Secured Internet of Things (IoT) Model using Blockchain

by

Abdul Muneem Khan
16101285
Fuad Hossain Oni
18241017
Jabed Omar
16201067
MD. Abdullah Al Noman
16201041
MD. Borhan Uz Jamil
16201026

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
BRAC University
December 2020

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at BRAC University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Abdul Muneem Khan
16101285

Fuad Hossain Oni
18241017

Jabed Omar
16201067

MD. Abdullah Al Noman
16201041

MD. Borhan Uz Jamil
16201026

Approval

The thesis titled "Secured Internet of Things (IoT) Model using Blockchain" submitted by

1. Abdul Muneem Khan (16101285)
2. Fuad Hossain Oni (18241017)
3. Jabed Omar (16201067)
4. MD. Abdullah Al Noman (16201041)
5. MD. Borhan Uz Jamil (16201026)

of Fall, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on January 11, 2021.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain
Assistant Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
BRAC University

Departmental Head:
(Chair)

Dr. Mahbub Alam Majumdar
Professor and Chairperson
Department of Computer Science and Engineering
BRAC University

Abstract

Anything which is connected to the internet is prone to threats. The internet has a saying that, "There are two types of hacked devices. One which is hacked, and the other one is which does not know that it is hacked". Security and privacy of valuable data have been a major issue over decades. Researches based on IoT device security is going around for quite a good time in this era and still has many scopes to build a more secure, data, and privacy protected IoT system. As beneficial as internet-connected devices are, they create some significant challenges. While we assume that these devices are protected with the same security level as the typical network server, it is not the case. IoT devices present some major security concerns, which we will talk about in broad. Blockchains can be a solution for securing IoT devices. Blockchain is a cryptographically secured, distributed ledger technology that allows for secure data transfer between parties. Blockchain is surely one of the promising and revolutionary technologies of this era because it reduces risk, stamps out fraud, and brings transparency in a scalable way for many uses.

Keywords: IoT; Blockchain; Ethereum; Security; IoT Security; Mining; Proof of Work; Smart Contract; Cryptography; Hash Functions, SHA256;

Dedication

All praise to the Almighty Allah for keeping us safe during the global pandemic. We dedicate our thesis to the late victims of COVID19. May their souls rest in peace.

Acknowledgement

We thank the Almighty Allah for whom we could complete our thesis with fewer interruptions.
Secondly, we are thankful to our parents for encouraging us to become what we are today. The path of getting an engineering degree was not possible without their help.
Lastly, we also thank our Supervisor Dr. Muhammad Iqbal Hossain Sir for providing a proper guideline to conduct our research. Without his directions, we are just a ship without radar.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ABI Application Binary Interface

$COVID-19$ Coronavrus Disease 2019

dApps Decentralized Apps

$DDoS$ Distributed Denial of Service

DoS Denial of Service

ECC Elliptic Curve Recovery

EPB Ethereum Private Blockchain

EVM Ethereum Virtual Machine

$FOTA$ Firmware Over-the-Air

IoT Internet of Things

IPC Inter Process Communication

IT Information Technology

$M2M$ Machine to Machine

$MITM$ Man In The Middle

$P2P$ Peer to Peer

$PBFT$ Practical Byzantine Fault Tolerance

PoA Proof of Authority

PoS Proof of Stake

PoW Proof of Work

$RFID$ Radio-frequency identification

RSA Rivest–Shamir–Adleman Crypto Algorithm

$SHA3$ Secured Hash Algorithm 3

Chapter 1

Introduction

The world is getting smaller and smaller in terms of connectivity. Thanks to the internet, everything is now connected. From the smartwatch in our hands to the fridge in our home, everything can be controlled by the internet now. The Internet of Things can be connected to the internet and can communicate with other internet-connected devices. As the internet of things can communicate with each other, sensitive data might be easily breached via a miscommunication or other means. IoT security is a much-underrated topic, which is becoming very important day by day. Blockchain is a digital ledger technology that ensures integrity and can be used in various fields. Blockchain uses a linked list like data structure but uses cryptocurrency and many other techniques to keep chain data secure. Blockchain is the most transparent and immutable or unchangeable way to store data to date.

1.1 Motivation

Nothing is done until we can feel the actual need for it. From our months of research, we have found out that though IoT will take over the world, there are some obstacles. Almost in all the papers, security was a major concern where protecting privacy has become an alarming issue. Moreover, in different layers of any system, we have got that the network layer is most under threat. As everything is connected to the internet, so it has become really easy to invade it, which is a major security concern. Preventing these kinds of invasions became our major motivation for what we started to think of something that could stop such threats. After researching and comparing for a while, we found out that blockchain is the most convincing way to do that. In the future decades, blockchain will surely play a major role to secure IoT based applications. Therefore, securing the IoT devices, protecting user's privacy, and preventing invaders from controlling the device are our major motivations behind using blockchain to secure IoT based systems.

1.2 Problem Statement

After the invention of IoT, its rapid growth of using clearly says how the world is being dependent on it day by day. For the increasing rate of IoT now, security has become the most concern issue for us. Millions of data are generated on the world by these devices in each second and stored in a centralized server for simplic-

ity. In contrast, this method can be vulnerable because of IoT connectivity with many devices. Insecure home thermostats, hacked baby monitors, Hackable medical devices, hacked autonomous cars, etc. are examples of real-life problems that we can face if IoT data gets breached. [12] We have discussed blockchain technology is the most secure technology till now. Still, we could not find any proper system where blockchain is implemented in IoT and made IoT more secured. We want to develop a model where blockchain can be used in IoT devices and make IoT data fully secured so that we don't have to suffer like the problems stated above.

1.3 Research Objective

Blockchain technology is currently one of the best technologies we have to keep our data secured and maintain privacy. To get or manipulate the stored data from a blockchain, a hacker would have to control and manipulate the data stored on every user's computer in the blockchain network or make a 51% attack. [16] There could be hundreds or thousands of computers, with each one saving a copy of a portion of blockchain data or all of the data. So, the hacker has to simultaneously bring down an entire network to take control of a blockchain, Blockchain would continue running to verify and record all the data on the network. The possibility of taking down a whole chain decreases along with the number of users on a network. The bigger the blockchain network is, the risk to get attacked by hackers becomes lesser, because of the complexity required to penetrate such networks. This complex configuration gives blockchain technology the ability to be known as the most secure form of storing and sharing information online maintaining confidentiality and integrity.

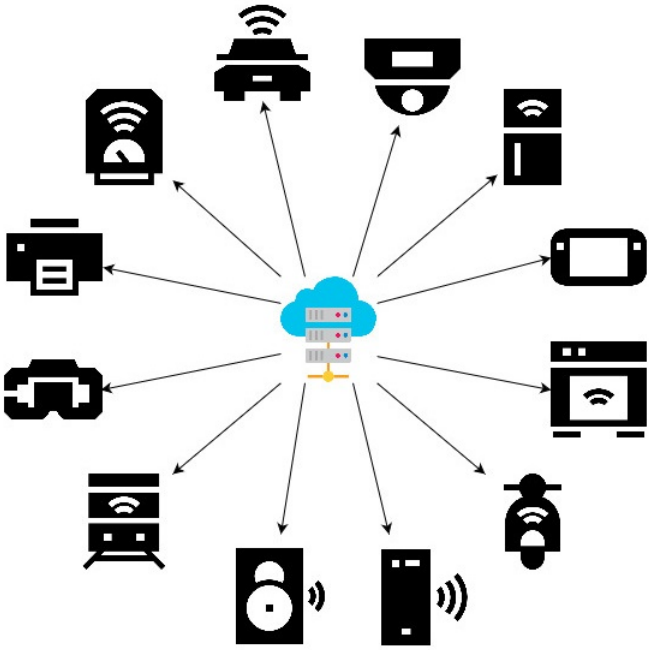

Figure 1.1: Conventional IoT System

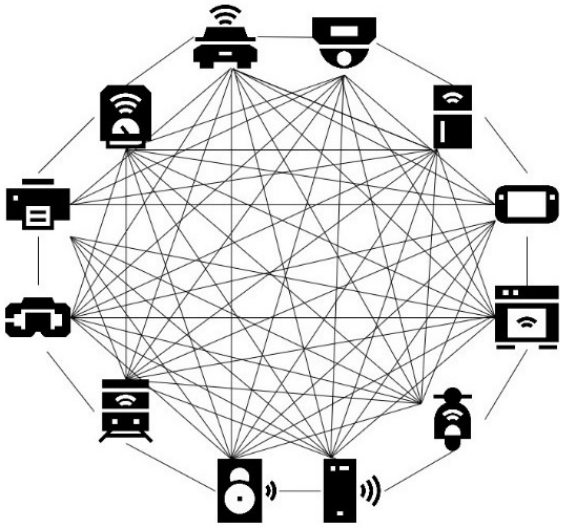

Figure 1.2: Blockchain-based IoT System

The IoT sector is one of the most fast-growing sectors in the IT industry, generating huge data. Figure 1.1 above shows the data transmission between a centralized server and IoT devices. This is the present or conventional system, which is also known as a centralized IoT system. A centralized IoT system usually has a server or a cloud where the data are saved and fetched from. Figure 1.2 shows a decentralized and blockchain-based IoT system in which all the IoT devices will be connected. There is no need for a centralized server in the second model as they are already inter-connected. Data generated from these IoT devices can be hazardous in the wrong person's hand. Still, securing IoT data is an important thing we often forget. We propose a model that merges blockchain technology with IoT devices to ensure user data safety and prevent stealing or compromising data.

1.4 Thesis Outline

This report focuses on building a blockchain-based IoT model that will improve device security and protect privacy. The authors aim to introduce such a model that, if implemented fully, can provide better security than the current conventional IoT model. The report describes the steps and methods that were followed by the authors.

At first, the 'Introduction' section, (Chapter 1) describes the motivation behind the study which motivated the authors to research this subject and work for that particular problem statement. The goal of our research is also briefly discussed here.

In the 'Background' part, (Chapter 2) we have widely discussed the IoT system, Blockchain, and some important related definitions which are used later in the report. We addressed some similar paper from Computer Science background which have related issues.

