12-2023

Private Ethereum Blockchain Implementation and Its Security Features for Smart Home IoT

Hasibul Grande Alam
The University of Texas Rio Grande Valley

Recommended Citation

Alam, Hasibul Grande, "Private Ethereum Blockchain Implementation and Its Security Features for Smart Home IoT" (2023). *Theses and Dissertations*. 1426.
https://scholarworks.utrgv.edu/etd/1426

PRIVATE ETHEREUM BLOCKCHAIN IMPLEMENTATION AND ITS SECURITY FEATURES FOR SMART HOME IoT

A Thesis

by

HASIBUL ALAM

Submitted in Partial Fulfilment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: Computer Science

The University of Texas Rio Grande Valley

December 2023

PRIVATE ETHEREUM BLOCKCHAIN IMPLEMENTATION
AND ITS SECURITY FEATURES
FOR SMART HOME IoT

A Thesis
by
HASIBUL ALAM

COMMITTEE MEMBERS

Dr. Emmett Tomai
Chair of Committee

Dr. Zhixiang Chen
Committee Member

Dr. Bin Fu
Committee Member

Dr. Qi Lu
Committee Member

December 2023

ABSTRACT

Alam, Hasibul, Private Ethereum Blockchain Implementation and its security features for Smart Home IoT. Master of Science (MS), December, 2023, 44 pp., 6 tables, 16 figures, references, 25 titles.

The security and privacy of IoT devices have become primary concerns as smart home networks are connected to the internet. Ethereum blockchain can be a solution to mitigate or prevent attacks – sniffing attacks, malware attacks, Eavesdropping, and Distributed Denial of Services (DDoS) attacks. Deploying Ethereum in resource constraint IoT devices is challenging due to resultant energy consumption, computational overhead, and delay. We adopted smart home as a case study to examine our methodology as a model for general IoT applications. This thesis work presents the implementation of private Ethereum blockchain that is optimized and installable on smart home IoT. We have used Hardhat framework to implement Ethereum blockchain where used Ether as a library, deployed smart contract written in Solidity, and reduced difficulty level of Proof-of-Work so that resource constant smart home devices can process mining without failure.

DEDICATION

The completion of my MS studies would not have been possible without the love and support of my family. Dr. Emmett Tomai, and Dr. Sheikh Ariful Islam wholeheartedly inspired, motivated and supported me by all means to accomplish this degree. Thank you for your love and patience.

ACKNOWLEDGMENTS

I will always be grateful to Dr. Emmett Tomai, chair of my dissertation committee, for all his mentoring and advice. Thanks to Dr. Sheikh Ariful Islam for guiding me in every step of the completion of my thesis work. My thanks go to my dissertation committee members: Dr. Zhixiang Chen, Dr. Bin Fu, and Dr. Qi Lu. Their advice, input, and comments on my dissertation helped to ensure the quality of my intellectual work.

I would also like to thank the UTRGV library team who helped me locate supporting documents for my research. Also, I would like to acknowledge the many volunteers who helped in my research endeavor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

The Internet of Things (IoT) has rapidly expanded over the past several years, revolutionizing our lifestyle and interaction with different devices. Several application areas, including transportation, smart homes, energy, agriculture, healthcare, manufacturing, and many more, are being significantly impacted by IoT. The IoTs refer to the concept of connecting devices – ”things”– that are integrated with software, sensors, and other technologies capable of connecting to the internet and other smart devices. The devices can be of any shape or size – from personal assistant devices (e.g., Google Glass) to smart microwaves that automatically cook your food, to smart refrigerators, to smart entry locks, to smart TVs, to self-driving cars, which have sensors that detect objects on their path and follow traffic signals. Over the past few years, IoT has emerged as one of the most epochal technologies of this century. According to Statista, there will be 29.42 billion connected smart devices by 2030 [1]. In addition, the International Data Corporation has also predicted that IoT devices will generate 80 Zeta Byte data by 2025 [2]. By the end of 2022, it is anticipated that the smart home market would be worth $222.90 billion [1].

An SH (smart home) provides various facilities such as – convenience, enhanced quality of life, comfort, security, etc. to our daily life. A smart home network interconnects heterogeneous devices on top of an IoT platform. The Internet of Things is vulnerable to security threats due to the lack of security measures and hardware limitations. Access to smart devices is effortless since they are confined, separated, and without an expert to manage them [3]. The smart things interact among themselves through a gateway using distinct wireless network protocols, which gives the adversaries an opportunity to eavesdrop. Due to the less processing capability of smart devices, applying advanced security measures is inconvenient for these devices. Correspondingly, IoT devices in smart homes may compromise with evildoers who observe the activity of inhabitants and steal personal information. An SH environment is susceptible to several security threats, including data privacy, authorization, authentication, issues with access control, and system configuration [4]. Conventional IoT systems or smart homes are centralized and connected to the cloud. If the central server compromises, the whole network can be exploited. Denial of Service attacks, Malware attacks, Hard-coded password attacks, Eavesdropping attacks, etc., are the most common attacks conducted in a smart home which causes inconvenience in smart home users' lifestyles and privacy. For example, adult children do not live with elderly parents. They can use the smart camera to monitor their parents and send messages to do physical exercise, take medicine, etc. Attackers can use the smart camera to verify if tenants are present in the house or collect visual data about the tenants.

To overcome security concerns, blockchain can be a potential solution which is a decentralized network system. The installation of blockchain in smart homes will protect against security threats – confidentiality, integrity, authorization, authentication, and single point of attack. Blockchain technology maintains a decentralized digital ledger with cryptography. Opposite to the conventional centralized network, blockchain operates with a distributed database system. However, the inclusion of blockchain technology results in a time-consuming, complex, and expensive system that propel technologists to explore optimization and feasibility of the technology in smart homes [5].

1.1 Previous Works

There are a number of papers proposing solutions against security attacks on IoT. While some presented approaches that not only prevent security attacks but also added a trustless system – which vanishes the centralized concept. It also takes special care of generated data by encrypting, hashing, and storing it in a distributed storage system.

1.1.1 A Lightweight Blockchain Concept

It's a new concept of Blockchain (BC) which is optimized for resource-constraint IoT devices [6]. This concept maintains the fundamental privacy and security advantages while eliminating the computational overhead, delay, and power consumption of the BC concept. This framework is a hierarchy-based structure that avoids the implementation of the PoW consensus algorithm for efficient resource utilization and better network scalability. As this framework lacks the inclusion of any consensus algorithms, IoT devices are controlled centrally in the local home network while using a public blockchain implementation for connecting multiple homes. This results, in each home having a proper blockchain implementation in addition to the lightweight blockchain framework, which negates the concept of avoiding BC installation. Though the lightweight blockchain framework is very efficient for IoT devices, it uses the traditional centralized concept – third-party involvement. As a third party gets involved there is a big question mark of trustworthiness. As a whole, this framework performs well if guaranteed trusted third party but as it implements Public BC for connecting multiple homes, this concept is not feasible for homeowners.

1.1.2 A Local Distributed Ledger System

Authors [7] presented a local distributed ledger system where each IoT device maintains a local distributed ledger while a home miner is responsible for mining blocks. The ledger in each device is maintained using a smart contract without implementing proper BC. What’s more, the authors also implement public BC for connecting multiple homes. Which forces the homeowner to have a home miner with the capacity to handle mining in Public BC. In addition to that transactions are compiled in a block every ten days, which nullifies the scope of using real-time data. Though this system tracks the transactions but fails to address security concerns, since not implementing blockchain in smart home IoT devices.

1.1.3 Ethereum as BC Platform

In this approach, authors implement proper Ethereum blockchain to manage and configure IoT devices [8]. They used RSA public key cryptosystem to manage keys – public keys are preserved in Ethereum, and private keys are kept in individual devices to avoid malicious attackers’ intervention in the system. This framework facilitates users writing their own Turing code, and communication of devices is controlled using smart contracts. Authors have focused only on the synchronization of devices and the security of the system while not mentioning the mechanism of installing Ethereum in resource constraint IoT devices as devices are unable to handle complex computations.

CHAPTER II

THEORETICAL BACKGROUND

The application of the Internet of Things (IoT) is visible in all aspects of humans' day-to-day affairs. With the development of smart devices, communication technology, cloud computing, mobile applications, etc., smart home systems have garnered a lot of attention. An SH is an IoT application that enables end users to remotely monitor and manage household appliances in real-time. A smart home, according to our definition, has a collection of gadgets with limited computational power that can communicate, and work together to provide occupants with quality of life, comfort, security, and convenience. Smart technologies have permeated every corner of human's everyday lives in the modern age, including lights, smartphones, thermostats, washing machines, refrigerators, smart TVs, and smart sensors. These intelligent gadgets interact and communicate with each other to create an intelligent environment as shown in Figure 2.1. When given access to the Internet, such an automation system transforms into an IoT-based smart home system [9].

2.1 Smart Home Architecture

The design and arrangement of the various parts and elements that make up a smart home system is referred to as smart home architecture. The smart home ecosystem is a tri-layered architecture - device layer, controller layer, and cloud service/storage layer, which is demonstrated in Figure 2.1.

