Blockchain-Based Security Management of IoT Infrastructure with Ethereum Transactions

Blockchain-Based Security Management of IoT Infrastructure with Ethereum Transactions

By

SK. Tanzir Mehedi
Student ID: IT-14012
Session: 2013-14

Abdullah Al Mamun Shamim
Student ID: IT-14019
Session: 2013-14

Supervised By

Kawsar Ahmed
Assistant professor
Department of ICT, MBSTU

A thesis is submitted in partial fulfilment of the requirement for the degree of Bachelor of Science (Engg.) in Information and Communication Technology.

Department of Information and Communication Technology
Mawlana Bhashani Science and Technology University
Santosh, Tangail-1902, Bangladesh

Approval

This is to certify that the thesis work submitted by SK. Tanzir Mehedi (IT-14012) and Abdullah Al Mamun Shamim (IT-14019) titled "**Blockchain-Based Security Management of IoT Infrastructure with Ethereum Transactions**" has been approved by the board of examiners for the partial fulfillment of the requirements for the degree of Bachelor of Science (Engineering) in the Department of Information and Communication Technology, Mawlana Bhashani Science and Technology University, Santosh, Tangail-1902, Bangladesh.

Examination Committee

1. Professor Dr. Muhammad Shahin Uddin
   Chairman
   Department of ICT
   MBSTU, Santosh, Tangail-1902

   Chairman
   Examination Committee

2. Professor Dr. Sajjad Waheed
   Department of ICT
   MBSTU, Santosh, Tangail-1902

   Member (Internal)
   Examination Committee

3. Kawsar Ahmed
   Assistant Professor
   Department of ICT
   MBSTU, Santosh, Tangail-1902

   Member (Internal)
   Examination Committee

4. Professor Dr. Kazi Khairul Islam
   Dean
   School of Science and Engineering
   Uttara University, Uttara, Dhaka

   Member (External)
   Examination Committee

Declaration

This thesis focuses on the **"Blockchain-Based Security Management of IoT Infrastructure with Ethereum Transactions"** has been carried out by SK. Tanzir Mehedi and Abdullah Al Mamun Shamim in the Department of Information and Communication Technology, Mawlana Bhashani Science and Technology University, Santosh, Tangail-1902, Bangladesh. We evidence that the thesis work or any part of this work has not submitted anywhere for the award of any degree or diploma. The information have allowed for this document accurate and valid to best of our cognition.

SK. Tanzir Mehedi

ID: IT-14012

Department of ICT, MBSTU

Abdullah Al Mamun Shamim

ID: IT-14019

Department of ICT, MBSTU

Kawsar Ahmed

Assistant Professor

Department of ICT, MBSTU

Dedication

To Our Parents and Respected Teachers.

This thesis is submitted to the Department of Information and Communication Technology in partial fulfillment of the requirements for the degree of Bachelor of Science (Engineering) in Information and Communication Technology.

Contact Information:

SK. Tanzir Mehedi
Student ID: IT-14012
Department of ICT, MBSTU
Santosh, Tangail-1902, Bangladesh
E-mail: tanzirmehedi.ict@gmail.com

Abdullah Al Mamun Shamim
Student ID: IT-14019
Department of ICT, MBSTU
Santosh, Tangail-1902, Bangladesh
E-mail: aamshamim.it@gmail.com

Supervised By:

Kawsar Ahmed
Assistant Professor
Department of ICT, MBSTU
Santosh, Tangail-1902, Bangladesh
E-mail: kawsar.ict@mbstu.ac.bd

Abstract

The blockchain is nothing but a skilled magician one after one it is providing the mankind with wonders in the era of information technology and financial industry. In recent years blockchain has received staggering attention as a means to provide a distributed, definitive and auditable for the Internet of Things (IoT). The predominant Internet of Things (IoT) is moving towards momentous scalability and security challenges. Blockchain technology is extravagant and entangles high bandwidth, prolongation and memory overhead that are not compatible with IoT devices. This paper brings forward a new definitive, intimate as well as lightweight masonry for IoT based blockchain technology which forsakes the memory overhead and centralized system while security and privacy benefits are maintaining. The preliminary investigation method is discussed a standardized IoT infrastructure; where data is stored and access is managed by a decentralized blockchain technology. The following system used terminal devices as network technology and Ethereum as the blockchain platform which produced such kind of backend that ensure high availability, security, and privacy while replacing traditional backend systems. Diametrically, we illustrate the simulation outcome to highlight our approach that significantly related to security and privacy of Blockchain-based IoT applications.

Keywords: Blockchain, Ethereum, Etherscan, Privacy, Security, Internet of Things, Proof-of-Work, Distributed System, Computing, Smart Contacts.