In the 'Proposed Model' section, (Chapter 3) we described briefly our proposed blockchain-based IoT model. We also showed a flow chart of the model and a

sequence diagram for a better understanding of the model at this part.

Then in the section 'Experimental Progress' (Chapter 4), we have described the configuration of the machine we used to implement this model, implementation experiment of the model, and testing experiment. All of this is stated step by step in this section of the report.

In the 'Result Analysis' (Chapter 5) section, we briefly evaluated our model for the Speed, Storage, and Cost factor. This section also states some advantages of our model for choosing the particular blockchain type.

Chapter 2

Background

2.1 IoT

The Internet of Things (IoT) is a bunch of devices that are connected to the Internet. It is normal to think about a laptop or a smart TV as an IoT device, but IoT incorporates more than that. Maybe copy machines, refrigerators at home, or the coffee pot in the breakroom connected with the internet. Internet of Things refers to any device, even those out-of-the-ordinary devices that can connect to the Internet. Most of the things with an on/off switch of today can connect to the Internet, making it a part of the IoT. Before 2019, the number of active IoT devices was estimated at 8.3 billion, but after 2019 the real number was around 9.5 billion. [18] But at the end of 2020, it is estimated that the number of IoT devices will cross 50 billion. [21] This shows how rapidly this field is growing and the importance of IoT device's data security. IoT protection is the field of technology concerned with protecting any devices that are connected to the internet in the internet of things (IoT). The safety of IoT devices and maintaining end-to-end security are hampered by many challenges. As the concept of network-connected devices and other artifacts is relatively recent, during the design process of a product, protection has not always been considered a top problem. Besides, since IoT has an emerging market, many product designers and manufacturers are more interested in rapidly selling their products rather than taking the appropriate steps to ensure protection from the beginning. IoT devices are often resource-constrained and do not provide the required computational resources for strong protection to be enforced. Many devices are thus unable to provide advanced security features. For example, because of its basic nature, a sensor that tracks humidity or temperature cannot handle advanced encryption or other security measures. Also, most IoT devices, put in the field or on a computer and left until the end of life, are "set it and forget it" kind. They hardly ever receive updates or patches on defense. From the point of view of a developer, building protection may become expensive from the outset, slow down construction, and cause the system not to operate as it should. We can also understand why there is a great need for IoT protection today.

IoT typically involves embedded devices with resource constraints, such as RFID and sensor nodes. The promises of traditional IoT devices include low memory, low processing capacity, and low battery life. Traditional networks, however, consist of powerful machines, servers, and smartphones with plenty of resources. There-

fore, conventional networks can be defended without any resource consideration by complex and multifactor security protocols. Contrary to this, IoT systems require lightweight security algorithms, such as battery life, memory, and processor use, that should maintain a balance between security and resource consumption. IoT devices, such as 802.15.4, 802.11a/b/g/n/p, ZigBee, NB-IoT, etc., often link to the internet or gateway devices via low bandwidth and low power wireless networking media. In conventional IT networks, however, end devices communicate through more secure and faster wired/wireless media, such as fiber optics, DSL/ADSL, WiFi, 4G, and LTE. Another distinction is that conventional network devices have almost the same OS and data format, but different data content and formats are available in the case of IoT due to application-specific features and lack of OS. Therefore, it is difficult to establish a standard security protocol that suits all types of IoT devices and systems because of this diversity. As a result, there is still a wide variety of IoT threats that endanger users' protection and privacy.

2.2 Blockchain

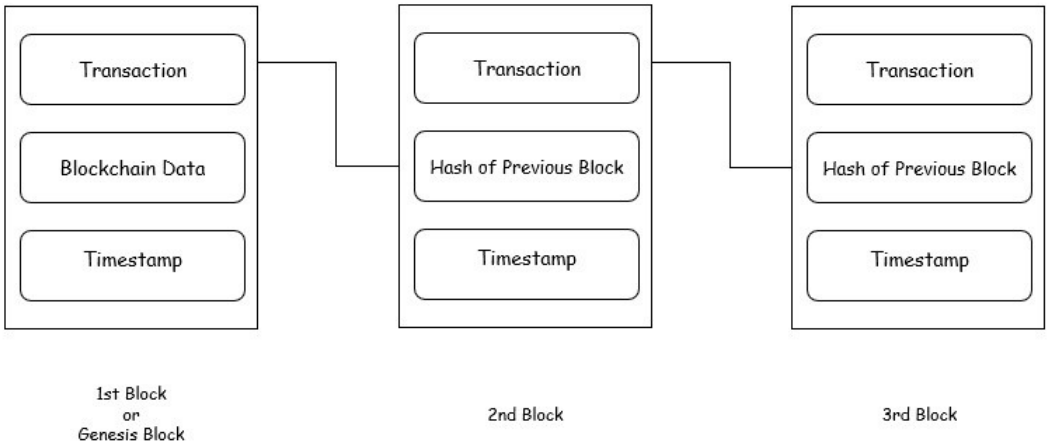

Figure 2.1: Concept of Blockchain

Figure 2.1 shows that blockchain is a structure that in many databases, known as the "block" stores transactional information, also known as the "chain," of the public in a network linked by peer-to-peer nodes. The storage is often referred to as a "digital ledger". Each exchange in this record is authorized by the advanced mark of the proprietor, which validates and defends the exchange from alteration. So, the data stored in this digital ledger is highly secure. Blockchain is a decentralized ledger that has recently gained a lot of popularity and appeal.

But by one day, it has not become mainstream, mainly ensuring that Blockchain Technology has taken credibility to this certain degree. We understand, for instance, that record keeping of records and transactions is a vital part of the business. This knowledge is often taken care of in-house or transferred by an outsider such as brokers, investors, or legal advisors expanding time, expense, or both on the company.

This long phase is avoided by Blockchain and allows a quicker transaction, thereby saving both time and money. People may assume that the words Blockchain and Bitcoin are synonymous, but that is not the case in reality. The technology that supports different applications linked to different industries is Blockchain. Yet Bitcoin is a currency that relies on the safeguarding of Blockchain technology. In this digital world, Blockchain is a growing emerging technology with many advantages and potential reach.

There are two main forms of blockchains currently available. They are:

1. Public Blockchain: A public blockchain is permissionless distributed ledger authorization where anyone can join the transaction and participate. There is a copy of the ledger for every peer on a public blockchain. A public blockchain is a network that is accessible for all. Anyone can download the protocol and the network can read, write, or participate. E.g. Bitcoin, Litecoin, Ethereum, NEO.

2. Private Blockchain: A private blockchain is also distributed and decentralized, but one needs to get permission to join and participate in this blockchain. Both public and private blockchains are decentralized, but private blockchains are more centralized because of their permission system and use cases. Also, a private blockchain is faster than a public blockchain as it contains a smaller number of nodes. E.g. Ethereum, Multichain, Hyperledger Fabric, Corda.

Public Blockchain	Private Blockchain
Public Blockchain is mostly permissionless.	Private Blockchain is mostly permissioned.
Node ID's are not known.	Node ID's are known.
Not as much of data privacy.	Availability of data security options.
Low transaction throughput.	Higher transaction throughput.

Table 2.1: Difference of Public and Private Blockchain

2.3 Related Definitions

- *ABI:* Application Binary Interface (ABI) is the standard way of communicating with contracts in the Ethereum ecosystem, both from outside the blockchain and for contract-to-contract interaction, according to Solidity docs [20]. As defined in this specification, data is encoded according to its form. The encoding does not define itself and therefore involves the decoding of a schema.

- *Bitcoin:* As an open-source code, Satoshi Nakamoto introduced the idea of bitcoin and released it in January 2009. Bitcoin is a decentralized currency, meaning it is not administered by any power or entity. No one regulates it, in particular. That's why it is said that it is regulated by everybody who participates in the scheme, and at the same time, nobody does. Blockchain was first

developed as a bitcoin center section, which rendered it the primary computerized currency to take care of the double-spending problem without the need for power or central server confidant. Bitcoin is an open code that, instead of bills and coins, is known by ciphered and anonymous codes. It enables simple registration of all kinds of financial transactions as it uses peer-to-peer (P2P) technology. Bitcoin is the first blockchain technology development framework and, based on this, it is still used in several instances.

- *Consensus:* A consensus algorithm is a mechanism by which all the peers of the Blockchain network reach mutual agreement on the distributed ledger's current state. Agreement calculations thus achieve reliability in the Blockchain network and, in a disseminated computing condition, create trust between obscure companions. Essentially, the consensus protocol makes sure that the only version of the truth decided upon by all the nodes in the Network is any new block added to the Blockchain.

  (i) Proof of work (PoW): It is the agreement calculation in the Bitcoin organization. Proof of work is utilized to affirm exchanges and produce new blocks to the chain. In this agreement calculation, miners go up against one another to finish the exchanges on the hash calculation and get remunerated. The fundamental working standards of this calculation are a mixed-up numerical riddle and the possibility to solve the problems.

  (ii) Proof of Stake (PoS): To get rid of the problem of PoW's latency, high computation, and energy costs, the idea of Proof of stake emerged. PoS indicates that people with more things to lose are less likely to attack the respective network. So, an entity with the highest coin on stake, for example, the number of coins times the days, only can mine a new block. But once the miners claim their reward, the coinage balance is auto-reset so that other miners/stakeholders can also get the chance to mine a block and get rewarded.

  (iii) Proof of Authority (PoA): Proof of Authority is Based on PoS. PoA is developed as an alternative to PoW. It is implemented by Parity. The authorities are already selected and each authority is assigned to a fixed time slot in the proof of authority algorithm. In the fixed slot, only the pre-selected authority will be able to generate blocks. The authority is verified based on its true ID. So, instead of economic value at stake, PoA implies the miner's ID at stake. Thus, a validator who falsely conducting anything will be known to all in the network. PoA assumes that the authorities are 100% trusted, and thus, it is wise to implement it on a private blockchain or permissioned ledgers.