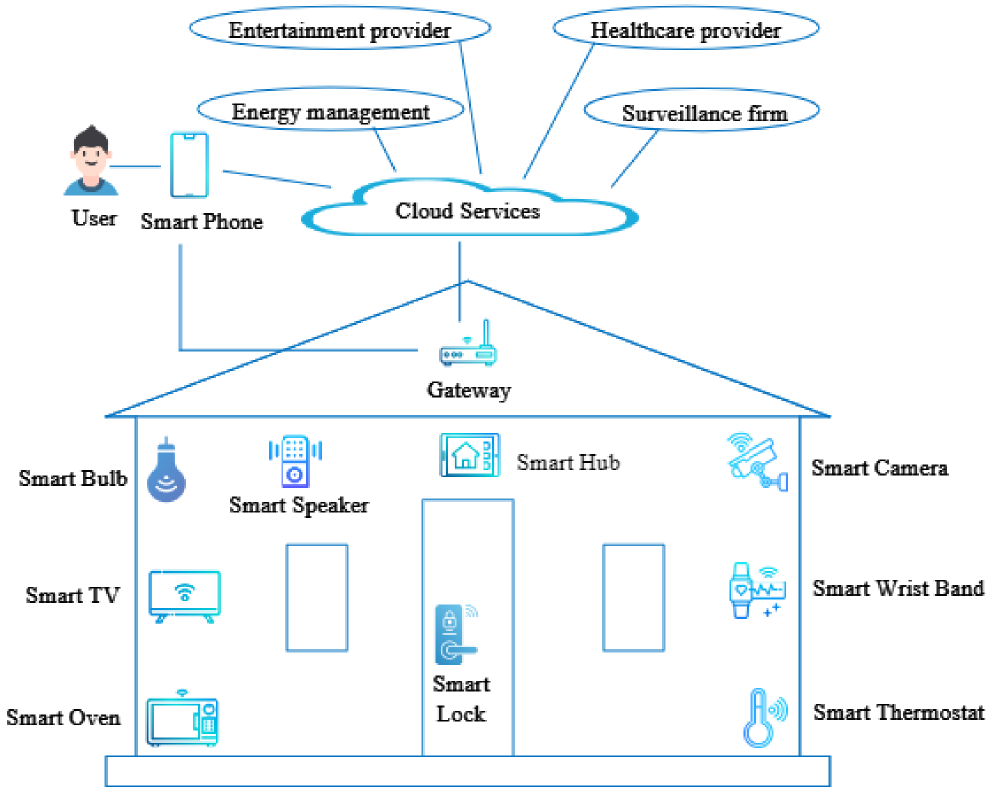


Figure 2.1: Generic smart home architecture

2.1.1 Device Layer

This layer is in charge of keeping an eye on the surroundings and gathering information. Since it works with the actual world, it must be developed using real-world objects that are kept in a home. The smart home device layer is composed of hardware consisting of sensors, actuators, and smart things.

Sensors can extract the characteristics of their surroundings and turn them into a digital output which is then analyzed by the system to identify the current status of the environment. There are various sensors used in smart home environments such as temperature sensors, motion sensors, contact sensors, etc.

Actuators have manipulation and control capability over the physical environment. It receives digital signals from the system that is then translated into actions - turning on/off lights, triggering alarms, activating speakers, etc.

Smart things are objects with sensors and actuators which are connected to the smart home network. Examples of smart things consist of objects like a smart bulb that switches on based on motion detection, a smart lock, a smart camera, etc.

2.1.2 Controller Layer

The controller layer functions as a central decision-making structure that collects and correlates data from smart home devices. It makes decisions and initiates activities by sending messages and commands to the relevant devices based on the information received and/or the situation. The controller can be a device (e.g., Alexa, Google Home, Xiaomi smart speaker, etc.) or a cloud application. There are several smart home devices available in the market such as Apple’s HomeKit, Google Nest Weave, Samsung's SmartThings, Alibaba Smart Living, etc. that are controlled by a compatible IVA (Intelligent Virtual Assistant) or application installed in a smartphone or computer. Though the majority of smart devices have the processing capacity and act like independent or autonomous devices (e.g., smart thermostats, smart TV, etc.), they require an interface between them and the cloud or user. In that case, an application or IVA functions as an interface that is a part of the controller layer [10].

2.1.3 Cloud Service/Storage Layer

This layer stores sensor or devices' generated data that is used by service providers to facilitate smart home users with specialized services. Services are applications hosted in the cloud that is used by the user to control or device management. Hosted applications are responsible for collecting data, processing data, analyzing data, decision-making, etc.

2.2 Smart Home Applications

There are several smart home applications or services. Presenting four major categories - Entertainment & comfort, Healthcare, Surveillance, and Energy management with use case scenarios. Table 2.1 shows key services in each category.

Table 2.1: Categories of Smart Home Applications

Entertainment & Comfort	**Healthcare**	**Surveillance**	**Energy management**
Simple to utilize and regulate	Befitting tenant housing, especially elderly	Identifying strangers	Ensuring efficient energy usage
Offer comfort	Continuous patient monitoring	Detecting movement of objects	Logical usage of gadgets
Reduce physical interaction with devices	Precautionary treatment warning	Preventing unfortunate incidents	Reliability and quality of devices
_	Easy interaction with medical institutes	_	_

2.2.1 Entertainment & Comfort

Smart home optimizes the lifestyle of its users through devices which are programmable or can be managed remotely using software programs. Thus, it improves the comfort, convenience, and interactivity of smart home users [11]. For example, smart lights automatically turn off when inhabitants of the home leave. And using motion sensor, smart lighting system turn on lights when it detects movements of inhabitants. A smart refrigerator is used to keep track of available food in it, if any food is fully consumed it triggers a shopping list to the homeowner.

Residents can control the temperature of the refrigerator using remote devices. A smart washing machine keeps checking the level of ingredients, if any of them falls below the threshold, it notifies to refill that item. Inhabitants can program a smart vacuum remotely to clean up dust. Smart watering is also another example of a programmed or scheduled system that opens all the conduits when a certain time is met for watering plants. Intelligent personal assistants are voice-controlled, hands-free gadgets that can perform a wide range of tasks including voice communication, internet surfing, playing videos/audio, and also managing other devices - smart thermostats, smart bulbs, etc. The most popular and commonly used intelligent personal assistants are Apple Siri, Google Assistant, Microsoft Cortana, and Amazon Alexa [10]. Nest Audio is a smart speaker that works as a command interface for Google Assistant. It allows users to question anything, it can be about navigating location, weather, breaking news, events, etc. Google Assistant does the searching and brings the solution for you. It can set alarms, place orders, manage a to-do list, play a playlist, etc. like other smart speakers such as Amazon's Echo, and Apple's Home Pod.

2.2.2 Healthcare

Healthcare services for inhabitants of smart homes have seen a revolutionary improvement, especially for elderly persons with disabilities. Different sensors or smart gadgets can be installed in a smart home to look for physical or mental abnormalities in the occupants. This prospect offers various advantages, including reduced expenses in contrast to institutional living or giving elderly people an opportunity to stay with their family instead of living alone in a healthcare institution. For example, smart speakers and screens are used to inform the patient or elderly person which medicine to take or other tasks, devices also notify the hospital or other family members if the health condition of the patient deteriorates, or immediate clinical health care needed. Generally, adult children in USA, Europe, or Japan do not live with elderly parents. They can use smart cameras to monitor their parents and send messages to do physical exercise, take medicine, etc. They can also help their parents turn lights on/off, lock the door, control the air-conditioner, etc. remotely using mobile devices [12]. Smart robots can be used to help elderly people with their daily activities and not let them feel alone. The Italian government made the decision to build a village with smart homes to give older folks the chance to enjoy healthy, prosperous, and adequate living amenities.

2.2.3 Surveillance

Several features of smart devices can be used for surveillance and security of smart homes. For instance, intrusion detection can be accomplished by using motion sensors and smart cameras. If the movement of an unknown person is detected, an intrusion detection application will trigger a message to the homeowner, even the system can send an audio/video message including the person's image. Smart surveillance cameras take advantage of image classification techniques to identify unknown people. Another situation, known as vacation mode, enables smart cameras, sensors, and alarms to work together to monitor any dubious movements, put the locks in closed mode, and control the lights on and off to mimic activity that would take place while the inhabitants of the smart home are present. Smart door locks with cameras can be used to give entry permission to privileged people [13]. Gas and smoke detection can be utilized to sense the ambiance of smart homes for security reasons as well as residents' health concerns. Ionization, Optical recognition, and air sampling methods are utilized in this application. The system notifies the users through email or message about health risks and alarms nearby fire services if smoke or fire is detected.

2.2.4 Energy Management

Energy saving is vital in smart homes. Smart things are utilized in smart homes to deliver cutting-edge technology and save energy consumption. These gadgets save energy while increasing effectiveness. For instance, a smart lighting system manages light bulbs by turning them on/off based on the presence of the residents in a room. In the daytime, the smart lighting system can properly utilize self-generated energy while at nighttime, it can save energy conservation by shutting off all the standby devices. The remote controlling feature of the smart home helps to reduce energy consumption such as users can remotely manage a smart air conditioner to control the indoor temperature based on the outdoor temperature and thermostat reading. Thus, smart home systems ensure efficient energy consumption also known as need-based usage. A smart grid provides efficient power supply and consumption using information technology and grid energy systems. Smart home systems take great advantage of this scenario and play a vital role in the communication between energy suppliers and consumers [14]. A smart meter can be integrated into the home to track energy consumption patterns. The system automatically sends energy consumption reports to users and vendors. Based on the report, the vendor can provide recommendations to reduce energy usage [15].

2.3 Blockchain Concept

In 2008, Satoshi Nakamoto proposed blockchain. Generally, BC is a distributed digital ledger of transactions, cryptographically signed and verified by miners. Blocks are connected to the previous one cryptographically which makes the blockchain tamper-proof. Blockchain comprises three elements -- block, node, and miner.

Every blockchain contains multiple blocks. BC users commit candidate transactions to the BC network through software -- web services, digital wallets, smartphone apps, desktop apps, etc. This software transmits the transactions to non-publishing or publishing nodes in the BC. In most blockchain implementations, the distributed transactions are added to the BC from a queue of waiting transactions. Blocks are made of two parts - block header and block data shown in Figure 2.2. The block header is composed of metadata whereas block data contains the list of transactions and ledger events. To explain it in more detail, it contains - the block number, the hash digest of the previous block header, the nonce value of a 32-bit whole number randomly generated during a block creation, the size of the block, A hash value of the block data that can be accomplished by different methods such as creating a hash of all the combined block data or by a Merkel tree. Blocks are linked together with the hash of the previous one, thus creating a blockchain. The connection of nodes to the chain makes BC a distributed ledger. Nodes are electronic devices that have copies of the blockchain and keep the BC network operational.

