis. This is accomplished by developing and analysing simplified smart contracts on the Ethereum blockchain as a proof of concept. Our framework is based on the fact that the Ethereum blockchain is tamper-proof and that the user keeps his private key in a secure manner. Also, because of the signed digital transaction and the blockchain's decentralised nature, attackers are unable to gain access to the network or impersonate a real user. However, several improvements to the prototype are considered in the next chapters.

In Chapter 4, research question two is answered. We further improve our prototype by proposing an authentication scheme which integrates attribute-based access control using smart contracts with an ERC-20 token (Ethereum Request For Comments) and edge computing to construct a secure framework for IoT devices in a smart home system. The edge server provides scalability to the system by offloading heavier computation tasks to the edge servers. We present the system architecture and design and discuss various aspects related to the testing and implementation of smart contracts. We show that our proposed scheme is secure by thoroughly analysing its security goals with respect to confidentiality, integrity and availability. Finally, we conduct a performance evaluation to demonstrate the feasibility and efficiency of the proposed scheme.

In Chapter 5, research question three is answered by extending our earlier framework and expanding the functional capabilities of our framework by adding differential privacy as a scheme to preserve the privacy of users. To send data from a private smart home to the cloud, we use machine learning and the differential privacy mechanism. We develop a decentralized, secure and privacy-preserving stochastic gradient descent (SGD) algorithm using blockchain. The experiment has been conducted to detect and classify a type of device in a private blockchain of the smart home. To provide accurate representations of the devices we use in the experiment, the dataset was produced by generating a pcap file using Wireshark to capture the network packets in our private network.

In Chapter 6, research question four is answered by assessing the performance of the differential privacy algorithm performance in three publicly available IoT datasets UNSW-NB15, NSL-KDD and ToN-IoT. We wanted to provide a more secure privacy guarantee for highly secure smart home data that was both more accurate and useful. The main purpose of using differential privacy as a privacy preserving scheme is to limit what can be inferred about individual training data from the model. Our techniques use the Rényi differential privacy (RDP) machine learning scheme and are based on a variant of the stochastic gradient descent function. Our findings show that differential private models can provide privacy protection against attackers by sacrificing a substantial amount of model utility. Therefore, we propose an empirical value of $\epsilon$ which optimally balances utility and privacy for the current smart home scenario datasets.

## 7.2 Contribution of the research

The research proposes a blockchain-based security and privacy framework to solve security challenges regarding IoT devices in smart home systems.

1. It employs blockchain technology as a distributed ledger to allow users and IoT devices in smart homes to be easily authenticated without relying on a trusted authority. This answers the thesis's first research question by demonstrating how the experimental prototype of Ethereum smart contracts can be used to secure access to smart home devices.

2. It uses ERC-20 token generation and an attribute-based access control mechanism that utilizes Ethereum smart contracts integrated with edge computing (servers) for authenticate user access to IoT smart home devices.

3. It incorporates blockchain and edge computing to set up a decentralised system to improve computing capabilities by divesting the mining and storage jobs to edge servers. The work on chapter four and six show better evaluation of our proposed architecture when compared with other work and that tackle the research question two.

4. It achieves security goals (confidentiality, integrity, and availability) and can overcome modification, DoS, linking, and inference attacks. That clearly answer the research question four by implementing several defence using the access control in chapter four and all related results has been explained in the chapter, and privacy mechanism in chapters five and six.

5. It ensures data privacy; the framework employs the differential privacy machine learning algorithm to send private smart home data to the cloud, this solve the research question three. According to our findings, differential private models can only provide privacy protection against attackers by significantly reducing model utility. Therefore, for the current datasets smart home scenario, it offers an empirical value of privacy budget $\epsilon$ that optimally balances utility and privacy.

6. It ensures the accurate classification of machine learning algorithms while increasing model privacy. The accuracy of any data analysis demands that noisy data have no impact on the correct output. Moreover, while adopting our approach, accurate data can be retrieved for additional analysis and enquiry. This has been addressed in chapter six where threat models, testing and measurement have been used to answer the research questions four. We Develop a new scheme for the threat analysis procedure using differential privacy algorithm. Moreover, we thoroughly test the proposed scheme against membership inference attack in particular to show how effective its metrics are for assessing privacy leakage.

This study focuses on developing a blockchain-based security and privacy framework that can offer a higher level of security in smart home systems while also introducing a decentralized approach to defend against the attacks mentioned above.

## 7.3 Future Directions

- Part-I of our thesis considered standard blockchains, i.e. Ethereum. This can be further improved by including a large number of IoT devices to check the scalability of the proposed framework. Also, our work can be enhanced by implementing the framework using other blockchain platforms such as Hyperledger Fabric and IBM Blockchain to achieve better performance and result.

- Part-II of our thesis is to explore the security of our framework. We build smart contracts that utilize attribute access control to authenticate users and IoT devices and ensure system resilience against modification and DoS attacks. The implementation can also be utilized to study the security of the framework against a broad range of possible smart home attacks.

- Part-III of our thesis related to privacy. The architecture employs a differential privacy machine learning algorithm to send private smart home data to the cloud. The proposed model's effectiveness is defined in terms of accuracy, utility and privacy leakage. We would like to improve these metrics by using larger datasets. Another interesting direction is to look into the impact of different differential privacy neural network models on membership inference attacks.