  (iv) Practical Byzantine Fault Tolerance (PBFT): In PBFT, there is one main node, and all the other connected nodes are known as backup nodes. The task of backup nodes is to agree on the present state of the system and verify any message passed in the network is not changed. The main node changes with time and in a provided algorithm such as round-robin selection. This consensus is labeled to be more effective than PoW regarding

latency and energy costs, but it can only function if the number of illegal or malicious nodes is up to 33% on the network. Practical byzantine fault tolerance is believed to be a costly protocol as the number of messages required to exchange between the nodes for consensus.

- *Coinbase:* Every Ethereum node saves the mining rewards to an Ethereum wallet or address. That wallet is called Coinbase. Usually, it is the first account on the node, but it can be changed.

- *dApps:* dApps is the full form of decentralized apps. They are like normal apps and offer similar functionalities, but the key difference is they have no centralized server, they run on a peer-to-peer network, such as a blockchain. That means no one person or entity has control of the network.

- *Embedded Software:* Instruction code that runs on hardware microcontrollers. Normally, it is performing explicit low-level functions regularly without utilizing an operating system. Embedded software is specialized for the particular hardware it runs on. It often has time and memory constraints, which must be addressed in IoT devices. Most IoT gadgets influence implanted programming, taking more time to compose than more abstracted server-side code.

- *Encryption:* Encrypting some data means hiding it so that it can only be read-only if the user has a valid password or code. Encryption is one of the basic features which makes blockchain immutable. Some of the encryption techniques are SHA256, RSA, AES, DES, etc.

- *Ethereum:* Ethereum is a decentralized platform and a type of blockchain that enables creating "smart contracts". It was originally considered as a better version of cryptocurrency to overcome the limits of Bitcoin. It arranges data as the normal blockchains but Ethereum can execute the smart contract and that feature of Ethereum can be used in various applications in many fields.

- *Ethereum Account:* All EVM can open an unlimited number of accounts. A combination of Ethereum address and its private key is mainly referred to as an account. An account is mandatory for holding transaction to smart contracts. Ethereum account is often called an Ethereum wallet.

- *Ethereum Address:* Every account in EVM has two unique and personal Ethereum addresses. As the name suggests, the public address or Ethereum address is known to others for signature verifications, and only that node knows the private address for other purposes.

- *EVM:* Full form of EVM is Ethereum Virtual Machine. Each node connected to the Ethereum network is called an Ethereum virtual machine.

- *Firmware Over-the-Air (FOTA):* FOTA is a technology that enables suppliers to patch bugs or remotely install new software features wirelessly after product distribution. It is an efficient way to upgrade and update a device remotely. Manufacturers can save resources on effective and opportune overhauls without having physical admittance to the gadget.

- *Genesis block:* Genesis block is the first block on any blockchain protocol. It contains the needed data like difficulty level, gas price, network id, pre allocations of tokens or coins, etc. to run the blockchain. As the structure says, every block on the blockchain contains the hashed value of the previous one. Genesis block also has a previous hash value, which is basically "0x00" because it is the first block.

- *IPC:* Inter-process communication usually works on a local computer or a single computer. It is a set of techniques for exchanging data among multiple threads in one or more processes.

- *Machine to Machine (M2M):* Interconnected devices trading information with each other, without human assistance. Machines monitor other machines without the need for human involvement. For instance, a machine can alarm when another part is required or separated, dispensing with manual observing, which gobbles up important time and assets.

- *Miners:* Mining is the process to launch new cryptocurrencies in the market. People who are in charge of mining are called miners. Miners work all day and night with powerful computers connected to the blockchain, ensuring that all the transactions are performed correctly.

- *Nodes:* The computers that are part of the blockchain network are called nodes. They store and distribute blockchain data continuously. Nodes check if a block of a transaction is valid or not and thus accepts or rejects it. Nodes also broadcast and spread transaction history to the other nodes that might need to synchronize with the current blockchain status. All miners in the blockchain are nodes, but all the nodes are not always miners. There can be different types of nodes in a blockchain network depending upon their capabilities and resources such as computation capability and memory size.

  (i) Simple/Normal Node: Simple node only sends and receives a transaction. It also does not hold all of the data in the blockchain rather it holds the most recent part of a blockchain. Regarding the IoT environment, Sensors or displays can be an example of a simple node.

  (ii) Full Nodes: Full nodes handle a total copy of Blockchain, but they have no authority to mine a block. Although, full nodes approve transactions dependent on the agreement or consensus rules of the particular blockchain

and contribute to adding or forking out a block. A double-spending or a malevolent transaction may not be permitted or transferred by a full-node. Full nodes are equipped for transactions. Therefore, full nodes are fundamental for the security of the blockchain. In an IoT system, a Raspberry Pi with more computational and memory assets in contrast to an Arduino can be a full node.

(iii) Miner/Validator Nodes: Miners are the full node that has the extra ability to mine or approve another block hence expanding the blockchain. Additionally, mining nodes are chosen according to explicit rules dependent on the kind of agreement convention known as the consensus protocol. E.g., In Bitcoin, the mining nodes need to address a cryptographic riddle, and the node that does it initially is qualified to mine the block. The miner node needs to present a Proof of Work (PoW) alongside the mined block so the remainder of the nodes can approve that the riddle has been effectively resolved. Now if that the block is acknowledged by the remainder of the network, the miner node at that point obtains a mining prize in form of transaction charges as particular cryptographic money. While, in the Proof of Stake (PoS) consensus protocol, miner or validator nodes are chosen rapidly dependent on the coinage, i.e., the number of coins they own and the time since they have those coins. Again, we can see in the vast majority of the (Byzantine Fault Tolerance) BFT-based consensus protocols, the validator is chosen in a round-robin to mine another new block. The remainder of the nodes, vote on the legitimacy of the block and its transactions. In most cases, the block is approved and added to the blockchain after getting 2/3 majority share votes in support of it.

- *Remix:* Remix is an online solidity-based IDE used to write, compile, and debug smart contracts.

- *Smart Contracts:* A smart contract is a kind of contract that can be self-executed with the terms of the agreement between two parties being directly written into lines of code. The code and the agreements in a smart contract exist across a blockchain network. The code controls the execution. And transactions called by smart contracts are trackable and irreversible. [17] Smart contracts' goal is to decrease the need for trust in mediums, authorization costs, fraud losses, dangerous and unplanned impunity of connectivity.

- *Solidity:* According to the tutorials point, solidity is a contract based, a high-level programming language for implementing smart contracts. [22] Solidity is influenced by C++, Python, JavaScript, and is designed to be used in EVM.

- *Token:* Tokens are units of significant worth procured through blockchain and used to obtain products and services.

- *Transaction:* A request to change the condition or situation of blockchain is called a transaction. Usually, the transaction comes at a certain price and it

depends on the blockchain properties.

2.4 Literature Review

According to a research paper [1], IoT is typically structured into 3 basic Layers. And they are the Application layer, Network layer, and Physical layer. The physical layer can face Node Tampering, RF interface, Node jamming, Malicious node injection, Physical damage, Social engineering, Sleep deprivation attack, etc. The network layer is hampered by Traffic analysis attacks, RFID spoofing, RFID cloning, Sinkhole attack, Man in the middle attack, DOS, DDOS, Sybil attack, etc. And in the application layer, there are several attacks like viruses and worms, Spywares, Adware, Malware, Trojan horse, DOS, etc. They also stated some layer-wise mitigations to be safe from those threats. The secure boot of IoT devices, Device authentication using low power techniques, ensuring data confidentiality, and maintaining data anonymity can safeguard us in the physical layer or IoT devices. For the network layer, securing communication between the devices, implementing routing security, and securing user data on the device is mentioned in the paper to counter-attack incoming threats. Lastly, data security, ACLs, Firewalls, and protective software like antivirus or anti-adware can prevent threats in the application layer. Finally, regardless of any layer, there are some suggestions to counter-attack threats in all layers which are, finding new threats, applying updates and patches, providing improvements, upgrading systems, using IDS (Intrusion Detection System) in the device, Securing IoT physical premises, monitoring devices, Trust management, etc.

In another research paper, [6] Sensor-based threats in IoT devices can be categorized into four broad categories based on the purpose and nature of the threats: Information Leakage, Transmitting Malicious Sensor Patterns or Commands, False Sensor Data Injection, and Denial-of-Service. Information leakage can be divided into few parts such as Keystroke Inference using Light Sensors, Motion Sensors, Audio Sensors, Video Sensors, Magnetic Sensors, Task Inference using Magnetic Sensors, Power Analysis, Location Inference, and Eavesdropping. Transmitting of Malicious Sensor Patterns or Commands can occur via Light Sensors, Magnetic Sensors, Audio Sensors. It also explained some existing security mechanisms to prevent sensor-based threats like Enhancing Existing Sensor Management Systems and Protecting Sensed Data. They provided some excellent future scope of their research, which included Study of Expected Functionality to Identify Threats, Control Sharing of Data among Sensors, Protect Sensor Data when at Rest, Prevent Leakage of Secret Data, Protect Integrity of Sensor Operations, and Adoption of Intrusion Mechanisms to Detect Attacks.

In one of the papers, [7] Blockchain is described as a game-changer to secure IoT data. They have provided a more secure and trustable Internet of Things model using blockchain. For blockchain-based IoT, several patterns are used. One of them is the communication model described in their paper. In this model, mainly three of the fundamental functions of a blockchain network are used, and they are peer-to-peer messaging, distributed data sharing, and autonomous coordination with the device. To strengthen IoT security with blockchain, the pillars are fundamen-

tal. Blockchain has four pillars, and they are consensus, ledger, Cryptography, and smart contract. Consensus preserves the sequences of transactions and allows access control at the level of a transaction. Also, the ledger records transaction detail due to which third party necessity is not needed. Moreover, Cryptography blinds the data with the unyielding crypto mechanism. Lastly, the smart contract verifies possession of the private key and verifies whether the message sender is a valid user or not. It also detects the message's integrity. There are also some requirements to strengthen IoT, which is possible with blockchain. The requirements are secure communication, authentication of users, discovering legitimate IoT at a large scale, and configuring IoT.