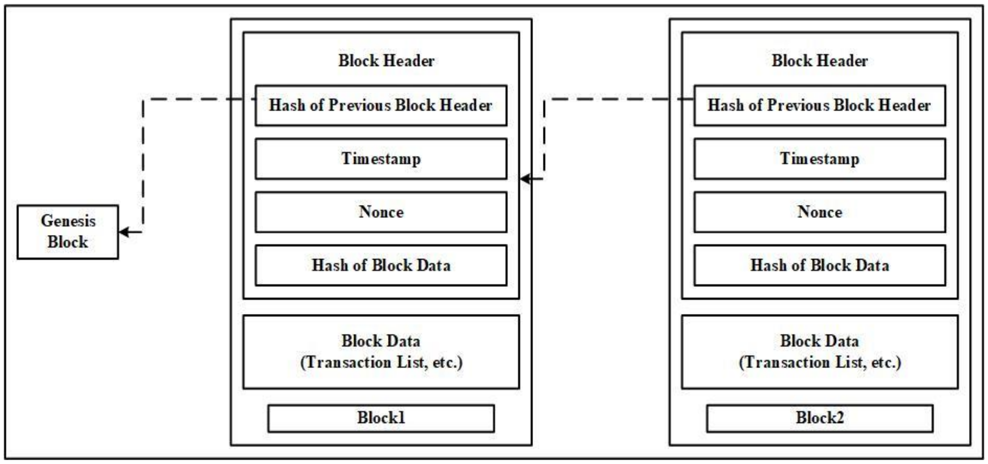

Genesis Block
Block Header
Hash of Previous Block Header
Timestamp
Nonce
Hash of Block Data
Block Data (Transaction List, etc.)
Block1
Block Header
Hash of Previous Block Header
Timestamp
Nonce
Hash of Block Data
Block Data (Transaction List, etc.)
Block2

Figure 2.2: Generic Block structure

Miners mine blocks and add them to the blockchain. They find the hash using special software that solves the complex math problem by changing nonce and timestamp [16]. If a valid hash is found, the block is added to the chain. Otherwise, miners will again attempt to solve the hash with a different nonce. This process will continue until a valid hash is found.

2.4 Categories of Blockchain

Based on control mechanism and authentication, BC can be classified into three kinds -- public, private, and consortium blockchain. These categories are described, and Table 2.2 presents a comparison among them.

Table 2.2: Comparison of various types of blockchain

Properties	Public	Private	Consortium
Nature	Open and decentralized	Restricted and controlled	Restricted and controlled
Consensus protocols	PoW, PoS, DPoS	PBFT, RAFT	PBFT
Transaction approval frequency	Long	Medium	Short
Participant type	Anonymous and Resilient	Trusted and identified	Trusted and identified
Permissions	Permissionless	Permissioned	Permissioned
Transparency	Low	High	High
Energy consumption	High	Low	Low

2.4.1 Public Blockchain

A permissionless blockchain, often known as a public blockchain, is an open decentralized network where users can publish blocks. Public BC platforms are often open-source software. It turns out that anybody can read the BC and add transactions to the blockchain since everyone has the power to publish blocks. Any user on the BC network has access to the ledger and may read, write, review, or audit it. Every user in a public BC collects the transactions, validates them, processes consensus mining, and in the end, commits the block to the public BC [17]. Utilizing this opportunity, malicious users may subvert the system by publishing blocks in an abnormal way. To avoid this issue, public BC often utilizes consensus protocol where multiparty is involved. Examples of such consensus protocols are proof of work, proof of stake, proof of authority, etc. Cryptocurrencies use public or permissionless blockchain networks.

2.4.2 Private Blockchain

A blockchain that is private or permissioned is a decentralized system where each published block must be authorized by a controlling authority. As a chosen person or a group in charge oversees the block mining process, access to an unknown or new user is restricted without an invitation from a controlling authority [18]. The controlling node has the authority to commit transactions and power assigning the permission to read and write to other nodes. This feature inclines the permissioned BC towards the centralization concept. However, other features of private BC such as -- transparent log, distributed ledger, smart contract, and consensus ensure the decentralization concept. Since all the nodes are trusted and the authority to publish blocks can be revoked (different from permissionless blockchain), consensus models in private BC are faster and computationally less expensive compared to public BC networks. The private blockchain is suitable for organizations and individual use [19]. The private blockchain is suitable for organizations and individual use.

2.4.3 Consortium Blockchain

The composition of private and public blockchains is known as a consortium blockchain where the consensus and block validation decisions are assigned to a group of permissioned individuals. In a consortium BC network, a multi-signature scheme is used in the mining process of blocks and mined blocks are considered valid if and only if they are signed and approved by the controlling node. The vulnerability against tampering attacks is the prime drawback of consortium BC [19]. In addition, controlling nodes can collaborate with adversaries to reverse or tamper transactions, thus threatening the irreversibility and immutability of the consortium BC network.

2.5 Consensus Protocols

A new version of the Byzantine Generals dilemma arises in blockchain networks where untrustworthy nodes must come to a consensus. In the Byzantine problem, if a part of the army attacks the city, they may become unsuccessful. Thus, a group of generals must reach an agreement on attacking the enemy or not via sending messages. However, there may be traitor generals among the engaged generals who might instruct certain generals to take different actions [20]. This trustless scenario is similar to the blockchain network where nodes are anonymous. Since the blockchain is a distributed system, some protocols are required to ensure the consistency of ledgers on distributed nodes. Table 2 shows a comparison among different consensus protocols.

2.5.1 Proof of Work (PoW)

PoW is a proof-based consensus system that works with the node that is authorized to add the most recent block that was mined to the chain along with the necessary proof [21]. According to the procedure, a batch of nodes or all the nodes broadcast candidate blocks of verified transactions, which rises confusion about which node will put the transactions into the block. To solve this problem, Proof of Work comes into work, where nodes have to solve a complex computational puzzle to achieve the right appending recently generated block to the BC. Nonce combined with known input is used to continuously calculate the hash value. Finding an acceptable nonce is difficult on top of predicting the valid output hash value. The endeavor of guessing an appropriate nonce by the nodes is titled the Power-of-Work. When a suitable nonce is found, miners send the block to all the nodes in the BC network to prove that the solution is valid. The miner adds the block to the chain if it gets approval from all the nodes. Figure 2.3 shows the process of block creation in the PoW protocol.

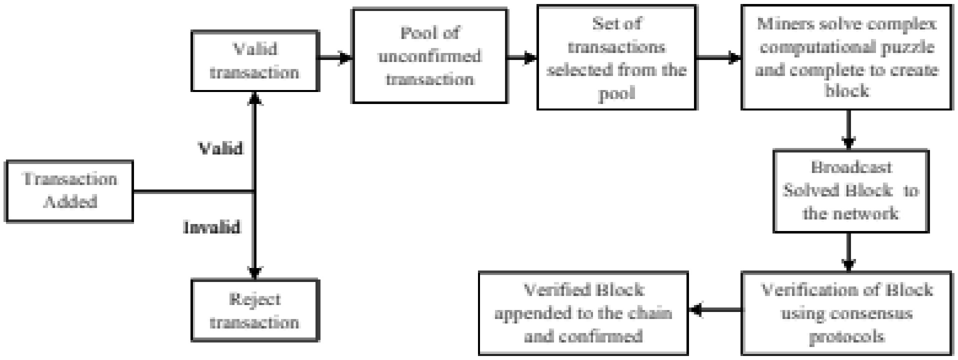

Transaction Added
Valid
Invalid
Valid transaction
Reject transaction
Pool of unconfirmed transaction
Set of transactions selected from the pool
Miners solve complex computational puzzle and complete to create block
Broadcast Solved Block to the network
Verification of Block using consensus protocols
Verified Block appended to the chain and confirmed

Figure 2.3: The process of block creation using PoW.

When multiple miners find an acceptable nonce and try to broadcast the blocks, the nodes face an ambiguity of which miner's block to receive and approve because nodes verify the early coming block and ignore the late arrival blocks. Thus, branches occur in the BC which is called the forking problem [17]. According to PoW, only the longest chain will be considered valid. Figure 2.4 shows the forking problem where two verified blocks P1 and F1 are generated from

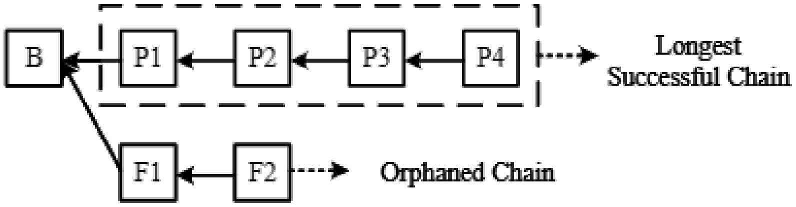


Figure 2.4: Forking problem in BC.

block B at the same time. When a P2 is appended to block P1, miners working on the branch F1-F2 leave this branch as orphaned and switch to the other branch. Generally, a fork or branch with six successive blocks is considered successful. The mining process in PoW is not sustainable since only one miner will be successful. The major drawback of this consensus protocol is the requirement of huge computational resources to solve the complex puzzle.

2.5.2 Proof of Stake (PoS)

PoS is a consensus model that consumes less energy than the PoW protocol. The miners involved in the block creation process in PoS prefer having a sufficient stake in the system rather than investing in resources that are used to solve difficult computational puzzles [22]. Achieving the right to validate blocks depends on the portion of stake in the system. A node after being chosen as a validator places a bet with the help of its stake. Successful approval of a block rewards the validator with fees. This property enables PoS to supply superior latency, throughput, and energy efficiency. PoS does have certain negatives, one of which is that a wealthy node would have more chances to validate blocks, leading to a single node controlling the blockchain network. Next, PoS is more prone to malicious activities than PoW as this consensus protocol requires a low mining cost compared to PoW. Lately discovered Nothing-at-stake problem is also another disadvantage of the PoS consensus protocol [23].