- As future work, A trust measure, how to ensure trust or measure trust in blockchain environment can be considered which potentially moves toward a more robust solution for a blockchain-based smart home framework.

APPENDIX A

# APPENDIX

# Bibliography

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, *Deep learning with differential privacy*, in Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, 2016, pp. 308–318.

[2] M. AbuNaser and A. A. Alkhatib, *Advanced survey of blockchain for the internet of things smart home*, in 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), IEEE, 2019, pp. 58–62.

[3] S. Aggarwal, R. Chaudhary, G. S. Aujla, A. Jindal, A. Dua, and N. Kumar, *Energychain: Enabling energy trading for smart homes using blockchains in smart grid ecosystem*, in Proceedings of the 1st ACM MobiHoc workshop on networking and cybersecurity for smart cities, 2018, pp. 1–6.

[4] L. Aguilar, S. Peralta, and D. Mauricio, *Technological architecture for iot smart buildings*, in 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), IEEE, 2020, pp. 1–6.

[5] S. B. Ahsan, R. Yang, S. A. Noghabi, and I. Gupta, *Home, safehome: Ensuring a safe and reliable home using the edge*, in 2nd {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 19), 2019.

[6] M. Alaa, A. A. Zaidan, B. B. Zaidan, M. Talal, and M. L. M. Kiah, *A review of smart home applications based on internet of things*, Journal of Network and Computer Applications, 97 (2017), pp. 48–65.

[7] M. R. Alam, *Exact and Approximate Solutions for Energy Cost Optimization in Smart Homes*, PhD thesis, Carleton University, 2017.

[8] J. Ali, T. Ali, S. Musa, and A. Zahrani, *Towards secure iot communication with smart contracts in a blockchain infrastructure*, arXiv preprint arXiv:2001.01837, (2020).

[9] A. Alnemari, S. Arodi, V. R. Sosa, S. Pandey, C. Romanowski, R. Raj, and S. Mishra, *Protecting infrastructure data via enhanced access control, blockchain and differential privacy*, in International Conference on Critical Infrastructure Protection, Springer, 2018, pp. 113–125.

[10] M. Amadeo, A. Molinaro, S. Y. Paratore, A. Altomare, A. Giordano, and C. Mastroianni, *A cloud of things framework for smart home services based on information centric networking*, in 2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC), IEEE, 2017, pp. 245–250.

[11] Z. Amin, H. Singh, and N. Sethi, *Review on fault tolerance techniques in cloud computing*, International Journal of Computer Applications, 116 (2015).

[12] N. Apthorpe, D. Reisman, S. Sundaresan, A. Narayanan, and N. Feamster, *Spying on the smart home: Privacy attacks and defenses on encrypted iot traffic*, arXiv preprint arXiv:1708.05044, (2017).

[13] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, *A trustworthy privacy preserving framework for machine learning in industrial iot systems*, IEEE Transactions on Industrial Informatics, 16 (2020), pp. 6092–6102.

[14] S. Arif, M. A. Khan, S. U. Rehman, M. A. Kabir, and M. Imran, *Investigating smart home security: Is blockchain the answer?*, IEEE Access, 8 (2020), pp. 117802–117816.

[15] H. Atlam and G. Wills, *Technical aspects of blockchain and iot advances in computers*, 2019.

[16] Y. N. Aung and T. Tantidham, *Review of ethereum: Smart home case study*, in 2017 2nd International Conference on Information Technology (INCIT), IEEE, 2017, pp. 1–4.

[17] S. Avizheh, T. T. Doan, X. Liu, and R. Safavi-Naini, *A secure event logging system for smart homes*, in Proceedings of the 2017 Workshop on Internet of Things Security and Privacy, 2017, pp. 37–42.

[18] L. Axon, *Privacy-awareness in blockchain-based pki*, Cdt Technical Paper Series, 21 (2015), p. 15.

[19] J. M. Batalla, A. Vasilakos, and M. Gajewski, *Secure smart homes: Opportunities and challenges*, ACM Computing Surveys (CSUR), 50 (2017), pp. 1–32.

[20] B. Bera, A. K. Das, M. Obaidat, P. Vijayakumar, K.-F. Hsiao, and Y. Park, *Ai-enabled blockchain-based access control for malicious attacks detection and mitigation in ioe*, IEEE Consumer Electronics Magazine, (2020).

[21] A. Boudguiga, N. Bouzerna, L. Granboulan, A. Olivereau, F. Quesnel, A. Roger, and R. Sirdey, *Towards better availability and accountability for iot updates by means of a blockchain*, in 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, 2017, pp. 50–58.

[22] M. A. Bouras, B. Xia, A. O. Abuassba, H. Ning, and Q. Lu, *Iot-ccac: a blockchain-based consortium capability access control approach for iot*, PeerJ Computer Science, 7 (2021), p. e455.

[23] M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part I*, vol. 4051, Springer, 2006.

[24] M. Bun and T. Steinke, *Concentrated differential privacy: Simplifications, extensions, and lower bounds*, in Theory of Cryptography Conference, Springer, 2016, pp. 635–658.

[25] V. Buterin et al., *A next-generation smart contract and decentralized application platform*, White Paper, 3 (2014).

[26] V. Buterin and F. Vogelsteller, *Erc20 token standard*, URL: https://theethereum. wiki/w/index. php/ERC20 Token Standard, (2015).