One more paper, [14] describes a simple, secure smart home system based on a refined version of blockchain called Consortium blockchain. When a user requests for door opening, it will check this request, whether it is a valid request or not. Via the internet, this request will go to Super Node, where first it will check the security implementation process on the incoming request. Then it will go to Blockchain Ledger if found, then generate block transaction if no, generate a new blockchain ledger. After that, it will go to broadcast a new block to the sensor then the P2P server. If the broadcast is yes, then it is valid, or it will send a response with an error. Lastly, after checking validity, if the Sensor target reference is yes, then the action is fulfilled or send an acknowledgment to the reference sensor. However, Current research clearly shows that using blockchain itself is a challenge as it is complex to implement, and the solution which is based on smart contract possibly can increase the system cost that motivates to simplify the blockchain implementation for smart homes. In conclusion, this article investigated the previous work by presenting a simplistic model to implement a secured architecture that utilizes a polished version of the blockchain. There have been made some significant development in this research like participation of pre-selected nodes in blockchain creation and consensus, Communication between sensors through mesh network topology, Supernode registering and authorizing the admin user via REST API, providing a private mechanism for the user's authorization and authentication, applying initial security checks to ensure confidentiality and integrity.

And in paper [11], the paper discusses the techniques and security in online transactions, vitality exchanging, digital money/cryptocurrency, the industrial internet of things, and so forth. Network layer security is essential while working over the internet. In any system, data security is essential against assailants and programmers. To resolve this problem, the first step is to secure the data from unauthorized access and protect it from thwart hackers. It stores all information in blocks associated together in a sequential way to make a continuous line. In this paper, blockchain is used to keep transaction data safe. Blocks containing transactions will be affirmed by explicit blockchain nodes. There are some challenges and limitations also, such as for conducting these procedures, energy level and cost is very high, it is a rising technology so many people do not know about blockchain. Thus, lack of awareness is another thing. To conclude, though it is not that much popular technology for ensuring security, it guarantees identity, privacy, and transaction security, which are the primary issues for information security, and thus it plays an imperative role in ensuring data transparency.

IoT technology is mostly used for a smart home system, but there is some obstacle regarding the security issue of data. One of the information that should be secure is the entryway/door lock information access. This information should not be defenseless against duplicating and hacking because an entryway/door is sincerely identified with the security of the property holder. Door lock access information can be utilized to discover who and when somebody is in or out of the house. Hence, we can expand home security if something doubtful occurs. One paper, [10] proposed a door lock system that will be secured by the Ethereum blockchain. Moreover, some other researchers mentioned the same kind of framework. For ensuring who will enter or exit from the house they proposed door lock system used a webcam to do face recognition. Webcam catches the face before it. At that point, the webcam will perceive the face, if it is enrolled. On the off chance that the face is as of now enlisted as the proprietor of access rights, the situation will check whether the webcam is enrolled on the blockchain network or not. The paper also clarifies transaction handling, which is using smart contracts to set methods made by the property holders. On the private blockchain, a miner is pre-selected to keep up the blockchain network. All the transaction data of the door lock system will be stored in a private blockchain. The proposed framework design comprises of nodes (webcam and mortgage holder), a miner, and a private blockchain, which is Ethereum blockchain, alongside the smart contract. The paper also shows the experimental results which were performed on the system, and there was some significant success as well.

One paper, [8] talks about through Ethereum contracts how to implement the Access Control List. IoT needs security proficiency, and to improve IoT security, we need to defeat hardware constraints, for example, heterogeneous abilities in processing assets. The particular system gives a safe environment for information trade among members and secure information storage. And by utilizing IoT with a PoA (Proof of Authority), solves the problem of solving complex mathematical riddles. They designed Geth for Raspberry Pi as an Ethereum light-node. They use Ethereum Harmony, which depends on Ethereum Protocol and this product permits us to control the environment of the network where we can add smart contracts, start the mining cycle, convey the among nodes, and add nodes to the Blockchain network. The Access Control List exists inside the Blockchain, and it will work as a smart contract where the ACL will store the MAC address, IP address, Encode of the device, and the ACL number. In the software implementation part, they address the implementation of Go Ethereum utilizing Proof of Authority consensus. Nodes inside the network will be having a similar access control list of smart contracts. consequently, Proof of Authority nodes shall keep up and control the nodes in the ACL smart contracts. Their paper gives various usage in various consensus algorithm, and they accepted that their upgrade improves IoT security, considering the suggested procedures applied.

The authors of a paper [9] were seemed to take benefit of the distributed quality of blockchains to set up a huge IoT control system and they prioritized smart contract-based tokens to implement a better access control mechanism. Keeping that under consideration they constructed a blockchain-based system that permits users to run and control IoT devices arranged in “groups” (e.g., turn on the lights of a

smart city). This system is constructed using the Ethereum blockchain, takes the limitations under consideration and capabilities of the IoT devices, and also the required items of the blockchain technology. They built a blockchain-based IoT design-oriented system on presently available technologies and defined its users and their internal communication where they build, implement, and verify an event-oriented IoT management solution dependent on Ethereum smart contracts.

There are various sectors in a smart home system where IoT slows down or does any job without giving priority to one job over another. But sometimes some emergency cases occur when the system has to give priority to a particular task. In such cases, these systems will dispatch an emergency call by themselves with the help of home user information and address as private information to public works such as police stations, fire service's offices, hospitals. This particular research paper [13] mentions and discusses immediate service when necessary for a Smart Home Technology dependent on Ethereum Blockchain with a smart contract for decentralized handling access control between unfaithful public assistances which are known as Home Service Providers(HSPs) and smart home IoT technologies. Here is needed an open-source platform that is compatible with decentralized application frameworks and also provides programmable smart contracts to trigger transactions automatically with specific conditions. As only Ethereum blockchain has all these functionalities, so it is chosen for the architecture. The most essential fact is the distributed data to all P2P nodes should be similar or identical, fixed, and pressurized. All these qualities should be in the blockchain network. A consensus algorithm is needed in a blockchain to gain a result on the rational state of the transaction between well spread P2P nodes. IPES and Ethereum were incorporated in well-spread miner systems in this paper. RSA asymmetric encryption algorithm which has a key length of 1024 bits was applied in the proposed work. Also, there is authentication which is a technique used to authenticate the user of IoT device identity to be able to read applications, computing systems, or resources. After that there is sensor manager installation and configuration and then comes the HSP's EM installation and configuration as well. The Ethereum package is acquired to configure the EM node and they have to make a particular information directory folder to store the wallet and database of the EPB. In the proposed system there are some kinds of transaction types like functions to supplicate reliability smart contract via JSON RPC. Finally, HSP's Ethereum Machine performance results are measured by running the mining techniques for each transaction on EM1 and EM2 multiple times. In this prototype, they applied a private Ethereum blockchain to maintain transactions between unfaithful or unworthy groups of users between Smart Home IoT systems and public service providers. So basically, the transactions between the Ethereum miners, home service providers, and homeowners were recorded and maintained by them. As future work, according to them, they will think of and implement if possible an access control mechanism all-time limitation QR code and design gas usage for all the transactions. They are also thinking of including the Ethereum wallet charge for emergency call service.

Network Address Translation (NAT) is the main problem that is stopping the large placement of the P2P routing which serves as a solution for the fatigue of IPv4 addresses. In this paper [3] the proposal is about a blockchain-based platform that

lets Traversal Using Relay NAT (TURN) servers run as relays for the Internet of Things (IoT) devices behind NAT. Blockchain technology delivers a network or node that is implemented by a consensus algorithm to accept the uprightness of the shared log. That also delivers End-to-End Security for limited or unlimited IoT devices. In this paper, they propose a field that makes P2P routing for IoT devices using blockchain technology possible. On the other hand, just by using a private IP address NAT lets devices on a local network connect to a stranger network from outside. Along with the pros, there are also some massive cons or limitations and drawbacks of using NAT in applications. Ethereum is well known as an open-source blockchain field that has smart contract-based scripting functionality. Also, a program or a transaction protocol that is made for executing automatically, control legally valid actions according to the terms of a contract is known as a smart contract. This is well distributed in the blockchain network. Any wallet containing sufficient ether and having a function dependent interface can deploy a smart contract. End-user devices can't participate in the system because of the limitation of storage and memory. The alternative to solve this problem is the Light node which can replace the Ethereum full node. For all Ethereum decentralized applications, the light chain is downloaded in the device or the system once. Merkle root is used by it for better and more secure verification of the necessary blockchain functions. Users of this system use blockchain as a well-spread database that can store all of their identity, Relayed Transport Address given by TURN and Public Key. There is a software component of this platform which has Wallet Management function (WM), TURN Servers, IoT Client-Module and Smart Contract. Moreover, TURN Servers and IoT Devices are the two major components of the configuration mechanism. Various technologies are implemented in the proposed system to verify it. To simulate the blockchain network they used a technology named the Ropsten Testnet. In the paper, they proposed a remarkably secure field dependent on Ethereum blockchain and public TURN servers so that the issue of NAT Traversal and End-to-End secure session setup can be solved. This specific solution can be applied to any type of NAT (including symmetric) because it works on all of them. This is quite possible as it is based on every type of IoT devices and TURN. In the future, they are thinking of recreating this system using the Port Control Protocol (PCP) so that they can replace TURN with it to solve NAT problems.

In one more paper [5], the authors propose an application dependent on blockchain technology which is mainly decentralized. It is for sharing sensor information based on the Internet of Things (IoT) and demonstrate some varieties of obstacles mentioned during the development process. The best part is blockchain technology and IoT are combined by this application and it is run by smart contracts that are implemented on Ethereum blockchain. Moreover, measurements of IoT weather sensors are shared by this application which works perfectly as a platform for this. It also functions on the Ethereum blockchain where it acts as a marketplace for IoT sensor data. Sensing-as-a-Service (S2aaS) business model combined with blockchain is applied by this particular application. A safer and user-friendly payment system competently combined with such applications is needed for the successful development of an S2aaS application. They created a decentralized S2aaS application that applies smart contracts in the Ethereum blockchain. Moreover, they developed a marketplace for IoT weather sensor information by using this application.