2.5.3 Delegated Proof of Stack (DPoS)

Like PoS, DPoS is also a consensus protocol where elections are held. Contrary to PoS's direct democratic strategy, DPoS pursues a representative democratic policy where the representatives called ‘witnesses’ are elected by nodes to create and verify blocks [24]. On behalf of the stakeholders, the elected representatives rotate on voting to verify the legitimacy of the previous block. The protocol enables quicker confirmation of transactions and block creation since it has a much lesser number of participants for validating blocks compared to PoS. It also provides the fine-tuning facility of block size. However, the major drawback of the protocol is its centralized nature where the giant-stake participants can influence other participants to vote them to become validators.

2.6 Smart Contract

The concept of a smart contract (SC) was first proposed in the early 1990s by Nick Szabo. The negotiation of a contract can be digitally facilitated, verified, or enforced using a smart contract, which is a computer script integrated into the blockchain. According to Szabo, SC is turning contractual conditions into programming scripts and embedding them into hardware or software that enables them to self-trigger [25]. The purpose of the smart contract is to exclude the trusted intermediaries during the operation of a transaction. The EVM (Ethereum virtual machine), a Turing-complete virtual processor, works with the Ethereum blockchain to facilitate and carry out programmable smart contract. A smart contract is initiated by calling a function via a transaction where the transaction sender takes ownership of the smart contract. Smart contracts contain states and functions. States are variables that store information like data or the location where a smart contract is deployed.

A writable state adds states to the blockchain while a constant state can never be edited or altered. Functions are scripts capable of modifying and reading states. While writing functions need ether to run because the transition of state needs to be enciphered in a new block, read-only functions don't. The mechanism of an SC is shown in Figure 2.5. When a certain condition is met, the smart IoT device triggers the actions of the Smart contract. In this example, the condition of the smart camera is to recognize the homeowner. Then sent the identity information to a smart lock, the transactions' light-weight proof of work verifies the local miner [7]. Ultimately, the smart lock gets unlocked and opens the door for the homeowner.

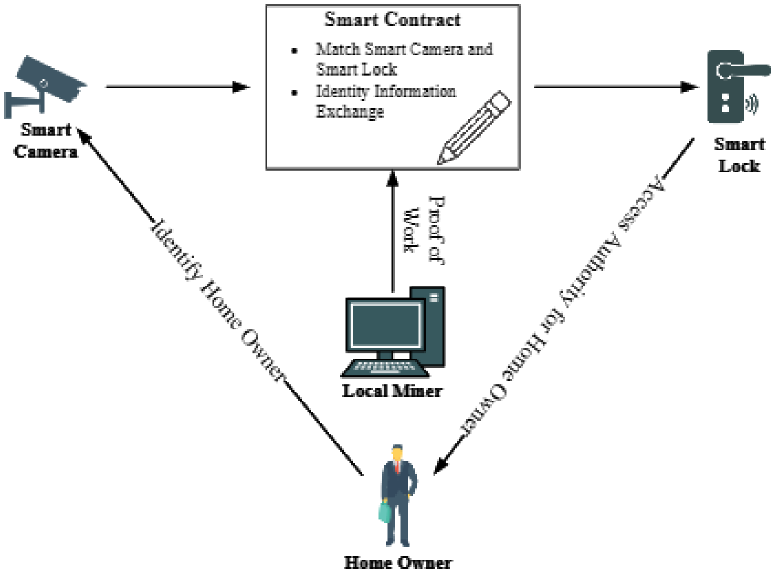

Smart Contract
Match Smart Camera and Smart Lock
Identity Information Exchange
Smart Camera
Smart Lock
Proof of Work
Identify Home Owner
Access Authority for Home Owner
Local Miner
Home Owner

Figure 2.5: Smart Contract workflow [7].

2.7 Security Features

Apart from providing distributed data storage system, the presented solution also provides data integrity, authentication and protection against security attacks. Due to distributed nature of the approach, the system nullifies the single-point failure problem. Passive attacks and active attacks are the two types of security attacks that can be conducted against smart homes. Passive assaults aim to watch without affecting data or network performance. These attacks are typically imperceptible and use transmissions, eavesdropping, and monitoring techniques to operate. In active attacks, the attackers use the data obtained by passive attacks to alter the data, messages, system operations, or system resources. Malware, sniffing attack, and denial-of-service attacks are typical active attacks. To mitigate or neutralize these attacks researcher recommended various mechanisms. Table 2.3 shows the security attacks and mitigation techniques proposed by researchers.

Table 2.3: Security attacks and mitigation techniques.

Security Attack	**Mitigation Technique**
Sniffing attack	Trustworthy network with authentication measures and encryption protocols [26]
Eavesdropping	Lightweight and portable encryption methods [27] [28]
Distributed Denial of Service attack (DDoS)	Lightweight DDOS mitigation system, Machine learning algorithms [29], Blockchain network [31] [32]
Malware attack	Whitelisting-based solution [36], Blockchain-based autonomous system [33]

2.7.1 Sniffing Attack

Attackers gather users' confidential information by placing malicious devices or sensors instead of actual devices. They infiltrate the system as sniffer programs and run them to steal private data while users are unaware of the exploitation [34]. It is crucial to make sure that devices are connected to a secure network with proper authentication mechanisms in order to prevent sniffing. In addition to that, as attackers track the network traffic to find users' credentials to conduct sniffing, the authors [35] proposed using encryption protocols such as AES, RSA, Triple DES, etc., to encrypt the data which leaves smart devices so that the original information is not understandable to adversaries. In our proposed approach, we have implemented a lightweight ECDSA encryption algorithm with 64 iterations that makes it more robust. ‘The Watson’, IBM’s supercomputer would take 0.65 billion billion years to crack a 32-byte encrypted data as it has 2^256 combinations. Even if attackers use powerful computers, they will not achieve their goal.

2.7.2 Eavesdropping

In this attack, attackers eavesdrop on the network communication to monitor or steal the data without any alteration. Due to the different technological limitations of smart home things, traditional encryption mechanisms cannot always be used [6]. Utilizing this scope, adversaries can access sensitive healthcare data and alteration of the data could lead to a life-and-death situation. To address the requirements, Thakor et al., [30] suggested the best suit algorithms such as SIMON, SPECK, PICCOLO, TWINE, PRESENT, and Midori for smart home devices. As mentioned earlier, our system uses a lightweight ECDSA algorithm along with distributed data storage feature that voids single-point device failure. For example, even if healthcare data is lost, the homeowner can retrieve it from other nodes in the blockchain network.

2.7.3 Distributed Denial of Service (DDoS)

Adversaries conduct a DDoS attack to disrupt or delay services temporarily or indefinitely to legitimate users [38]. In the context of Smart Home, a group of compromised devices scattered over the internet called a botnet is utilized to operate a DDoS attack against a target device or network shown in Figure 2.6.

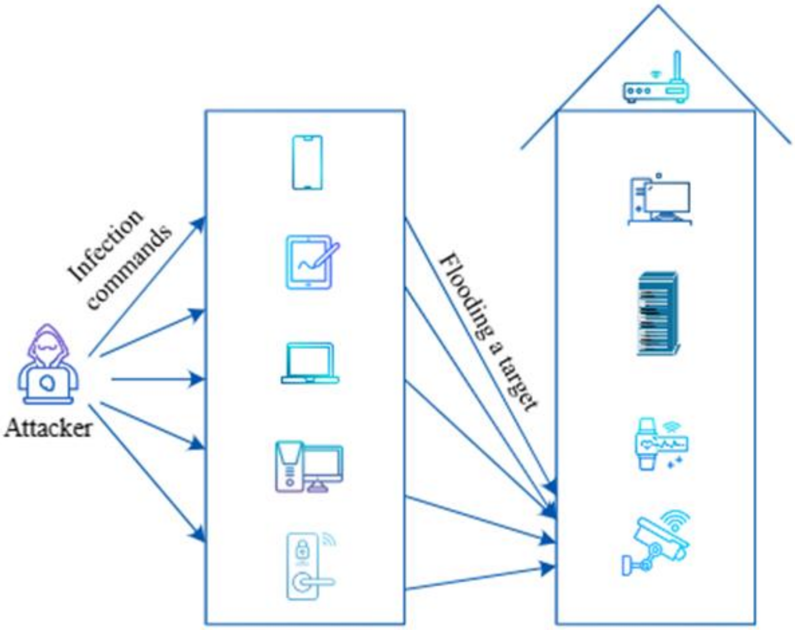


Figure 2.6: DDoS attack procedure.

To successfully operate the attack, attackers launch several techniques, such as flooding with requests or responses to exceed the bandwidth of the victim system (UDP Floods, ICMP Floods), exploiting protocol flaws (SYN Floods, Ping of Death), or flooding a victim device with service requests out of its capacity to respond. Launching a flood attack, adversaries do not require protocols flaw or identification of devices but flood the targets [38], as the Internet of Things has limited memory and processing capacity. DDoS attack in the smart home system has different consequences, as targeting a smart refrigerator causing food waste is different than the consequence of neutralizing a motion sensor that detects movement in a room while the effect of targeting a health service can have a disastrous impact on patient health. Although DDoS attacks are highly challenging to identify and stop, there are strategies to lessen their effects. In [32], the authors presented a blockchain-based DDoS detection system on the edge of the IoT network. They did not discuss the authentication and implementation process in depth and the proposed approach does not prevent DDoS attacks. The authors of [31] proposed a deep-learning-based blockchain system. In the approach, switch authenticity is managed by the blockchain, and a deep Boltzmann machine is used to detect anomalies. Again, this system also does not prevent the attack and the cost of communication and computation is expensive. Contrary to the above-mentioned solutions, our proposed system provides an authentication process for the devices before letting to join in the blockchain that prevents the scope of conducting DDoS attacks. Table 2.4 shows the comparison of the authentication time taken by our system and existing solutions.