[27] S. Chavda, *Smart home technology adoption rises with more australians over 60 buying tech devices*.
`https://www.savvy.com.au/smart-home-technology-adoption-rises-with-more-australia`
May 2022.

[28] H. Chen, M. Pendleton, L. Njilla, and S. Xu, *A survey on ethereum systems security: Vulnerabilities, attacks, and defenses*, ACM Computing Surveys (CSUR), 53 (2020), pp. 1–43.

[29] X. Chen, J. Ji, C. Luo, W. Liao, and P. Li, *When machine learning meets blockchain: A decentralized, privacy-preserving and secure design*, in 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2018, pp. 1178–1187.

[30] X. Chen, X. Wang, and K. Yang, *Asynchronous blockchain-based privacy-preserving training framework for disease diagnosis*, in 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 5469–5473.

[31] K. Christidis and M. Devetsikiotis, *Blockchains and smart contracts for the internet of things*, Ieee Access, 4 (2016), pp. 2292–2303.

[32] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, *A survey on security and privacy issues of bitcoin*, IEEE Communications Surveys & Tutorials, 20 (2018), pp. 3416–3452.

[33] E. F. Coutinho, D. E. Paulo, A. W. Abreu, and I. B. Carla, *Towards cloud computing and blockchain integrated applications*, in 2020 IEEE International Conference on Software Architecture Companion (ICSA-C), IEEE, 2020, pp. 139–142.

[34] J. P. Cruz, Y. Kaji, and N. Yanai, *Rbac-sc: Role-based access control using smart contract*, Ieee Access, 6 (2018), pp. 12240–12251.

[35] P. Cuff and L. Yu, *Differential privacy as a mutual information constraint*, in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 43–54.

[36] T. L. N. Dang and M. S. Nguyen, *An approach to data privacy in smart home using blockchain technology*, in 2018 International Conference on Advanced Computing and Applications (ACOMP), IEEE, 2018, pp. 58–64.

[37] P. De Filippi, *The interplay between decentralization and privacy: the case of blockchain technologies*, Journal of Peer Production, Issue, (2016).

[38] Y.-A. De Montjoye, L. Radaelli, V. K. Singh, et al., *Unique in the shopping mall: On the reidentifiability of credit card metadata*, Science, 347 (2015), pp. 536–539.

[39] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, *Scalable and secure architecture for distributed iot systems*, in 2020 IEEE Technology & Engineering Management Conference (TEMSCON), IEEE, 2020, pp. 1–6.

[40] P. Dhillon and M. Singh, *Internet of things attacks and countermeasure access control techniques: a review*, International Journal of Applied Engineering Research, 14 (2019), pp. 1689–1698.

[41] B. Dickson, *Decentralizing iot networks through blockchain*, TechCrunch, TechCrunch, (2016).

[42] X. Dong, B. Guo, Y. Shen, X. Duan, Y. Shen, and H. Zhang, *A self-controllable and balanced data sharing model*, IEEE Access, 7 (2019), pp. 103275–103290.

[43] A. Dorri, S. S. Kanhere, and R. Jurdak, *Blockchain in internet of things: challenges and solutions*, arXiv preprint arXiv:1608.05187, (2016).

[44] ——, *Towards an optimized blockchain for iot*, in 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI), IEEE, 2017, pp. 173–178.

[45] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, *Blockchain for iot security and privacy: The case study of a smart home*, in 2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops), IEEE, 2017, pp. 618–623.

[46] ——, *Lsb: A lightweight scalable blockchain for iot security and anonymity*, Journal of Parallel and Distributed Computing, 134 (2019), pp. 180–197.

[47] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, *Blockchain: A distributed solution to automotive security and privacy*, IEEE Communications Magazine, 55 (2017), pp. 119–125.

[48] H. Duan, Y. Zheng, Y. Du, A. Zhou, C. Wang, and M. H. Au, *Aggregating crowd wisdom via blockchain: A private, correct, and robust realization*, in 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom, IEEE, 2019, pp. 1–10.

[49] S. K. Dwivedi, R. Amin, and S. Vollala, *Blockchain based secured information sharing protocol in supply chain management system with key distribu-*

*tion mechanism*, Journal of Information Security and Applications, 54 (2020), p. 102554.

[50] S. K. DWIVEDI, R. AMIN, S. VOLLALA, AND R. CHAUDHRY, *Blockchain-based secured event-information sharing protocol in internet of vehicles for smart cities*, Computers & Electrical Engineering, 86 (2020), p. 106719.

[51] C. DWORK, *Differential privacy: A survey of results*, in International conference on theory and applications of models of computation, Springer, 2008, pp. 1–19.

[52] C. DWORK AND G. N. ROTHBLUM, *Concentrated differential privacy*, arXiv preprint arXiv:1603.01887, (2016).

[53] H. EBADI, D. SANDS, AND G. SCHNEIDER, *Differential privacy: Now it's getting personal*, ACM Sig plan Notices, 50 (2015), pp. 69–81.

[54] G. EIBL AND D. ENGEL, *Differential privacy for real smart metering data*, Computer Science-Research and Development, 32 (2017), pp. 173–182.

[55] W. EJAZ AND A. ANPALAGAN, *Internet of things for smart cities: technologies, big data and security*, Springer, 2019.