Operators of the sensor can register their sensors on the Ethereum blockchain and measurements of the registered sensors can be bought by the users as well. By using cryptocurrencies, they deliver the users with the capacity to execute P2P transactions on the Ethereum blockchain which is the utilization of blockchain technology. NTUA token (National Technical University of Athens token) is a custom experimental token that is used as a currency in the marketplace. It's used along with Ether which is mainly created specifically for research and experimentation purposes, to research as many characteristics of the Ethereum as possible. The ability to perform safe P2P transactions with cryptocurrencies delivered by Ethereum and the ability to use are the main benefits of the decentralized application. High transaction confirmation delay is a problem that should be changed in the future and difficulty to stop scams (a registered sensor that does not supply the information) are the most alarming issues of the specific application. As it is connected with the social engineering aspects, so the solution to this problem is out of the hands of the presently ongoing research work. They have a plan to expand the present research work to bring development in multidisciplinary aspects.

In a study [2], with and without using blockchain an IoT system is supposed to be developed and then make a comparison between two of the systems. MQTT is used as a communication protocol in the IoT system without the blockchain technology. On the other hand, the system developed with blockchain technology has used Ethereum combined with a smart contract. Both the IoT systems are supposed to be analyzed and tested for their safety level by perceiving their safety aspects and simulating attacks. The result was crystal clear as the IoT system designed with blockchain technology has a higher level of security than the IoT system designed without the blockchain technology. In recent years, the development of IoT technology has increased at a huge rate but has gone along with security problems. The insecurity of communication that happens between IoT devices is one of the common security issues that arise. In this particular research, the design and building up of the IoT system have been conducted with and without using blockchain technology to check and compare the results of the test and experiments. Like the previous paper, Ethereum is used as the field of a blockchain network in the IoT system. Along with the Ethereum to store and retrieve necessary information from the blockchain network, smart contracts are used as well. Calculating and checking all the results of various tests can prove that the IoT system developed using blockchain technology can prevent security threats that are seemed in communication between IoT devices as it has a higher level of safety measures than the IoT system developed without using blockchain technology. So by using blockchain technology in IoT systems data integrity can be well guaranteed. This can be observed and verified from testing of attack simulations and monetization of avalanche effects accomplished where the application of blockchain technology has better security in the IoT system.

Chapter 3

Proposed Model

In most of the IoT network, the devices which need to be connected to that device are known to us. For example, if we have a smart air conditioner, it will be connected to our thermostat to sense the temperature and control the air conditioner depending on that data; or if we have a smart water motor system in the reserve tank, the motor will be connected to some depth sensor in our water tank which will send a signal to turn the motor on or off. Our model will be a blockchain-based IoT model, there will not be any centralized server in it. Every device will be running a private and local blockchain instance which is by definition not accessible outside of the network and possibly the most secure system till now in terms of immutability. Each device will be called nodes. So, each device will have a unique Ethereum address. Also, Every IoT devices should be inter-connected by a peer to peer connection manually. We will store the Ethereum addresses of those devices on a smart contract, which we know that our node or device needs to be connected with to function, in the blockchain instance. In our system, devices or nodes which are not enlisted in our smart contract shall not be communicated by the nodes on our blockchain. This will ensure security from malicious attacks or commands to IoT devices. Features of our model are the following:

1. Only the Ethereum account owner on the device shall be able to add or delete a device to the blockchain network. We shall call the owner, 'Home miner'. In other words, only our home miner shall be able to add or delete nodes in the chain; thus, only the home miner can decide whether that particular node can communicate with another particular node or not.

2. We will use a local private Ethereum blockchain for implementing our system. Ethereum system has to be installed on all of the IoT devices which want to use this secured model.

3. 3 types of requests can be done by the home miner in our system. Add a device, delete a device, and the other is to check connectivity. As the name suggests, add or delete is requested to add or delete a new device on the network, which is only valid if our home miner requests.

4. Regardless of which request a device sends, it has to send a valid Ethereum account address to add or delete or check connectivity as well as any random message with the request. The message and the Ethereum address of the requester generates a signature that is encrypted, and our system gets the

signature and the hashed message as soon as the requester sends the message. Now our smart contract shall decrypt the signature and will get the Ethereum address of the sender.

5. Our system already knows the Ethereum address of our home miner. If any other node sends add or delete requests rather than our home miner, the system will not allow the request and submit a transaction containing the possible intrusion. If our home miner sends the request, then that address shall be checked for validity to add or delete and will simply be added or deleted from the network, and a transaction containing the address and information regarding the device shall be stored in the blockchain.

6. Check connectivity request is made when our home miner tries to check if another device is connected to our network or not. In other words, when an IoT device wants to know whether it will communicate to another IoT device holding the provided Ethereum address or not, it sends a connectivity check request. For this particular request, the smart contract first checks the requester's account address. If the address is matched with our home miner, that means the request is valid and our home miner wants to know if the address it sent should be communicated with or not. Then our smart contract checks if the requested address is enlisted in our list or not and sends the result true or false thereby. But if the sender is not our miner, that means someone from outside is trying to access our system, then the system will count this as an intrusion and take the provided steps.

7. The verification of the home miner is done using Elliptic-curve cryptography. It is a widely known algorithm to verify digital signatures [19]. Elliptic-curve cryptography (ECC) is public-key cryptography which is based on the algebraic structure of elliptic curves over finite fields. This method verifies the signature of a sender by using the public key. The signature can only be generated using the sender's private key.

Our model shall be a combination of IoT and Blockchain, which shall provide more secure communication between IoT devices. We have proposed and implemented the Blockchain part but this blockchain-based model needs to be implemented on the network layer or add a blockhain layer after the network layer of IoT architecture to provide security. It can be implemented in such a way that if an IoT device requests to communicate with another IoT device, the 2nd device shall check connectivity before actually communicating with the first device. If the first device is permitted to communicate or in other words, the address of that device exists on the connected device list, only then the second device will communicate with the first device.

A flow chart defining the basic functionality of our proposed model is given in figure 3.1 below:

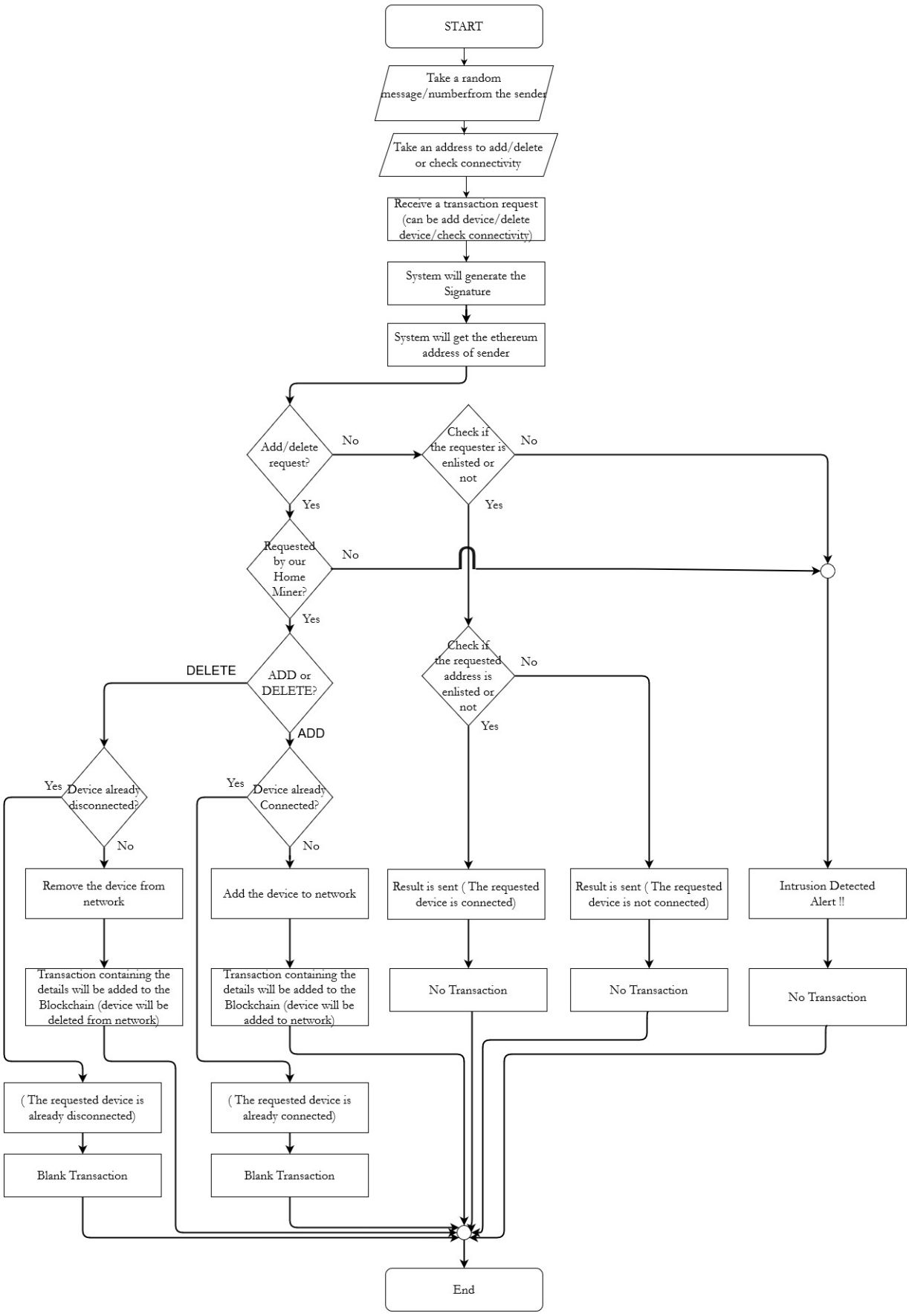

Figure 3.1: Flowchart for Blockchain-based IoT model

The sequence diagram for our model is given below:

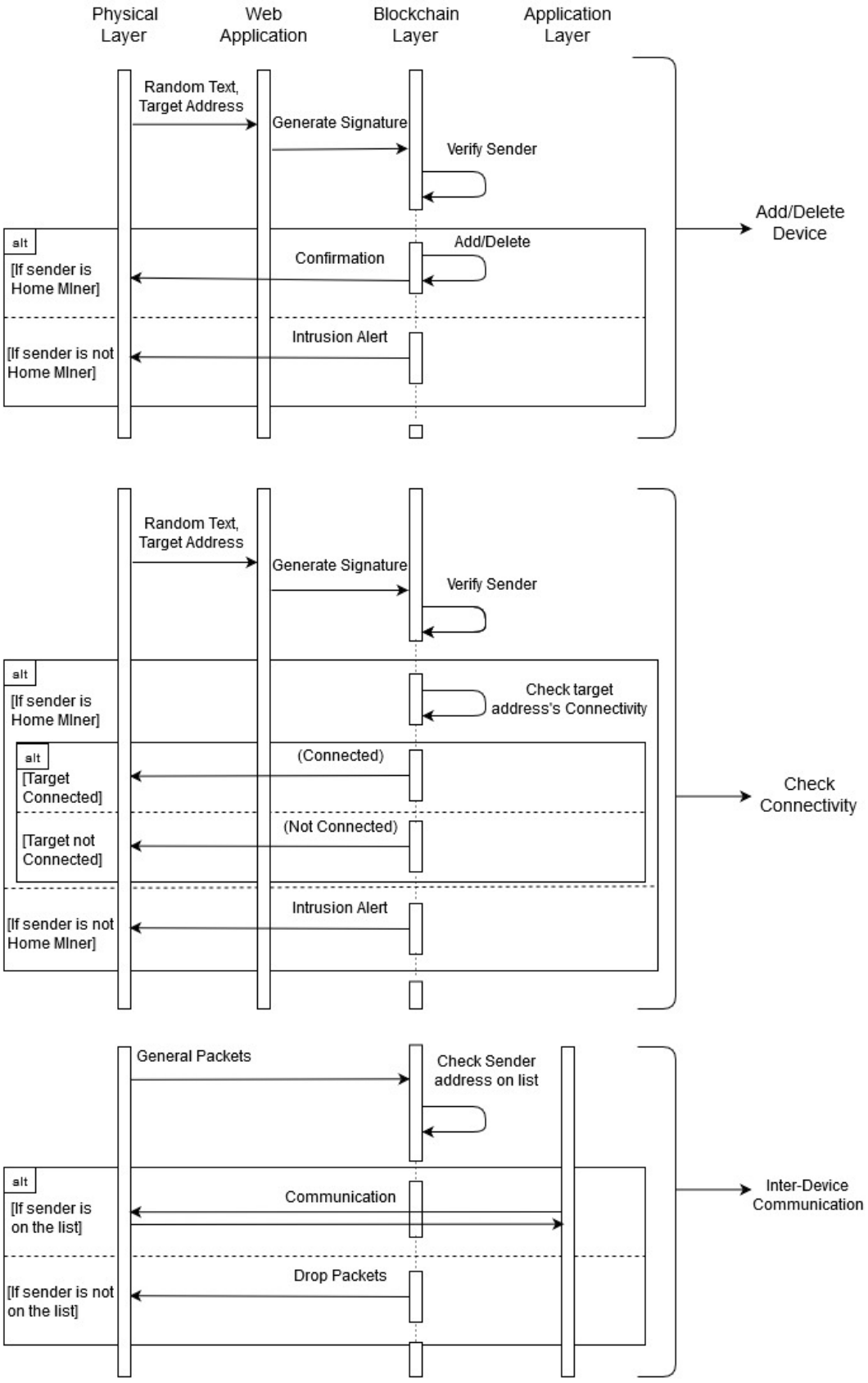

Figure 3.2: Sequence diagram for Blockchain-based IoT model

Chapter 4

Experimental Progress

4.1 Experimental Configuration

For the implementation of our model, we used a machine that has a processor of core i3 6th generation. This particular processor has 2 cores and 4 threads built into it. Our processor has a clock speed of a maximum of 2.3 GHz and has 3 MB cache memory. We used Ubuntu 20.04 LTS for implementation experiments. Also, our machine had 4 GigaBytes of DDR3 1600 MHz of physical memory or RAM installed. Table 4.1 shows the configuration of our system where we will implement our blockchain.

Processor Type	Intel 6th Gen Core i3-6100U Processor
Processor Speed	2.30 GHz, 3 MB Cache
Operating System	Ubuntu
Version	20.04 LTS
Memory	4 GB DDR3 1600 MHz

Table 4.1: Experimental System Configuration

4.2 Experimental Implementation of the Model

We can deploy a private blockchain on our local computer by creating a genesis block first. Then we can initialize and start our very own private blockchain. We could deploy our smart contract on our private blockchain and proceed further, but we used a Truffle framework named Ganache for easy processing. Ganache is a personal blockchain developed by Truffle which can be used easily to deploy or develop dApps. Figure 4.1 shows the startup of Ganache on our machine. After starting, Ganache gives us 10 Ethereum accounts, each loaded with 100 ethers running on a private blockchain.

As we have our blockchain running on Ganache, we need to deploy a smart contract to our blockchain. For that, we used the Remix IDE. We wrote our solidity code on the Remix IDE, compiled it, and then collected the ABI. Figure 4.2 shows how we have collected the Application Binary Interface. Then we deployed the smart

contract on our ganache blockchain. After deployment, we got a deployment address for our smart contract. Figure 4.3 shows the collection deployment address. The deployment address is needed to send the virtual ethers to the smart contract for the needed transaction.

We also designed a web application to communicate with our blockchain via smart contract. Figure 4.4 shows the initialization of the system. It is a simple representation of our planned IoT-Blockchain model. We can think of this page as a user interface of our proposed system. Our device has a unique Ethereum address and if any account without that address requests to manipulate data on our blockchain, it will be ignored. If we request to check connectivity to any address like figure 4.5 after initialization, as no device or address is saved on our blockchain, it will give us output that the requested address is not connected. If we try to add a device like figure 4.6 and give the proper inputs which contain a random message and the Ethereum address of the target device which we want to add, it will show us the output that the device has been added. Then if we request a connectivity check with a random message and that target address, we gave input before, we get output like figure 4.7 which ensures us that our addition of device addresses is working. Then suppose we want to delete a particular device from our network, we request to delete it like in figure 4.8. We can again ensure that the particular device/Ethereum address is not on our trusted device list by checking the connectivity like figure 4.9.