Table 2.4: Comparison of the authentication time taken by existing approaches.

Related Work	**Number of messages**	**Authentication time (ms)**
[40]	8	2.1×10^{1}
[41]	5	2.3×10^{1}
Proposed system	2	1.7×10^{-2}

2.7.4 Malware Attack

With the passage of time, adversaries are showing more interest in malware attacks on IoT devices. This attack is of the same nature as the DDoS attack. Popular malware attacks such as Bashlite, Silex, and Mirai are conducted on smart home devices. To carry out the attack, attackers look for vulnerable devices nearby by finding open Transport Control Protocol ports or IP addresses. Once a port is found, it conducts a brute-force attack using a dictionary of IoT devices' most common user credentials. The authors of [36] proposed a whitelisting-based solution preventing malware from spreading in IoT. The solution works in two phases. On a clean device, the profiling module creates a hash for all the programs running on the IoT devices in the first phase and preserves it in the database. In the second phase, the application whitelisting is conducted by the "Application Monitor" computing hash of the application before its execution and comparing it with the stored hash in the database. In [33], presented a blockchain-based architecture to prevent Mirai attacks on IoT devices where the network is partitioned into different Autonomous Systems (AS). A list of IP addresses of IoT devices is stored and shared using blockchain. Each AS keeps track of communication activity within the network and decides if a node is compromised by comparing the total number of packets delivered with a predetermined threshold value. Though this approach is a promising one, the block propagation delay increases based on the consensus protocol and the size of AS. In our proposed system, a compromised device even does not get the chance to join the blockchain network, since each device goes through an authentication process. Even if there exists a compromised device, the system recurrently checks the available ether on each trusted device after a set time interval. If the gas limit is insufficient to make a transaction, it indicates the detection of an infected device.

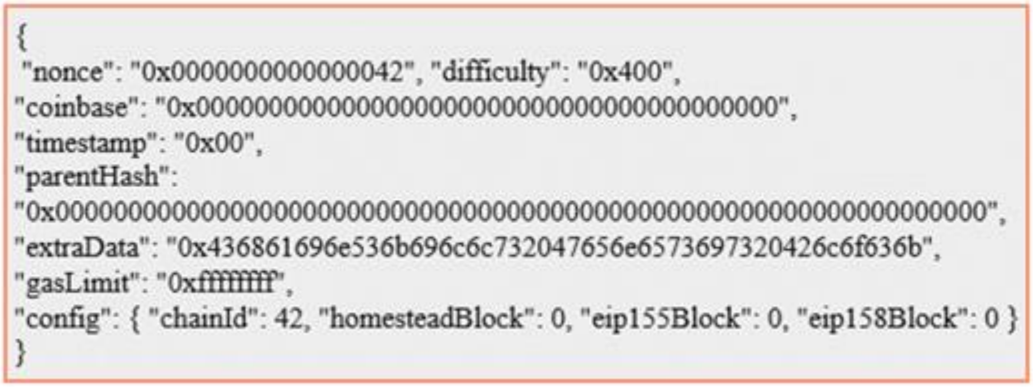
CHAPTER III

CONTRIBUTION

The integration of blockchain in the Smart home makes it a more robust and secure system. Different types of blockchain (private, public, and consortium) can be implemented in smart home architecture. However, before choosing a blockchain for smart homes, one should be careful because the erroneous selection of BC may result in low performance, energy inefficiency, and security loophole. As smart homes consist of a small number of IoT devices and low computational power, we have implemented private blockchain. Especially selected Ethereum because it is programable - we can add smart contracts based on our requirements.

3.1 Implementation of Private Ethereum Blockchain

To begin with, after installing Geth, we have initialized the genesis block of the blockchain in each node. The genesis block includes a nonce (which is generated by consensus protocol), difficulty (which defines the complexity of the computational puzzle), coin base (account number of the node), timestamp, "parentHash" (Hash of the previous block), extraData (Hash of the data of current block), "gaslimit" (cost for mining a block in term of ether), and config (includes network setting). Figure 3.1 shows a sample of a genesis block. The difficulty level is inversely proportional to the complexity of generating nonce. By modifying the difficulty level, we can increase and decrease the computational complexity. The less is difficulty level, the less computational power is required to solve an acceptable nonce. PoW algorithm takes the difficulty level as an argument and produces a unique 32-bit nonce. The higher the difficulty level, the higher the number of leading zeros in the generated nonce. Whichever node first solve the nonce, it gets the authority to add the block in the blockchain.



```
{
 "nonce": "0x0000000000000042", "difficulty": "0x400",
"coinbase": "0x0000000000000000000000000000000000000000",
"timestamp": "0x00",
"parentHash":
"0x0000000000000000000000000000000000000000000000000000000000000000",
"extraData": "0x436861696e536b696c6c732047656e6573697320426c6f636b",
"gasLimit": "0xffffffff",
"config": { "chainId": 42, "homesteadBlock": 0, "eip155Block": 0, "eip158Block": 0 }
}
```

Figure 3.1: Genesis block.

3.1.1 PoW consensus protocol

To get the authority to add a block of transactions in the blockchain, nodes need to come to a consensus who should get the privilege. Here comes the necessity of Proof-of-Work protocol, which help to decide to find the appropriate node. According to the protocol, all nodes need to find an acceptable nonce, whichever node find first, it gets the authority to add a block in the blockchain. Working procedure of PoW is given below:

1. Includes all transactions that are waiting to be added in the blockchain to create a new block.
2. A Merkle tree verifies and summarizes all the transactions.
3. If it is valid, then selected transactions are included in the block.
4. Miners generates a hash of block by changing nonce and time stamp.
5. The system then compares the generated hash with the target.

6. If the hash is above the target value, then it starts again from step 4.
7. If the hash is below the target value, then the PoW is verified as a success and added the block to the blockchain.

A Merkle tree is a binary tree where each leaf node is the hash of a transaction. Each intermediary parent is the hash of its children. Merkle Tree makes sure any transaction is not modified maintaining the tree. If a transaction is modified, the hash of the root will not match the latest rendered hash. To generate the hash of the block, we have deployed the keccak-256 hash algorithm on the ether engine. We gave the data of the block as an input while it gets the current nonce from the ether engine. A hash is generated using the function and put into Direct Acyclic Graph. Later on, it fetches the hash from the DAG and iterates this process 64 times and produces a 32-bit hash. Figure 3.2 shows the process of hash generation.

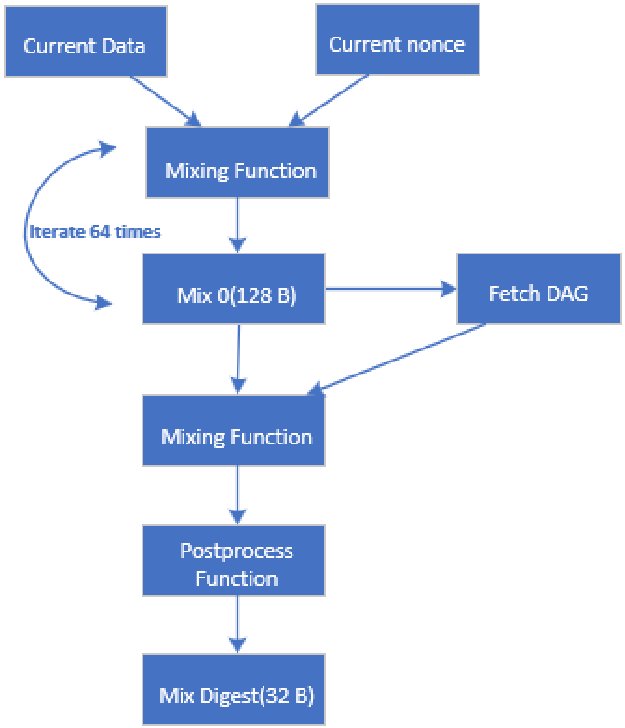


Figure 3.2: Hash creation process.

3.1.2 Encryption Algorithm

For encrypting all the transactions (data) generated in the devices, we have used Elliptic Curve Digital Signature Algorithm (ECDSA) in the ether engine. ECDSA is a public key encryption algorithm that generates keys based on algebraic expression of elliptic carves over finite fields. It produces one public keys which is known to all the nodes in the blockchain network and a private key that is preserved in each node. To encrypt the hash of data, nodes use their private key and then send to the memory pool, which is later on broadcasted to the network. Other nodes verify the hash using private key of the respective node. Figure 3.3 shows the encryption process of the generated hash.