[56] Q. FENG, D. HE, S. ZEADALLY, M. K. KHAN, AND N. KUMAR, *A survey on privacy protection in blockchain system*, Journal of Network and Computer Applications, 126 (2019), pp. 45–58.

[57] E. GAETANI, L. ANIELLO, R. BALDONI, F. LOMBARDI, A. MARGHERI, AND V. SASSONE, *Blockchain-based database to ensure data integrity in cloud computing environments*, (2017).

[58] K. GAI, J. GUO, L. ZHU, AND S. YU, *Blockchain meets cloud computing: A survey*, IEEE Communications Surveys & Tutorials, 22 (2020), pp. 2009–2030.

[59] K. GAI, Y. WU, L. ZHU, M. QIU, AND M. SHEN, *Privacy-preserving energy trading using consortium blockchain in smart grid*, IEEE Transactions on Industrial Informatics, 15 (2019), pp. 3548–3558.

[60] K. GAI, Y. WU, L. ZHU, Z. ZHANG, AND M. QIU, *Differential privacy-based blockchain for industrial internet-of-things*, IEEE Transactions on Industrial Informatics, 16 (2019), pp. 4156–4165.

[61] P. P. Gaikwad, J. P. Gabhane, and S. S. Golait, *A survey based on smart homes system using internet-of-things*, in 2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), IEEE, 2015, pp. 0330–0335.

[62] D. Geneiatakis, I. Kounelis, R. Neisse, I. Nai-Fovino, G. Steri, and G. Baldini, *Security and privacy issues for an iot based smart home*, in 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, 2017, pp. 1292–1297.

[63] S. S. Gill, P. Garraghan, and R. Buyya, *Router: Fog enabled cloud based intelligent resource management approach for smart home iot devices*, Journal of Systems and Software, 154 (2019), pp. 125–138.

[64] A. M. Girgis, D. Data, S. Diggavi, A. T. Suresh, and P. Kairouz, *On the renyi differential privacy of the shuffle model*, arXiv preprint arXiv:2105.05180, (2021).

[65] R. M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik, *Sgd: General analysis and improved rates*, in International Conference on Machine Learning, PMLR, 2019, pp. 5200–5209.

[66] V. Goyal, O. Pandey, A. Sahai, and B. Waters, *Attribute-based encryption for fine-grained access control of encrypted data*, in Proceedings of the 13th ACM conference on Computer and communications security, 2006, pp. 89–98.

[67] K. Gram-Hanssen and S. J. Darby, *‚Äúhome is where the smart is‚Äù? evaluating smart home research and approaches against the concept of home*, Energy Research & Social Science, 37 (2018), pp. 94–101.

[68] M. D. Gregorio, *Blockchain: A new tool to cut costs*. `https://www.pwc.com/m1/en/media-centre/articles/blockchain-new-tool-to-cut-costs.html`, 2017.

[69] R. Gupta, P. Kanungo, and N. Dagdee, *Hd-maabe: Hierarchical distributed multi-authority attribute based encryption for enabling open access to shared organizational data*, in International Conference on Intelligent Computing and Smart Communication 2019, Springer, 2020, pp. 183–193.

[70] S. Gurung and S. Chauhan, *A dynamic threshold based algorithm for improving security and performance of aodv under black-hole attack in manet*, Wireless Networks, 25 (2019), pp. 1685–1695.

[71] S. Gusmeroli, S. Piccione, and D. Rotondi, *A capability-based security approach to manage access control in the internet of things*, Mathematical and Computer Modelling, 58 (2013), pp. 1189–1205.

[72] R. Hadianto and T. W. Purboyo, *A survey paper on botnet attacks and defenses in software defined networking*, International Journal of Applied Engineering Research, 13 (2018), pp. 483–489.

[73] D. Han, H. Kim, and J. Jang, *Blockchain based smart door lock system*, in 2017 International conference on information and communication technology convergence (ICTC), IEEE, 2017, pp. 1165–1167.

[74] S. H. Hashemi, F. Faghri, P. Rausch, and R. H. Campbell, *World of empowered iot users*, in 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI), IEEE, 2016, pp. 13–24.

[75] M. U. Hassan, M. H. Rehmani, and J. Chen, *Privacy preservation in blockchain based iot systems: Integration issues, prospects, challenges, and future research directions*, Future Generation Computer Systems, 97 (2019), pp. 512–529.

[76] ——, *Differential privacy in blockchain technology: A futuristic approach*, Journal of Parallel and Distributed Computing, 145 (2020), pp. 50–74.

[77] ——, *Performance evaluation of differential privacy mechanisms in blockchain based smart metering*, arXiv preprint arXiv:2007.09802, (2020).

[78] J. Herrera-Joancomartí and C. Pérez-Solà, *Privacy in bitcoin transactions: new challenges from blockchain scalability solutions*, in International Conference on Modeling Decisions for Artificial Intelligence, Springer, 2016, pp. 26–44.

[79] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone, et al., *Guide to attribute based access control (abac) definition and considerations (draft)*, NIST Special Publication, 800 (2013), pp. 1–54.

[80] S. HUH, S. CHO, AND S. KIM, *Managing iot devices using blockchain platform*, in 2017 19th International Conference on Advanced Communication Technology (ICACT), IEEE, 2017, pp. 464–467.

[81] X. HUO AND M. LIU, *Privacy-preserving distributed multi-agent cooperative optimization–paradigm design and privacy analysis*, IEEE Control Systems Letters, (2021).