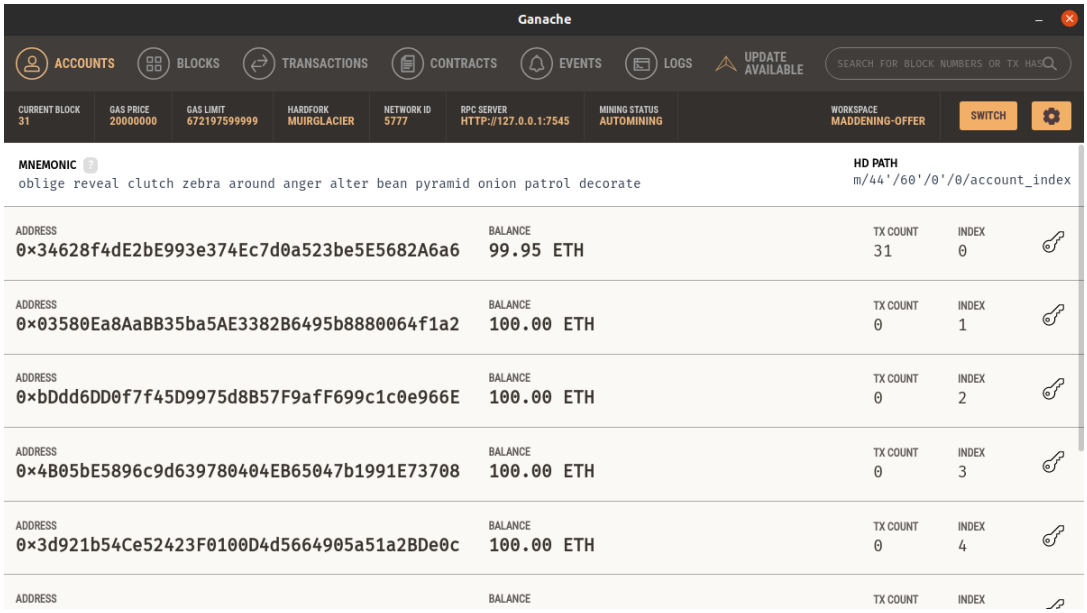

Figure 4.1: Starting Ganache UI

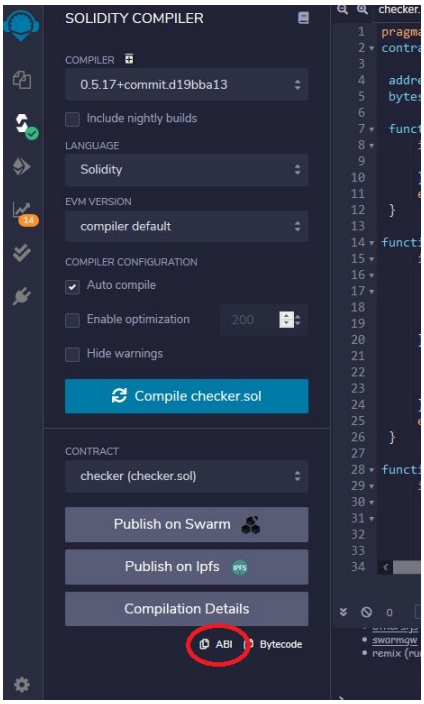

Figure 4.2: Collecting ABI from Smart Contract

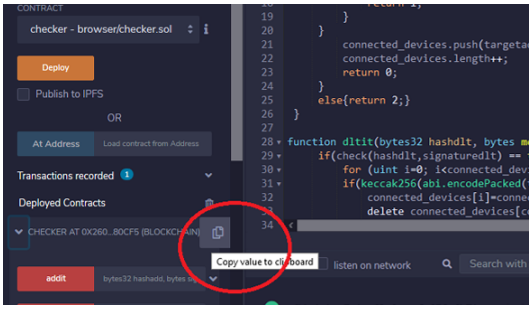

Figure 4.3: Collecting Smart Contract Address after Deployment

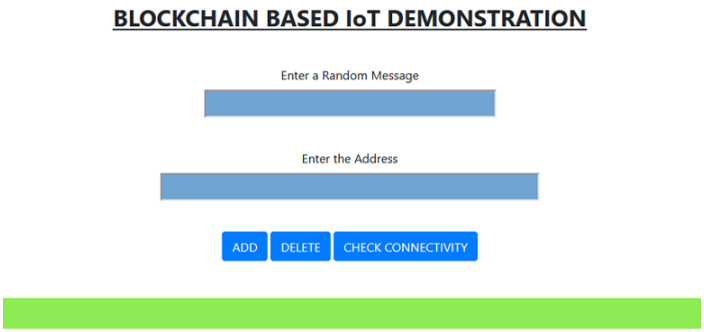

Figure 4.4: Initialization of the System

Figure 4.5: Checking Connectivity

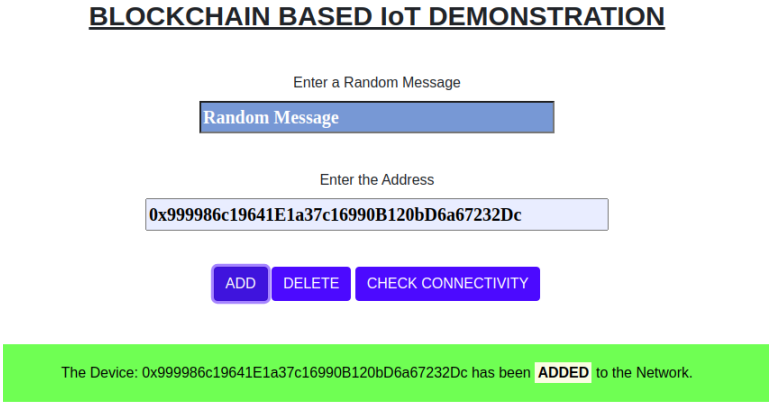

Figure 4.6: Adding a device

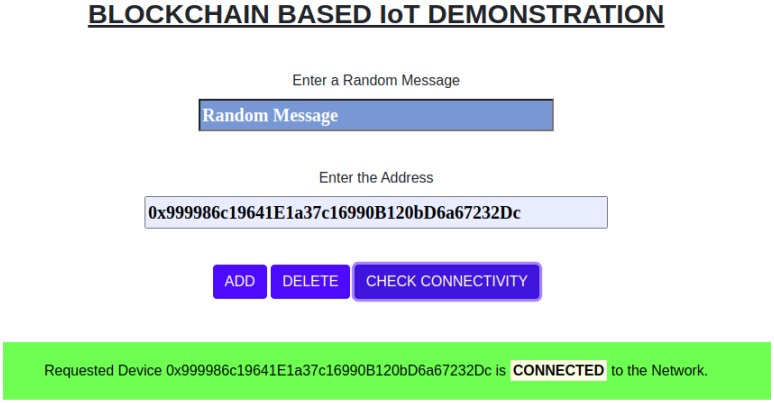

Figure 4.7: Checking Connectivity

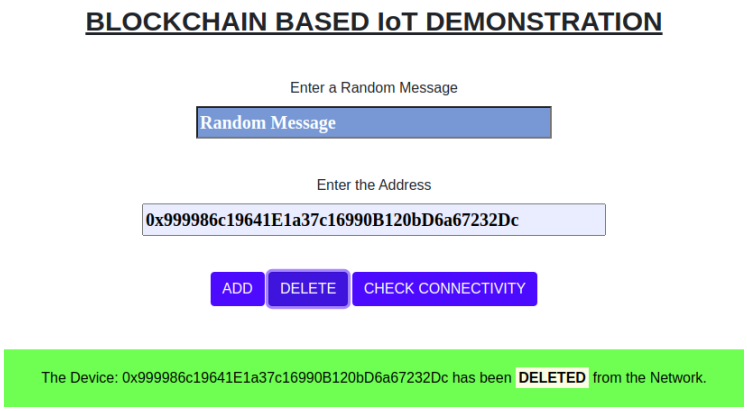

Figure 4.8: Deleting a device

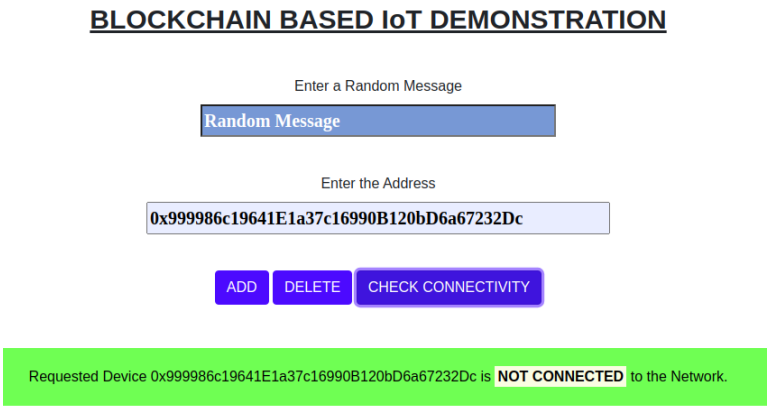

Figure 4.9: Checking Connectivity

4.3 Experimental Testing of the Model

Our network is a local private Ethereum network. By definition, it is designed such as it cannot be accessed from outside the network. For testing purposes, we changed a digit of the smart contract address on the web application. Then we tried requesting check connectivity and got the output of possible intrusion as shown in figure 4.10. This message leaves us to the decision that if any digit in the setup of the model is wrong, can be detected and the instruction after that can be provided as needed. Then as an intruder, we tried to manipulate blockchain data shown in figure 4.11. It is shown that the data is manipulated from the intruder's view but after connecting properly again, that is when we connected to the proper smart contract address again, we could see in figure 4.12 that the data was not changed and though we tried to delete a device as an intruder, the device was not deleted from the network.

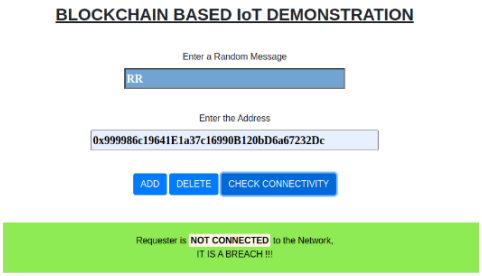

Figure 4.10: Intrusion Detection

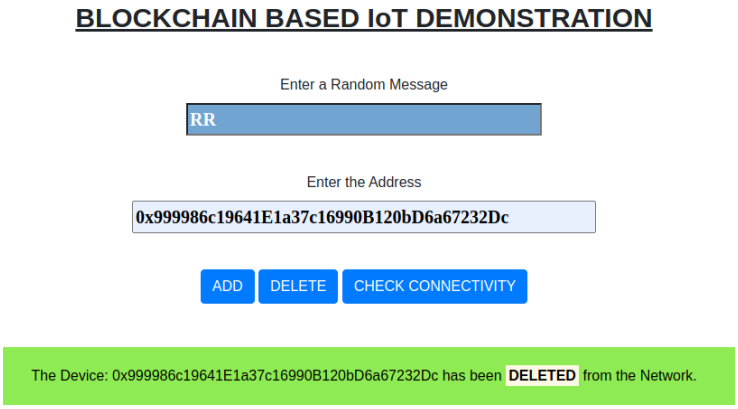

Figure 4.11: Intruders view to a request

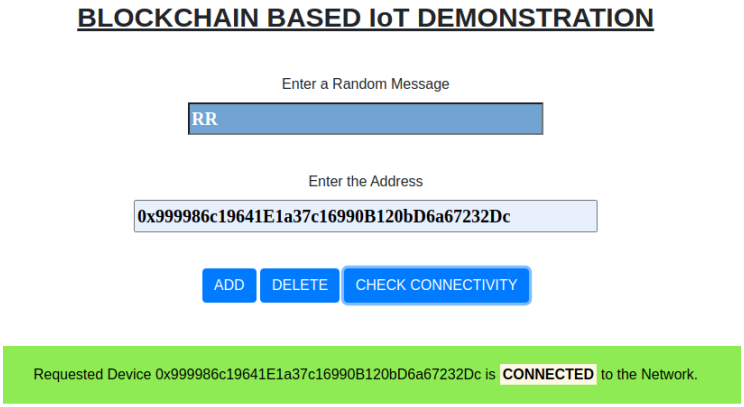

Figure 4.12: Home Miner's view of a request

Chapter 5

Result Analysis

5.1 Speed

The bitcoin blockchain can generate transactions in a 10-minute interval [4]. Ethereum on the other hand can generate up to 20 transactions per second. In our proposed model, 20 inputs per second is more than enough for what we need. We only need to think about the speed if we need to enroll more than 20 devices per second. As we have to taken the Ethereum address inputs by human interaction, it is practically very rare to get up to 20 inputs in a second. So, our system will not face this challenge.

Moreover, there is a term called 'mempool'. Mempool is the holding area of a node's transaction. For example, if a node is mining a block and a valid transaction request comes, the new request shall be placed in mempool. When the node is done with mining the block, then the transaction will be processed. In another word, the longer it takes to mine a block, the higher of chances that transactions will fall in mempool. And the speed of mining a block depends on the difficulty level of a blockchain. In our proposed system and the implementation, the transaction process time was not very long. As ours is a private blockchain, the difficulty level can be set to low so that it can function properly in the high-end devices as well as in the low computational powered devices.

5.2 Storage

Storage is one of the major bottlenecks that blockchain face. Current size of the Ethereum blockchain is around 1.498 GB [15] and it is getting larger day by day as more blocks get added to the chain. On our proposed IoT-Blockchain model, we only save a list of Ethereum addresses as a string. From the nature of our model, it should not have a huge number of blocks and that concludes us to a decision that storage should not be an issue for our model.

5.3 Cost

Our proposed model uses a private local blockchain, which is free of cost. If we somehow manage to run blockchain on IoT devices in near future, the implementation cost of this model will be Zero. Our model only needs a local private blockchain running 24/7 and our deployed smart contract on the blockchain to run properly. Ethers of a local blockchain are virtual ethers and it will not cost a single cent to the device owner.