Acknowledgements

All the praise and gratefulness go to the Almighty Allah for giving us perseverance in work, ability and intelligence to complete, as well as, to submit our thesis work successfully without any major problem. Alhamdulillah, we would like to express our open liability to our honorable supervisor Kawsar Ahmed, Assistant Professor, Department of ICT who has supported us at various stage to complete the research work and for giving us his valuable guidance and sagacity, emulating suggestion and advice, continuous encouragement and reliance over the time to produce this report within a very short time. His guidance, punctuality, judgment, and simplicity will be rememberable in our memory.

Finally, we would like to acclaim all other respectable teachers and friends who have helped, inspired and given us mental support at different periods during the completion of our research work.

SK. Tanzir Mehedi
Abdullah Al Mamun Shamim

Preface

This bachelor thesis is schemed based on the results achieved from the laboratory experiment. This is carried out in the Department of Information and Communication Technology (ICT), Faculty of Engineering, at Mawlana Bhashani Science and Technology University (MBSTU), Santosh, Tangail - 1902, Bangladesh.

This thesis includes six chapters which are briefed as follows:

Chapter-1

- Chapter 1: Affords an exhaustive discussion on the significance of the work.

Chapter-2

- Chapter 2: Theme of Blockchain.

Chapter-3

- Chapter 3: IoT-Blockchain Integration Methods.

Chapter-4

- Chapter 4: Proof of Concepts.

Chapter-5

- Chapter 5: Evaluation and Analysis.

Chapter-6

- Chapter 6: Discussion and Conclusion.

Bibliography

Contents

List of Figures

List of Tables

List of Abbreviations

BC	Blockchain
IoT	Internet of Things
DOS	Denial of Service
POW	Proof of Work
P2P	Peer-to-Peer
DS	Distributed System
EVM	Ethereum Virtual Machine
ABI	Application Binary Interface
REN	Rinkeby Etherscan Network
PLC	Power Line Communication
UDP	User Datagram Protocol
JSON	JavaScript Object Notation
VPN	Virtual Private Network
WSN	Wireless Sensor Network
SP	Service Provider
CIA	Confidentiality, Integrity and Availability
PK	Public Key
MFN	Mining Full Node
NMFN	Non-mining Full Node
NMLN	Non-mining Light Node
MAN	Mining Archive Node
BBSMIIET	Blockchain-Based Security Management of IoT Infrastructure with Ethereum Transaction

Chapter 1

Introduction

The introduction typically describes the scope of this document and gives the brief explanation and summary of this document. It also explain certain elements that are important to the essay though explanations are not part of the main text. You have an idea about the following thesis before actually start reading it. The purpose and goals of the following writing is generally followed by the body and conclusion.

1.1 Background

The progression of technology has made us enter into a technologist world where the main target is to deal with the security and privacy of information from various cyber-attacks [1]. Many new networkable devices are becoming knowledgeable nowadays with the help of IoT and blockchain technology. This technology comprehends everything to bring the world closer to our hands. To ensure the security, secrecy and centralization challenges of 50 billion of IoT devices in 2020 [2] there has been increased interest in blockchain technology. Recent networkable IoT devices are less memory, low energy and very lightweight that is why these devices must dedicate most of their gain able energy and executing fundamental functionality which effectively works for the security and privacy of the challenge [3].

In addition, numerous threatening frameworks depend on the centralized system and in this manner not necessarily well-suited for IoT network due to the difficulty of scalability as well as many-to-one nature of traffic. The existing system mostly either disclose irrelevant or imperfect data for protecting user privacy and safety. In such a situation, to think about privacy and safety, IoT system demands a lightweight, scalable, and distributed system [4]. To overcome from this we integrated IoT with blockchain technology to take challenges as a result of it’s a lightweight, scalable, distributed, and private behavior. Blockchain technology not only addresses these scalable, distributed, and private behavior, but also shows a way for integrating all kinds of IoT devices to a common blockchain-based infrastructure [5] [6].

Development of IoT infrastructures suffers from of collecting, storing, and processing data in the cloud server [7]. To select a significant method that enables data transmission from all kinds of IoT devices is another problem. For this reason, we propose a solution to explore what the future of the IT infrastructure will be. In this paper, a blockchain-based IoT infrastructure has described and also described how to integration of various types of terminal devices with it. Our aims are to standardize the way of communication and send data to their data repositories safely as well as create a peer-to-peer, fault-tolerant infrastructure which provided a standard way to query and acquire terminal device (IoT) data for the creation of next-generation products and services. In order to acquire these goals, we have investigated how a peer-to-peer network may be used to store data which enable IoT gateways to push data and interact with other peers by Ethereum Blockchain transaction.

1.2 Problem Statement

Recent reports on IoT and Blockchain have created public interest and concern, and there are important implications for security in these technologies. The need for security in IoT devices is the same and even more to the need for security in all other computing systems to make sure that information is not stolen, modified, or access to it denied.