[82] M. IANSITI AND K. R. LAKHANI, *The truth about blockchain*. `https://hbr.org/2017/01/the-truth-about-blockchain`, 2017.

[83] A. ISMAILISUFI, T. POPOVIĆ, N. GLIGORIĆ, S. RADONJIC, AND S. ŠANDI, *A private blockchain implementation using multichain open source platform*, in 2020 24th International Conference on Information Technology (IT), IEEE, 2020, pp. 1–4.

[84] B. JAYARAMAN AND D. EVANS, *Evaluating differentially private machine learning in practice*, in 28th {USENIX} Security Symposium ({USENIX} Security 19), 2019, pp. 1895–1912.

[85] N. JOHNSON, J. P. NEAR, AND D. SONG, *Towards practical differential privacy for sql queries*, Proceedings of the VLDB Endowment, 11 (2018), pp. 526–539.

[86] S. W. JUNG AND S. JUNG, *Personal oauth authorization server and push oauth for internet of things*, International Journal of Distributed Sensor Networks, 13 (2017), p. 1550147717712627.

[87] M. A. KHAN, S. ABBAS, A. REHMAN, Y. SAEED, A. ZEB, M. I. UDDIN, N. NASSER, AND A. ALI, *A machine learning approach for blockchain-based smart home networks security*, IEEE Network, 35 (2020), pp. 223–229.

[88] T. KIM, J. KIM, J. NOH, AND S. CHO, *Lightweight smart home security system using multiple rss-based voting*, in 2018 IEEE International Conference on Consumer Electronics (ICCE), IEEE, 2018, pp. 1–4.

[89] N. KORONIOTIS, N. MOUSTAFA, E. SITNIKOVA, AND J. SLAY, *Towards developing network forensic mechanism for botnet activities in the iot based on machine learning techniques*, in International Conference on Mobile Networks and Management, Springer, 2017, pp. 30–44.

[90] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, *Hawk: The blockchain model of cryptography and privacy-preserving smart contracts*, in 2016 IEEE symposium on security and privacy (SP), IEEE, 2016, pp. 839–858.

[91] N. Kshetri, *Can blockchain strengthen the internet of things?*, IT professional, 19 (2017), pp. 68–72.

[92] P. Kumar, G. P. Gupta, and R. Tripathi, *Tp2sf: A trustworthy privacy-preserving secured framework for sustainable smart cities by leveraging blockchain and machine learning*, Journal of Systems Architecture, 115 (2021), p. 101954.

[93] C. Lazaroiu and M. Roscia, *Smart district through iot and blockchain*, in 2017 IEEE 6th international conference on renewable energy research and applications (ICRERA), IEEE, 2017, pp. 454–461.

[94] Y. Lee, S. Rathore, J. H. Park, and J. H. Park, *A blockchain-based smart home gateway architecture for preventing data forgery*, Human-centric Computing and Information Sciences, 10 (2020), pp. 1–14.

[95] M. Li, W. Gu, W. Chen, Y. He, Y. Wu, and Y. Zhang, *Smart home: architecture, technologies and systems*, Procedia Computer Science, 131 (2018), pp. 393–400.

[96] C. Lin, D. He, N. Kumar, X. Huang, P. Vijayakumar, and K.-K. R. Choo, *Homechain: A blockchain-based secure mutual authentication system for smart homes*, IEEE Internet of Things Journal, 7 (2019), pp. 818–829.

[97] H. Lin and N. W. Bergmann, *Iot privacy and security challenges for smart home environments*, Information, 7 (2016), p. 44.

[98] I.-C. Lin and T.-C. Liao, *A survey of blockchain security issues and challenges.*, International Journal of Security and Networks, 19 (2017), pp. 653–659.

[99] Y. Liu, Q. Lu, C. Zhu, and Q. Yu, *A blockchain-based platform architecture for multimedia data management*, Multimedia Tools and Applications, (2021), pp. 1–17.

[100] T. Ma, *Reliability and security investigations on video streaming over satellite links*.

[101] Y. Mahmoodi, C. Gross, S. Reiter, A. Viehl, and O. Bringmann, *Security requirement modeling for a secure energy trading platform*.

[102] N. Malik, *Blockchain Based Security for Vehicular Ad hoc Networks*, PhD thesis, 2020.

[103] J. Mao, Q. Lin, and J. Bian, *Application of learning algorithms in smart home iot system security*, Mathematical Foundations of Computing, 1 (2018), p. 63.

[104] Z. Mei, *The application of cloud computing in the practice teaching of business english major in higher vocational colleges*, in Journal of Physics: Conference Series, vol. 1634, IOP Publishing, 2020, p. 012009.

[105] T. N. Minh-Thai and N. Thai-Nghe, *An approach for developing intelligent systems in smart home environment*, in International Conference on Future Data and Security Engineering, Springer, 2015, pp. 147–161.

[106] I. Mironov, *Rényi differential privacy*, in 2017 IEEE 30th Computer Security Foundations Symposium (CSF), IEEE, 2017, pp. 263–275.

[107] I. Mironov, K. Talwar, and L. Zhang, *R\'enyi differential privacy of the sampled gaussian mechanism*, arXiv preprint arXiv:1908.10530, (2019).