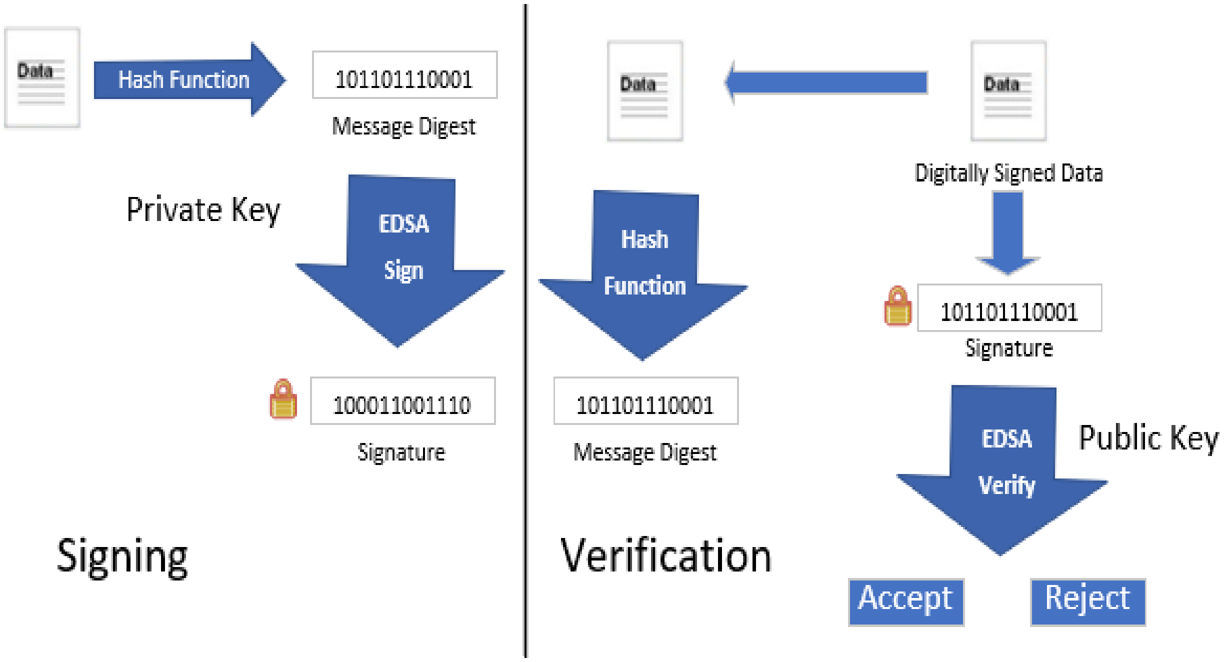


Figure 3.3: ECDSA Signing and Verification.

3.1.3 Proof-of-Authority (PoA) Consensus Protocol

Proof-of-Authority is a consensus protocol that allows private blockchain nodes to reach consensus efficiently. It is an alternative to PoS as it is more energy efficient and needs fewer computational resources. This algorithm has a concept of a preapproved validator that has the authority to approve a block of trusted transactions. To become a validator, a node must meet three requirements. First, a node must be trustworthy without any criminal record. Second, the node's identity must be validated on the network to confirm the real identity of it. Finally, A candidate node should have a good reputation and the ability to invest money. Out of all the candidate, a random validator or leader get selected using a formula.

Leader = ((time - first) / step) % nodes

Where time is the current time, first is the time stamp of the first block, and step is the total number of blocks. All the nodes in the blockchain network have permission to mine block but the block approved by the validator will publish only. As validators are selected randomly, it negates any influence of a well-reputed node. This algorism is perfect for IoT devices as it does not need to solve computational puzzles which results in low energy consumption. We have also implemented this protocol to compare the performance with PoW.

3.1.4 Verify IoT Before Connecting it to the BC

In this system, each device that wants to connect to the blockchain generates Elliptic curve private and public keys and sends the public key to the home manager. After getting the request, the home manager creates an id (hash) using keccak256 consisting of the public key, a random number, and the signature of the manager’s private key.

Algorithm 1: Verify Device

```
// Check if the id known to the Home manager

Begin

If(id in trustedList)

     Then

         Return true

      Else

          Return false

End;
```

3.1.5 Adding the Device to BC

Before adding the device to the BC, the program checks if the id is in the trusted list, the time complexity of the algorithm is O(n) where n is the number of id in the trusted list.

Algorithm 2: Adding Device

```
// Takes id as a parameter
Begin
If(id.length == 0)
  Return error()
Else
   If(id in trustedList)
      Return {netId, rpc_port}
   Else
      Return error()
End;
```

3.1.6 Transaction Accomplishment

To accomplish a transaction, we implemented a smart contract consisting of two functions – Send() and Receive(). Whenever a transaction occurs between two nodes, it gets triggered. It excludes trusted intermediaries during the operation of a transaction. The smart contract is deployed on the Ethereum Virtual Machine (EVM).

Algorithm 3: Transaction Between Nodes

```
Begin
// Send function takes msg and destination as parameter
Check if receiver is trusted
Send {
    packet[destination] = receiver;
    packet[data] = msg;
}
// Receive function parse the massage body in JSON format
Check if sender in TrustedList
Receive{
    msg = msg.JSON()
    data = msg.data
    Triger response if requested
}
End
```

3.1.7 Check Gas Limit

In our system, each node has a specific gas limit to utilize by default, and it is programable. Nodes earn ether gas as a reward when it adds a block to the blockchain successfully. If the gas limit of a node is ‘Zero’ or not sufficient to make a transaction, that indicates something went wrong. It happens when a node makes unnecessary with the intention to do evil.

Algorithm 4: Check Gas Limit

```
// Takes Node’s Id, trustedList as parameters
Begin
    If (Id in trustedList)
        Loop through gasRecord
            If (id == accountNo)
                Return gasRecord[id]
     Else
         Return error()
```

CHAPTER IV

RESULT AND DISCUSSION

This chapter demonstrates the result after implementing blockchain on IoT devices. What's more, it includes machine setup, Performance evaluation of PoW and PoA algorithm, energy consumption by a node while mining block and security features against attacks. Finally, we will also compare the efficiency of our system compared to other existing solutions.

4.1 Machine Setup

A Raspberry Pi 3B+ and a Personal Computer has been used to setup the environment to implement private ethereum blockchain. Table 4.1 shows details configuration of the devices.

Table 4.1: Machine specification.

Machine	**Operating System**	**CPU**	**RAM**	**Tools (Software)**
Raspberry Pi 3B+	Raspbian	ARM Cortex-A53 1.4GHz	1 GB	Geth(go-ethereum), Node.js, Solidity, Truffle
Personal computer	Linux (Ubuntu – 20.04)	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz	12 GB	Geth(go-ethereum), Node.js, Solidity, Truffle

4.2 Performance Evaluation of PoW and PoA

To measure the performance of PoW and PoA protocol, we have calculated the average time to mine blocks by a number of nodes at different hash per second (h/s). It demonstrates how fast PoW gets (the time takes to mine a block) when we change the difficulty level. The change in difficulty level manipulates the probability of finding a nonce. Reducing the difficulty level results in solving a less complex computational puzzle that leads to finding a nonce in a faster time.

Table 4.2: Performance of PoW at different difficulty level by a node.

Difficulty (h/s)	Average time to mine a block (millisecond)
128	58
256	97
512	142
1024	239

To verify if the PoW protocol performs better with multiple devices, we have created several virtual nodes and implemented blockchain on them and figured out the average time taken by nodes to create blocks. Figure 4.3, 4.4, 4.5, 4.6 shows the performance of PoW with multiple devices.

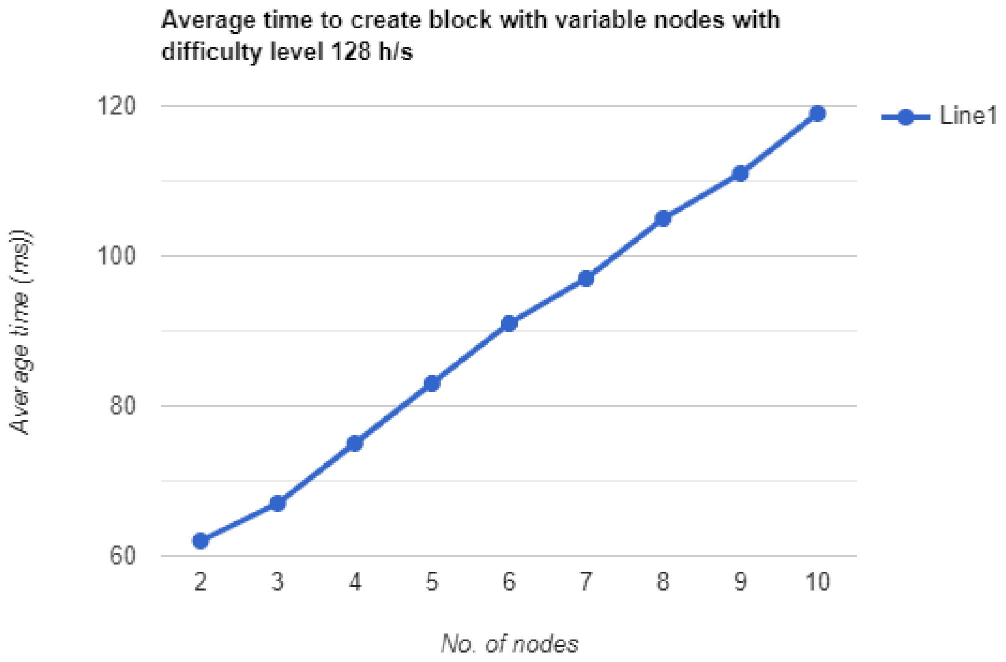


Figure 4.1: Performance of PoW at 128 h/s.

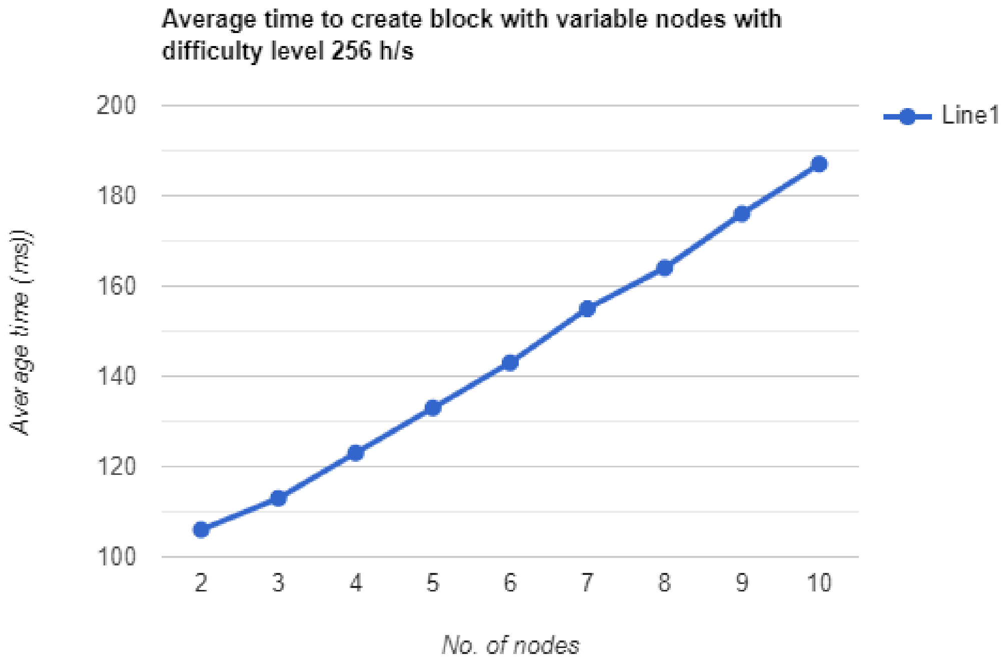


Figure 4.2: Performance of PoW at 256 h/s.