However, if we want to implement this system where a large number of devices shall be attached or which covers large areas, like a smart city or smart power grid, we might have to make changes to the model. Switching on a public blockchain shall be a good idea then. Public blockchain can be a little costly but still, it will be worth for ensuring security. Or if a smart hospital or library wants to implement this model, they can attach a cloud server to the blockchain and keep the non-sensitive data or books on the cloud so that the blockchain does not hold unnecessary data and stays as light as possible.

Chapter 6

Conclusion and Future Works

The goal of this paper was to design a safer and more secured IoT model. Although some drawbacks like design and scalability issues, this model can improve the security of the Internet of Things (IoT) devices dramatically. This model can be a better alternative to the conventional IoT systems of now. As the chain is fully local, the chance to get access to the connected devices list is theoretically impossible. But if somehow it is accessed by someone unknown, it can be detected and it will not let them manipulate the saved data on the blockchain. This describes how this merged IoT and blockchain technology can secure user data and privacy. We described how a data breach can be a potential threat to individuals. We believe that proper implementation of this model can take time but it will provide security and privacy to individuals and keep everyone safe from potential threats.

We must already know by now that the limitation of IoT devices is small memory or processing power. Our system might not need very much power but it should be very difficult to implement it on some devices like a smartwatch or smart bulbs which rarely have a memory greater than some megabytes. Lastly, in the era of automation, we are proposing something manual alike. In our model, we ensured privacy and security by removing automation.

To properly implement our system, an IoT device has to be running blockchain 24/7. But practically, this kind of IoT based system is still not used. In future, if IoT devices have the needed computational energy for running light blockchain then the system can be fully implemented. Right now, all blockchain technologies face some difficulties as it is an emerging technology. Although the resources are mostly open, but the practice of blockchain technology is very rare these days. But as the number of blockchain-based researches grows higher, the technology is getting better day by day.

References

[1] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of things: Security vulnerabilities and challenges," en, in *2015 IEEE Symposium on Computers and Communication (ISCC)*, 2015, 180–187.

[2] D. Fakhri and K. Mutijarsa, "Secure iot communication using blockchain technology," in *2018 International Symposium on Electronics and Smart Devices (ISESD)*, IEEE, 2018, pp. 1–6.

[3] E. Kfoury and D. Khoury, "Securing natted iot devices using ethereum blockchain and distributed turn servers," in *2018 10th International Conference on Advanced Infocomm Technology (ICAIT)*, IEEE, 2018, pp. 115–121.

[4] D. Labrien, "5 pressing issues that slow down blockchain development and adoption," 2018. [Online]. Available: https://channels.theinnovationenterprise.com/articles/5-pressing-issues-that-slow-down-blockchain-development-and-adoption.

[5] G. Papadodimas, G. Palaiokrasas, A. Litke, and T. Varvarigou, "Implementation of smart contracts for blockchain based iot applications," in *2018 9th International Conference on the Network of the Future (NOF)*, IEEE, 2018, pp. 60–67.

[6] A. Sikder, G. Petracca, H. Aksu, and T. Jaeger, "A survey on sensor-based threats to internet-of-things (iot) devices and applications," en, *Researchgate.net*, 2018, Online Available. [Online]. Available: https://www.researchgate.net/publication/322975901_A_Survey_on_Sensor-based_Threats_to_Internet-of-Things_IoT_Devices_and_Applications.

[7] M. Singh, A. Singh, and S. Kim, "Blockchain: A game changer for securing iot data," en, in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 2018, 51–55.

[8] K. AlJemy, M. AlAnazi, M. AlSofiry, and A. Baig, "Improving iot security using blockchain," en, in *2019 IEEE 10th GCC Conference & Exhibition (GCC)*, 2019, 1–6.

[9] N. Fotiou, I. Pittaras, V. A. Siris, S. Voulgaris, and G. C. Polyzos, "Secure iot access at scale using blockchains and smart contracts," in *2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, IEEE, 2019, pp. 1–6.

[10] U. Nadiya, M. Rizqyawan, and O. Mahnedra, "Blockchain-based secure data storage for door lock system," en, in *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 2019, 140–144.

[11] M. Nehe and S. Jain, "A survey on data security using blockchain: Merits, demerits and applications," fr, in *2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC)*, 2019, 1–5.

[12] O. Sullivan, *The worst and weirdest IoT hacks of all times*, en. Financemonthly.com, Sep. 5, 2019.

[13] T. Tantidham and Y. N. Aung, "Emergency service for smart home system using ethereum blockchain: System and architecture," in *2019 IEEE international conference on pervasive computing and communications workshops (PerCom Workshops)*, IEEE, 2019, pp. 888–893.

[14] S. Arif, M. Khan, S. Rehman, M. Kabir, and M. Imran, "Investigating smart home security: Is blockchain the answer?" en, *IEEE Access*, vol. 8, 117802–117816, 2020.

[15] "Ethereum chain full sync data size," 2020. [Online]. Available: https://ycharts.com/indicators/ethereum_chain_full_sync_data_size.

[16] J. Frankenfield, "51% attack," en, *Investopedia.com*, Aug. 28, 2020, Online Available. [Online]. Available: https://www.investopedia.com/terms/1/51-attack.asp.

[17] J. Frankenfield, *Smart contracts*, 2020. [Online]. Available: https://www.investopedia.com/terms/s/smart-contracts.asp.

[18] "Iot 2019 in review: The 10 most relevant iot developments of the year," en, *Iot-analytics.com*, Jan. 7, 2020, Online Available. [Online]. Available: https://iot-analytics.com/iot-2019-in-review.

[19] M. Zuidhoorn, *The magic of digital signatures on ethereum*, 2020. [Online]. Available: https://medium.com/mycrypto/the-magic-of-digital-signatures-on-ethereum-98fe184dc9c7.

[20] *Contract abi specification — solidity 0.5.3 documentation*. [Online]. Available: https://docs.soliditylang.org/en/v0.5.3/abi-spec.html.

[21] *Popular internet of things forecast of 50 billion devices by 2020 is outdated*, en, Ieee.org, Available. [Online]. Available: https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated.

[22] "Solidity tutorial - tutorialspoint," en, *Tutorialspoint.com*, Available. [Online]. Available: https://www.tutorialspoint.com/solidity/index.html.

Appendix

Smart Contract Code:

```
pragma solidity ^0.5.16;
contract checker {
 address [] connected_devices;
 bytes32 miner = 0
   x8ea5947f4a057cd88cb5d0523bdb47efd3de8cd4ae45704b0c7d277e95dd17b7
   ;    //this will vary upon network to network
 function check(bytes32 hash, bytes memory signature) public view
   returns (bool) {
     if(keccak256(abi.encodePacked(msg.sender)) == miner &&
   keccak256(abi.encodePacked(recover(hash,signature))) == miner){
      return true;
     }
     else{return false;}
 }

function addit(bytes32 hashadd, bytes memory signatureadd, address
   targetadd) public payable returns (int) {
     if(check(hashadd,signatureadd) == true){
         for (uint i=0; i<connected_devices.length; i++) {
         if(keccak256(abi.encodePacked(targetadd)) == keccak256(abi
   .encodePacked(connected_devices[i]))) {
             return 1;
         }
     }
         connected_devices.push(targetadd);
         connected_devices.length++;
         return 0;
     }
     else{return 2;}
 }

function dltit(bytes32 hashdlt, bytes memory signaturedlt, address
   targetdlt) public payable returns (int) {
     if(check(hashdlt,signaturedlt) == true){
         for (uint i=0; i<connected_devices.length; i++) {
         if(keccak256(abi.encodePacked(targetdlt)) == keccak256(abi
   .encodePacked(connected_devices[i]))) {
             connected_devices[i]=connected_devices[
   connected_devices.length-1];
             delete connected_devices[connected_devices.length-1];
             connected_devices.length--;
             return 0;
         }
     }
```

```
    }
    else{return 1;}
}

 function recover(bytes32 hash, bytes memory signature)
   public
   pure
   returns (address)
 {
   bytes32 r;
   bytes32 s;
   uint8 v;
   bytes memory prefix = "\x19Ethereum Signed Message:\n32";
   bytes32 prefixedHash = keccak256(abi.encodePacked(prefix, hash)
  );
   // Check the signature length
   if (signature.length != 65) {
     return (address(0));
   }

   // Divide the signature in r, s and v var*iables
   // ecrecover takes the signature parameters, and the only way
  to get them
   // currently is to use assembly.
   // solium-disable-next-line security/no-inline-assembly
   assembly {
     r := mload(add(signature, 0x20))
     s := mload(add(signature, 0x40))
     v := byte(0, mload(add(signature, 0x60)))
   }

   // Version of signature should be 27 or 28, but 0 and 1 are
  also possible versions
   if (v < 27) {
     v += 27;
   }

   // If the version is correct return the signer address
   if (v != 27 && v != 28) {
     return (address(0));
   } else {
     // solium-disable-next-line arg-overflow
     return ecrecover(prefixedHash, v, r, s);    //another functio
  ( heart of address recovery *)
   }
 }

function connectivity_check(bytes32 hashcnc, bytes memory
   signaturecnc, address targetcnc) public view returns (bool ,
   bool) {
     bool requester_coneected= false;
     bool target_connected=  false;
     for (uint i=0; i<connected_devices.length; i++) {
         if(keccak256(abi.encodePacked(msg.sender)) == keccak256(
   abi.encodePacked(connected_devices[i])) || keccak256(abi.
   encodePacked(msg.sender)) == miner) {
             if(keccak256(abi.encodePacked(msg.sender)) ==
   keccak256(abi.encodePacked(recover(hashcnc,signaturecnc)))){
```

```
                requester_coneected=true;
            }
        }
    for (uint j=0; j<connected_devices.length; j++) {
    if(keccak256(abi.encodePacked(targetcnc)) == keccak256(abi.
   encodePacked(connected_devices[j]))) {
    target_connected=true;
        }
        }
    }
    return (requester_coneected, target_connected);
 }
}
```