[108] S. N. Mohanty, K. Ramya, S. S. Rani, D. Gupta, K. Shankar, S. Lakshmanaprabu, and A. Khanna, *An efficient lightweight integrated blockchain (elib) model for iot security and privacy*, Future Generation Computer Systems, 102 (2020), pp. 1027–1037.

[109] S. Moin, A. Karim, Z. Safdar, K. Safdar, E. Ahmed, and M. Imran, *Securing iots in distributed blockchain: Analysis, requirements and open issues*, Future Generation Computer Systems, 100 (2019), pp. 325–343.

[110] M. Moniruzzaman, S. Khezr, A. Yassine, and R. Benlamri, *Blockchain for smart homes: Review of current trends and research challenges*, Computers & Electrical Engineering, 83 (2020), p. 106585.

[111] P. Moriggl, P. M. Asprion, and B. Schneider, *Blockchain technologies towards data privacy,Äîhyperledger sawtooth as unit of analysis*, in New Trends in Business Information Systems and Technology, Springer, 2021, pp. 299–313.

[112] M. Möser, R. Böhme, and D. Breuker, *An inquiry into money laundering tools in the bitcoin ecosystem*, in 2013 APWG eCrime researchers summit, Ieee, 2013, pp. 1–14.

[113] C. V. B. Murthy and M. L. Shri, *A survey on integrating cloud computing with blockchain*, in 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), IEEE, 2020, pp. 1–6.

[114] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, Decentralized Business Review, (2008), p. 21260.

[115] ——, *Bitcoin: A peer-to-peer electronic cash system*, tech. rep., Manubot, 2019.

[116] Y. Nakamura, Y. Zhang, M. Sasabe, and S. Kasahara, *Capability-based access control for the internet of things: an ethereum blockchain-based scheme*, in 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–6.

[117] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, *Integration of blockchain and cloud of things: Architecture, applications and challenges*, IEEE Communications Surveys & Tutorials, 22 (2020), pp. 2521–2549.

[118] A. Ouaddah, A. Abou Elkalam, and A. Ait Ouahman, *Fairaccess: a new blockchain-based access control framework for the internet of things*, Security and Communication Networks, 9 (2016), pp. 5943–5964.

[119] D. Pal, S. Funilkul, N. Charoenkitkarn, and P. Kanthamanon, *Internet-of-things and smart homes for elderly healthcare: An end user perspective*, IEEE Access, 6 (2018), pp. 10483–10496.

[120] M. K. Pandit, R. N. Mir, and M. A. Chihisti, *Machine learning at the edge of internet of things*, CSI Communications, 41 (2017), pp. 28–30.

[121] N. Papernot, *TensorFlow: Implement differential privacy with tensorflow privacy*. Software available from tensorflow.org.

[122] S. Patil, S. Joshi, and D. Patil, *Enhanced privacy preservation using anonymization in iot-enabled smart homes*, in Smart Intelligent Computing and Applications, Springer, 2020, pp. 439–454.

[123] G. PENDER-BEY, *The parkerian hexad*, Information Security Program at Lewis University, (2016).

[124] S. PIRBHULAL, H. ZHANG, M. E. E ALAHI, H. GHAYVAT, S. C. MUKHOPADHYAY, Y.-T. ZHANG, AND W. WU, *A novel secure iot-based smart home automation system using a wireless sensor network*, Sensors, 17 (2017), p. 69.

[125] N. PRUSTY, *Building blockchain projects*, Packt Publishing Ltd, 2017.

[126] A. QASHLAN, P. NANDA, AND X. HE, *Automated ethereum smart contract for block chain based smart home security*, in Smart Systems and IoT: Innovations in Computing, Springer, 2020, pp. 313–326.

[127] A. QASHLAN, P. NANDA, AND X. HE, *Security and privacy implementation in smart home: Attributes based access control and smart contracts*, in 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, 2020, pp. 951–958.

[128] A. QASHLAN, P. NANDA, X. HE, AND M. MOHANTY, *Privacy-preserving mechanism in smart home using blockchain*, IEEE Access, 9 (2021), pp. 103651–103669.

[129] L. QI, Y. LU, AND Y. LI, *A centralized home gateway architecture based on the analysis of user log*, (2014).

[130] C. QU, M. TAO, AND R. YUAN, *A hypergraph-based blockchain model and application in internet of things-enabled smart homes*, Sensors, 18 (2018), p. 2784.

[131] M. A. RAHMAN, M. RASHID, S. BARNES, M. S. HOSSAIN, E. HASSANAIN, AND M. GUIZANI, *An iot and blockchain-based multi-sensory in-home quality of life framework for cancer patients*, in 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), IEEE, 2019, pp. 2116–2121.

[132] H. RAJADURAI AND U. D. GANDHI, *A stacked ensemble learning model for intrusion detection in wireless network*, Neural Computing and Applications, (2020), pp. 1–9.

[133] S. RAJESHWARI AND A. CHANDRASEKAR, *Access control management in collaborative environment: A review*.

[134] E.-I. REMIX, *Welcome to remix documentation*, 2019.

[135] I. Riabi, H. K. B. Ayed, and L. A. Saidane, *A survey on blockchain based access control for internet of things*, in 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), IEEE, 2019, pp. 502–507.

[136] C. R. G. Rodríguez et al., *Using differential privacy for the internet of things*, in IFIP International Summer School on Privacy and Identity Management, Springer, 2016, pp. 201–211.