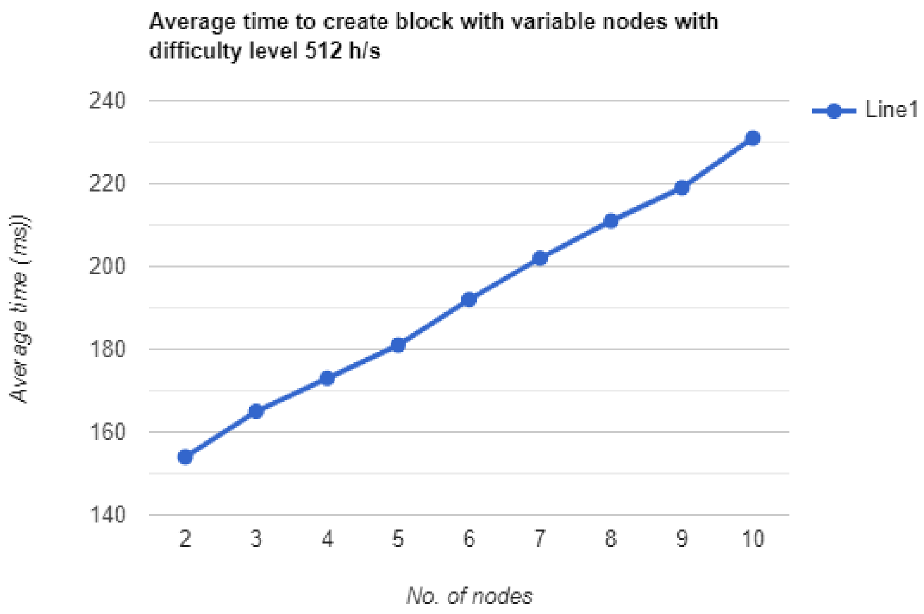


Figure 4.3: Performance of PoW at 512 h/s.

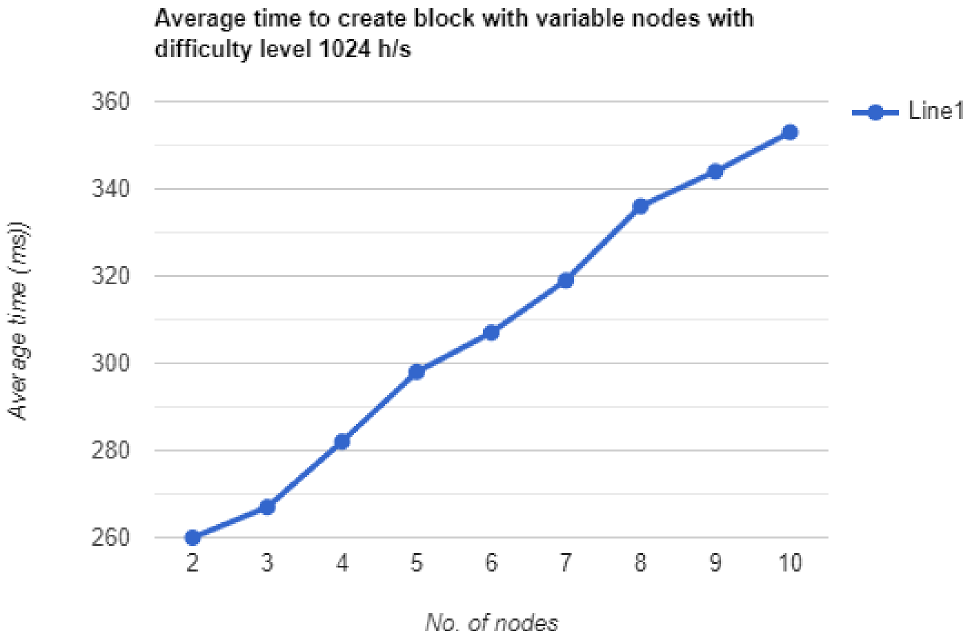


Figure 4.4: Performance of PoW at 1024 h/s.

In the Proof-of-Authority algorithm, since the manager decides the time interval for creating a node, time is constant respective to mining blocks in multiple nodes environment. We have set 300 milliseconds of time interval to mine a block. It also does not have any parameters like difficulty level or solving complex mathematical puzzles of Proof-of-Work, as required parameters are set by the manager. Figure 4.5 shows the block creation time of PoA in multiple nodes environment where after every 300 ms, a block is mined.

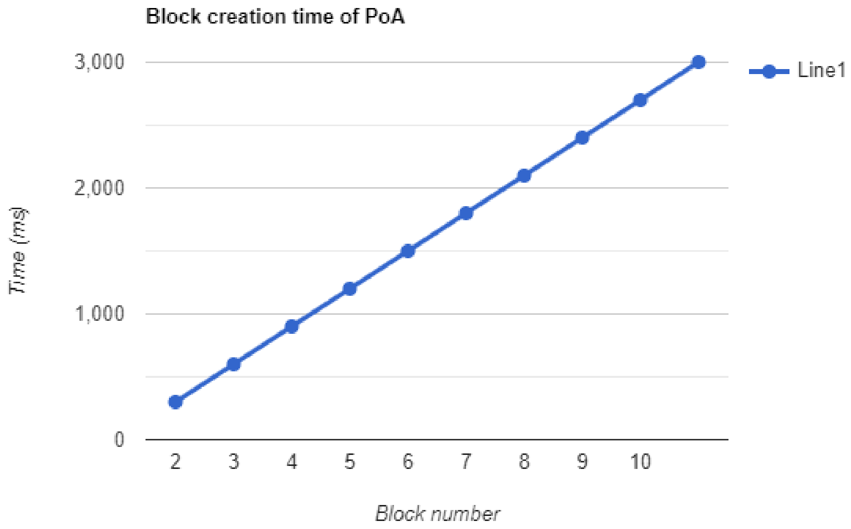


Figure 4.5: Performance of PoA.

4.3 Energy Consumption

Generally, excessive energy consumption by blockchain nodes is a constraint of blockchain. We have measured the energy consumption of a node when it is mining using ‘PowerTop’ software. Since we implemented private blockchain instead of public blockchain and reduced difficulty level, it is obvious how energy efficient our system is from the generated reading. Figure 4.6 shows the energy consumption by a node when it is running PoW, PoA, and in an idle state. Nodes are more energy efficient when the Proof-of-Authority protocol is used compared to the Proof-of-Work protocol due to the absence of finding an acceptable nonce.

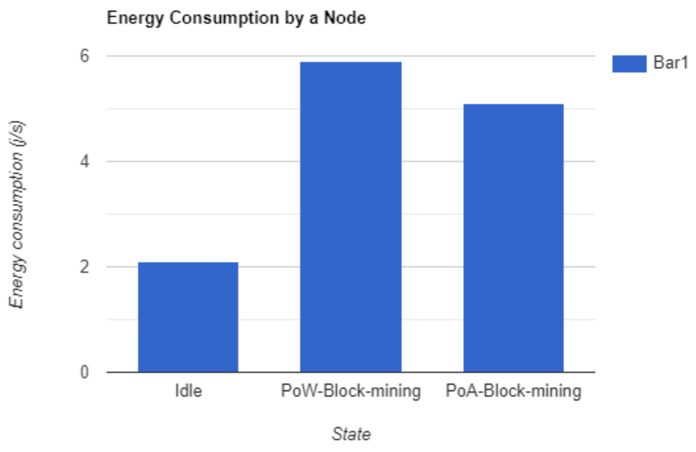


Figure 4.6: Energy consumption by a node.

Figure 4.7 shows the energy consumption by PoW installed node with different difficulty level. Energy consumption increases when difficulty level increase as the node need to solve more complex mathematical puzzle.

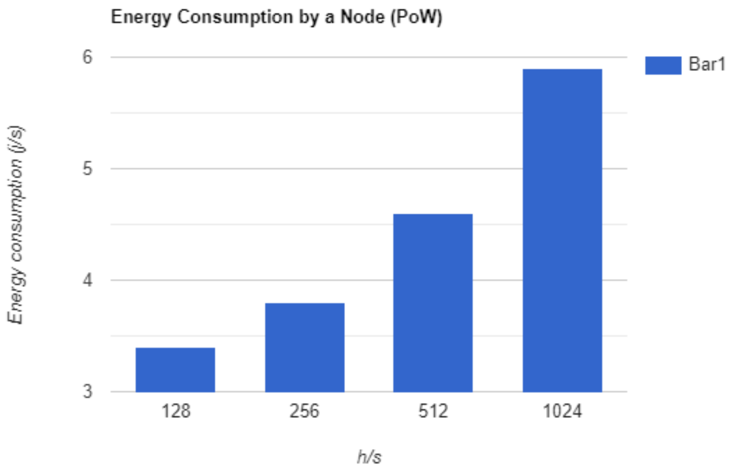


Figure 4.7: Energy consumption by a node at different difficulty level.