[137] G. Rosner and E. Kenneally, *Privacy and the internet of things: Emerging frameworks for policy and design*, in Rosner, Gilad and Kenneally, Erin, Privacy and the Internet of Things: Emerging Frameworks for Policy and Design (June 7, 2018). UC Berkeley Center for Long-Term Cybersecurity/Internet of Things Privacy Forum, 2018.

[138] S. Rouhani, R. Belchior, R. S. Cruz, and R. Deters, *Distributed attribute-based access control system using a permissioned blockchain*, arXiv preprint arXiv:2006.04384, (2020).

[139] B. E. Sabir, M. Youssfi, O. Bouattane, and H. Allali, *Towards a new model to secure iot-based smart home mobile agents using blockchain technology*, Engineering, Technology & Applied Science Research, 10 (2020), pp. 5441–5447.

[140] G. Sahith, *Home automation towards security and privacy to accomplish as smart home using data analytics*, International Journal of Research in computer and communication Technology (IJRCCT), 6 (2017), p. 12.

[141] O. Samuel, N. Javaid, M. Awais, Z. Ahmed, M. Imran, and M. Guizani, *A blockchain model for fair data sharing in deregulated smart grids*, in 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–7.

[142] M. Satyanarayanan, *The emergence of edge computing*, Computer, 50 (2017), pp. 30–39.

[143] D. Servos and S. L. Osborn, *Current research and open problems in attribute-based access control*, ACM Computing Surveys (CSUR), 49 (2017), pp. 1–45.

[144] A. R. Shahid, N. Pissinou, C. Staier, and R. Kwan, *Sensor-chain: a lightweight scalable blockchain framework for internet of things*, in 2019 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, 2019, pp. 1154–1161.

[145] W. She, Z.-H. Gu, X.-K. Lyu, Q. Liu, Z. Tian, and W. Liu, *Homomorphic consortium blockchain for smart home system sensitive data privacy preserving*, IEEE Access, 7 (2019), pp. 62058–62070.

[146] K.-A. Shim, *A survey of public-key cryptographic primitives in wireless sensor networks*, IEEE Communications Surveys & Tutorials, 18 (2015), pp. 577–601.

[147] E.-J. Shin, *Addressing Privacy, Fairness, and Scalability Challenges for Context-aware Applications in Smart Environments*, University of California, Irvine, 2019.

[148] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, *Membership inference attacks against machine learning models*, in 2017 IEEE Symposium on Security and Privacy (SP), IEEE, 2017, pp. 3–18.

[149] P. K. Singh, R. Singh, S. K. Nandi, and S. Nandi, *Managing smart home appliances with proof of authority and blockchain*, in International Conference on Innovations for Community Services, Springer, 2019, pp. 221–232.

[150] S. Singh, I.-H. Ra, W. Meng, M. Kaur, and G. H. Cho, *Sh-blockcc: A secure and efficient internet of things smart home architecture based on cloud computing and blockchain technology*, International Journal of Distributed Sensor Networks, 15 (2019), p. 1550147719844159.

[151] V. Sivaraman, H. H. Gharakheili, A. Vishwanath, R. Boreli, and O. Mehani, *Network-level security and privacy control for smart-home iot devices*, in 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2015, pp. 163–167.

[152] S. Song, K. Chaudhuri, and A. D. Sarwate, *Stochastic gradient descent with differentially private updates*, in 2013 IEEE Global Conference on Signal and Information Processing, IEEE, 2013, pp. 245–248.

[153] B. Sowmiya, V. Abhijith, S. Sudersan, R. S. J. Sundar, M. Thangavel, and P. Varalakshmi, *A survey on security and privacy issues in contact tracing application of covid-19*, SN Computer Science, 2 (2021), pp. 1–11.

[154] B. L. R. Stojkoska and K. V. Trivodaliev, *A review of internet of things for smart home: Challenges and solutions*, Journal of Cleaner Production, 140 (2017), pp. 1454–1464.

[155] A. Stojmenski, B. Joksimoski, I. Chorbev, and V. Trajkovikj, *Smart home environment aimed for people with physical disabilities*, in 2016 IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP), IEEE, 2016, pp. 13–18.

[156] Y. Sun, H. Song, A. J. Jara, and R. Bie, *Internet of things and big data analytics for smart and connected communities*, IEEE Access, 4 (2016), pp. 766–773.

[157] P. Tasatanattakool and C. Techapanupreeda, *Blockchain: Challenges and applications*, in 2018 International Conference on Information Networking (ICOIN), IEEE, 2018, pp. 473–475.

[158] A. Thakkar, A. A. Bhatti, and J. Vasa, *Correlation based anonymization using generalization and suppression for disclosure problems*, in Advances in Intelligent Informatics, Springer, 2015, pp. 581–592.

[159] M. Thakur et al., *Authentication, authorization and accounting with ethereum blockchain*, (2017).

[160] V. Torra and G. Navarro-Arribas, *Big data privacy and anonymization*, in IFIP International Summer School on Privacy and Identity Management, Springer, 2016, pp. 15–26.

[161] I. Torre, F. Koceva, O. R. Sanchez, and G. Adorni, *A framework for personal data protection in the iot*, in 2016 11th international conference for internet technology and secured transactions (ICITST), IEEE, 2016, pp. 384–391.