CHAPTER V

CONCLUSION

Since people have accelerated using smart devices in their daily life, the security and privacy of smart homes and data become crucial. To counter these issues, blockchain technology has been introduced in the IoT environment. As IoT devices are resource-constant, straightforward blockchain implementation would face issues like latency, computational overhead, and energy consumption. To overcome these setbacks, the proposed system improvised blockchain implementation and added a device authentication mechanism. Encryption and hashing of the data provide security against passive attacks while authentication mechanism and gas limit is used to protect DDoS and Malware attacks. To make our proposed system available for mass use, a benchmark for device configuration against difficulty level is needed, which will help homeowners choose the appropriate IoT for a specific difficulty level. Since the system is private, it saves from storing public node's data which is unnecessary for the users while storing the data generated within the BC network. As the storage capacity of smart devices is very low, further research can be conducted to find a storage convenient smart home solution.

REFERENCES

[1] M. Greenfield, "Statista Inc.," [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/. Accessed [27 2 2023].

[2] J. Hojlo, "IDC," [Online]. Available: https://blogs.idc.com/2021/01/06/future-of-industry-ecosystems-shared-data-and-insights/. [Accessed 27 2 2023].

[3] J. Buchmann, Introduction to cryptography, Springer, 2004.

[4] A. Dorri, S. S. Kanhere, and R. Jurdak, “Blockchain in internet of things: Challenges and solutions,” ArXiv, vol. abs/1608.05187, 2016.

[5] Y. Zhou, M. Han, L. Liu, Y. Wang, Y. Liang, and L. Tian, “Improving iot services in smart-homes using blockchain smart contract,” pp. 81–87, 2018.

[6] A. Dorri, S. S. Kanhere and R. Jurdak, "Towards an Optimized BlockChain for IoT," 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI), Pittsburgh, PA, USA, 2017, pp. 173-178.

[7] Y. Zhou, M. Han, L. Liu, Y. Wang, Y. Liang and L. Tian, "Improving IoT Services in Smart-Home Using Blockchain Smart Contract," 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), Halifax, NS, Canada, 2018, pp. 81-87, doi: 10.1109/Cybermatics_2018.2018.00047.

[8] S. Huh, S. Cho and S. Kim, "Managing IoT devices using blockchain platform," 2017 19th International Conference on Advanced Communication Technology (ICACT), PyeongChang, Korea (South), 2017, pp. 464-467, doi: 10.23919/ICACT.2017.7890132.

[9] C. Yang, E. Mistretta, S. Chaychian, and J. Siau, “Smart home system network architecture,” pp. 174–183, 2016.

[10] B. Hammi, S. Zeadally, R. Khatoun, and J. Nebhen, “Survey on smart homes: Vulnerabilities, risks, and countermeasures,” Computers Security, vol. 117, p. 102677, 2022.

[11] M. Mahnoosh, B. Srinivas, and S. Benjamin, “Personalized speech recognition for internet of things,” in 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), 2015, pp. 369–374.

[12] K. Murad, D. Sadia, J. Sohail, G. Moneeb, G. Hemant, and M. SC, “Context-aware low power intelligent smarthome based on the internet of things,” Computers & Electrical Engineering, vol. 52, pp.208–222, 2016.

[13] L. Changmin, Z. Luca, C. Kwanghee, and C. Hyeong-Ah, “Securing smart home: Technologies, security challenges, and security requirements,” in 2014 IEEE Conference on Communications and Network Security, 2014, pp. 67–72.

[14] B. L. Risteska Stojkoska and K. V. Trivodaliev, “A review of internet of things for smart home: Challenges and solutions,” Journal of Cleaner Production, vol. 140, pp. 1454–1464, 2017.

[15] K. Nikos, P. Eleni, and P. Andreas, “Survey in smart grid and smart home security: Issues, challenges and countermeasures,” IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 1933–1954, 2014.

[16] M. Moniruzzaman, S. Khezr, A. Yassine, and R. Benlamri, “Blockchain for smart homes: Review of current trends and research challenges,” Computers & Electrical Engineering, vol. 83, p. 106585, 05 2020.

[17] B. Bhushan, A. Khamparia, K. M. Sagayam, S. K. Sharma, M. A. Ahad, and N. C. Debnath, “Blockchain for smart cities: A review of architectures, integration trends and future research directions,” Sustainable Cities and Society, vol. 61, p. 102360, 2020.

[18] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and G. Das, “Everything you wanted to know about the blockchain: Its promise, components, processes, and problems,” IEEE Consumer Electronics Magazine, vol. 7, no. 4, pp. 6–14, 2018.

[19] D. Yaga, P. Mell, N. Roby, and K. Scarfone, “Blockchain technology overview,” ArXiv, vol. abs/1906.11078, 2019.

[20] L. Su and N. H. Vaidya, “Reaching approximate byzantine consensus with multi-hop communication,” CoRR, vol. abs/1411.5282, 2014.

[21] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, “A survey on consensus mechanisms and mining strategy management in blockchain networks,” IEEE Access, vol. 7, pp. 22 328–22 370, 2019.

[22] F. Saleh, “Blockchain without waste: Proof-of-stake,” The Review of Financial Studies, vol. 34, 07 2020.

[23] W. Li, S. Andreina, J.-M. Bohli, and G. Karame, “Securing proof-of- stake blockchain protocols,” pp. 297–315, 09 201.

[24] “Delegated proof of stake (dpos),” Available at https://how.bitshares.works/en/master/technology/dpos.html (2022/03/17).

[25] N. Szabo, “The idea of smart contract,” Available at https:// nakamotoinstitute.org/the-idea-of-smart-contracts/, (2022/03/19).

[26] T. Haseeb, Z. Shakir, A. Rashid, H. Mudassar, A.-T. Fadi, and B. Muhammad, “Smart home security: challenges, issues and solutions at different iot layers,” The Journal of Supercomputing, vol. 77, no. 12, pp. 14 053–14 089, 2021.

[27] A. Tejasvi, C. Vinay, S. Biplab, and C. K.-K. Raymond, “Consumer iot: Security vulnerability case studies and solutions,” IEEE Consumer Electronics Magazine, vol. 9, no. 2, pp. 17–25, 2020.

[28] L. Navdeep and K. Raman, “Analysis of lightweight cryptography algorithms for iot communication,” in Congress on Intelligent Systems. Singapore: Springer Singapore, 2021, pp. 397–406.

[29] H. Mahmudul, I. M. Milon, Z. M. I. Islam, and H. MMA, “Attack and anomaly detection in iot sensors in iot sites using machine learning approaches,” Internet of Things, vol. 7, p. 100059, 2019.

[30] T. Hiroaki, Y. Shingo, and A. Takuya, “Consideration of iot structure in mitigation against mirai malware,” in 2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), 2018, pp. 1–3

[31] M. Singh, G. S. Aujla, A. Singh, N. Kumar and S. Garg, "Deep-Learning-Based Blockchain Framework for Secure Software-Defined Industrial Networks," in IEEE Transactions on Industrial Informatics, vol. 17, no. 1, pp. 606-616, Jan. 2021.

[32] S. Badruddoja, R. Dantu, L. Widick, Z. Zaccagni and K. Upadhyay, "Integrating DOTS With Blockchain Can Secure Massive IoT Sensors," 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), New Orleans, LA, USA, 2020, pp. 937-946.

[33] Z. Ahmed, S. M. Danish, H. K. Qureshi and M. Lestas, "Protecting IoTs from Mirai Botnet Attacks Using Blockchains," 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Limassol, Cyprus, 2019, pp. 1-6.

[34] V. Shivangi, R. Jyotsnamayee, M. Janit, V. Saurav, and P. Chetana, “Internet of things (iot): A vision, architectural elements, and security issues,” in 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017, pp. 492–496.

[35] T. Haseeb, Z. Shakir, A. Rashid, H. Mudassar, A.-T. Fadi, and B. Muhammad, “Smart home security: challenges, issues and solutions at different iot layers,” The Journal of Supercomputing, vol. 77, no. 12, pp. 14 053–14 089, 2021.

[36] T. S. Gopal, M. Mallesh, J. G, P. R. L. Eswari, and E. Magesh, “Mitigating mirai malware spreading in iot environment,” in 2018 International Conference on Advances in Computing, communications, and Informatics (ICACCI), 2018, pp. 2226–2230.

[37] A. C, M. H, and O. A, “Defense for distributed denial of service attacks in cloud computing. Tunisia,” 2015.

[38] B. Hammi, D. Guillaume, and K. Rida, “Understanding Bot clouds from a system perspective: A principal component analysis,” in 2014 IEEE Network Operations and Management Symposium (NOMS), 2014, pp. 1–9.

[39] Y. Ryo, H. Daisuke, and N. Yu, “Light-weight DDoS mitigation at network edge with limited resources,” in 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), 2021, pp.1–6.

[40] Kothmayr T, Schmitt C, Hu W, Brünig M, Carle G. DTLS based security and two-way authentication for the Internet of Things. Ad Hoc Networks. 2013 Nov 1;11(8):2710-23.

[41] Yeh HL, Chen TH, Liu PC, Kim TH, Wei HW. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors. 2011 May 2;11:4767-79.

BIOGRAPHICAL SKETCH

Hasibul Alam graduated with a Master of Science in Computer Science program from the University of Texas Rio Grande Valley (UTRGV) on December 14th of 2023. He started his journey at UTRGV as an awardee of the prestigious Presidential Graduate Research Assistantship from 2021. Since the beginning of his journey at the UTRGV, he has been performing research on the Internet of Things and Blockchain under Dr. Sheikh Ariful Islam. During his stay, he has not only worked as a research assistant but also worked as a teaching assistant for several courses in the computer science department.

Being born and raised in Bangladesh, he obtained his Bachelor of Science from Daffodil International University on 21 September 2015. Hasibul also obtained a Master of Science degree in Computer Science from Jahangirnagar University on 29 November 2017. For any help or information, anyone can reach out to him using the following email address: hasibulalammail@gmail.com.