[162] L. Touati and Y. Challal, *Poster: Activity-based access control for iot*, in Proceedings of the 1st International Workshop on Experiences with the Design and Implementation of Smart Objects, 2015, pp. 29–30.

[163] A. Vacca, A. Di Sorbo, C. A. Visaggio, and G. Canfora, *A systematic literature review of blockchain and smart contract development: Techniques, tools, and open challenges*, Journal of Systems and Software, 174 (2021), p. 110891.

[164] T. R. Vance and A. Vance, *Cybersecurity in the blockchain era: A survey on examining critical infrastructure protection with blockchain-based technology*, in 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), IEEE, 2019, pp. 107–112.

[165] D. Ventura, S. Monteleone, G. La Torre, G. C. La Delfa, and V. Catania, *Smart edifice,Äîsmart everyday interoperating future devices*, in 2015 International Conference on Collaboration Technologies and Systems (CTS), IEEE, 2015, pp. 19–26.

[166] J. Vora, A. Nayyar, S. Tanwar, S. Tyagi, N. Kumar, M. S. Obaidat, and J. J. Rodrigues, *Bheem: A blockchain-based framework for securing electronic health records*, in 2018 IEEE Globecom Workshops (GC Wkshps), IEEE, 2018, pp. 1–6.

[167] N. I. Voropai, I. N. Kolosok, E. S. Korkina, and A. B. Osak, *Issues of cybersecurity in electric power systems*, Energy Systems Research, 3 (2020).

[168] Y.-X. Wang, B. Balle, and S. P. Kasiviswanathan, *Subsampled rényi differential privacy and analytical moments accountant*, in The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 1226–1235.

[169] Q. Wen, X. Dong, and R. Zhang, *Application of dynamic variable cipher security certificate in internet of things*, in 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, vol. 3, IEEE, 2012, pp. 1062–1066.

[170] J. Wilcke and F. Lange, *Go-ethereum private network*, 2017.

[171] G. Wood et al., *Ethereum: A secure decentralised generalised transaction ledger*, Ethereum Project Yellow Paper, 151 (2014), pp. 1–32.

[172] G. Xu, C. Shen, M. Liu, F. Zhang, and W. Shen, *A user behavior prediction model based on parallel neural network and k-nearest neighbor algorithms*, Cluster Computing, 20 (2017), pp. 1703–1715.

[173] L. Xu, T. Bao, and L. Zhu, *Blockchain empowered differentially private and auditable data publishing in industrial iot*, IEEE Transactions on Industrial Informatics, (2020).

[174] J. Xue, C. Xu, and Y. Zhang, *Private blockchain-based secure access control for smart home systems*, KSII Transactions on Internet and Information Systems (TIIS), 12 (2018), pp. 6057–6078.

[175] D. Yang, J. Gavigan, and Z. Wilcox-O,ÄôHearn, *Survey of confidentiality and privacy preserving technologies for blockchains*, R3 Technical Paper, (2016).

[176] M. Yang, A. Margheri, R. Hu, and V. Sassone, *Differentially private data sharing in a cloud federation with blockchain*, IEEE Cloud Computing, 5 (2018), pp. 69–79.

[177] Q. Yang and H. Wang, *Privacy-preserving transactive energy management for iot-aided smart homes via blockchain*, IEEE Internet of Things Journal, (2021).

[178] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, *Integrated blockchain and edge computing systems: A survey, some research issues and challenges*, IEEE Communications Surveys & Tutorials, 21 (2019), pp. 1508–1532.

[179] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, *Privacy risk in machine learning: Analyzing the connection to overfitting*, in 2018 IEEE 31st Computer Security Foundations Symposium (CSF), IEEE, 2018, pp. 268–282.

[180] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, *A survey on the edge computing for the internet of things*, IEEE access, 6 (2017), pp. 6900–6919.

[181] M. Yutaka, Y. Zhang, M. Sasabe, and S. Kasahara, *Using ethereum blockchain for distributed attribute-based access control in the internet of things*, in 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–6.

[182] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, *Smart contract-based access control for the internet of things*, IEEE Internet of Things Journal, 6 (2018), pp. 1594–1605.

[183] Y. ZHENG, H. DUAN, AND C. WANG, *Learning the truth privately and confidently: Encrypted confidence-aware truth discovery in mobile crowdsensing*, IEEE Transactions on Information Forensics and Security, 13 (2018), pp. 2475–2489.

[184] Z. ZHENG, S. XIE, H. DAI, X. CHEN, AND H. WANG, *An overview of blockchain technology: Architecture, consensus, and future trends*, in 2017 IEEE international congress on big data (BigData congress), IEEE, 2017, pp. 557–564.

[185] Y. ZHOU, M. HAN, L. LIU, Y. WANG, Y. LIANG, AND L. TIAN, *Improving iot services in smart-home using blockchain smart contract*, in 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, 2018, pp. 81–87.

[186] Y. ZHU, Y. QIN, Z. ZHOU, X. SONG, G. LIU, AND W. C.-C. CHU, *Digital asset management with distributed permission over blockchain and attribute-based access control*, in 2018 IEEE International Conference on Services Computing (SCC), IEEE, 2018, pp. 193–200.

[187] Y. ZHU AND Y.-X. WANG, *Poission subsampled rényi differential privacy*, in International Conference on Machine Learning, PMLR, 2019, pp. 7634–7642.