1.3 Research Question

Based on background that we have conducted in chapter 3 about the topic, security issues in Blockchain based IoT infrastructures, we see that more research must be performed on highlighting possible security threats. We have not been able to find any academic research that conducts a comprehensive security risk assessment to IoT-based smart university highlighting security risks, countermeasures and impacts. To research this gap, the following research questions are defined:

- What is necessity of Blockchain based IoT?
- What are the security threats of Blockchain based IoT System?
- What are the consequences of these threats (Impacts)?
- Are there suitable countermeasures to propose?
- What to recommend the users?

By identifying threats and the impacts we can derive risks because risk consists of both threats and the impacts. It is very important to do research on security issues in Blockchain based IoT system for better understanding and avoiding serious consequences. Without security risk assessment or highlighting threats, it is impossible to provide assurance for the system and justify security measures taken. Further, this new technology, IoT, in order to get broad acceptance among users, security must be better and trust is essential to implement this technology. Thus, security is one of the areas that must be put into the highest priority when implementing the Blockchain based IoT infrastructures.

1.4 Expected Contribution

The research findings will be some useful contribution in providing a better understanding of the security threats about the topic and will make people (users) aware of the potential security and the measures which can be taken to mitigate these vulnerability, concerning their smart devices, either directly or indirectly. Hopefully, the findings will lead to further researches by others within the area of security in Blockchain based IoT Infrastructures.

We propose a solution to explore what the future of the IT infrastructure will be. In this paper, a blockchain-based IoT infrastructure has described and also described how to integration of various types of terminal devices with it. Our aims are to standardize the way of communication and send data to their data repositories safely as well as create a peer-to-peer, fault-tolerant infrastructure which provided a standard way to query and acquire terminal device (IoT) data for the creation of next-generation products and services. In order to acquire these goals, we have investigated how a peer-to-peer network may be used to store data which enable IoT gateways to push data and interact with other peers by Ethereum Blockchain transaction.

The focus of this research will be solely on identification of security issues, suitable countermeasures and impacts identification in Blockchain based IoT infrastructures as well as giving recommendations to the users. For this purpose, scenarios will be provided. The complexity of the smart services is not the scope of this research paper. Simple services will be created to demonstrate user control of the IoT devices and the

communication of data but a comprehensive. The focus of this paper is performing a security risk assessment on critical information assets. Behind the crypto-currency system - is starting to be adopted for ensuring enhanced security and privacy in the Internet of Things (IoT). The primary aim of this article is to investigate the research question, "To what extent can the Blockchain be used in enhancing the overall security of the Internet of Things (IoT) ecosystems?" and draw appropriate conclusions.

1.5 Conclusion and Outline

In order to deal with the increasing number of IoT devices, it is essential to standardize the method of communication among them. Our propose infrastructure is highly committed for the future fourth industrial revolution where automation and data exchange among various manufacturing technologies is the main challenge. To develop our novel infrastructure we discuss the methodology one after one by dividing into six chapters. The next chapter 2 presents a comprehensive overview on blockchain technology and how it is used in the field of IoT. Chapter 3 provides a complete technique of how IoT and Blockchain can be integrated in a body. Chapter 4 provides how to proof this work by different methods. Chapter 5 we will discuss about evaluation and analysis of our method and finally chapter 6 provides a complete discussion of which parts of the IoT and blockchain technology can be improved as well as the challenge of how to tackle upcoming various future automation and data exchange manufacturing technology.

Chapter 2

Theme of Blockchain

2.1 Introduction

The purpose of this chapter is to understand about the Blockchain, Etherscan as well as Ethereum technology why we use in the field of IoT. However, there are still many challenges of blockchain technology such as scalability and security problems waiting to be overcome. This section presents a comprehensive overview on blockchain technology and how it is used in the field of IoT.

2.2 Blockchain

Blockchain, the foundation of Bitcoin, has received extensive attentions recently. Blockchain serves as an immutable ledger which allows transactions take place in a decentralized manner. Blockchain-based applications are springing up, covering numerous fields including financial services, reputation system and Internet of Things (IoT), and so on. However, there are still many challenges of blockchain technology such as scalability and security problems waiting to be overcome. This paper presents a comprehensive overview on blockchain technology. We provide an overview of blockchain architecture firstly and compare some typical consensus algorithms used in different Blockchains. Furthermore, technical challenges and recent advances are briefly listed.

A blockchain is a growing list of records, called blocks, which are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data [8]. Another definition of Blockchain is “The blockchain is an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions but virtually everything of value.” – S. Session in M. C. Conference (2018) [9]. By design, a blockchain is resistant to modification of the data. It is "an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way". For use as a distributed ledger, a blockchain is typically managed by a peer-to-peer network

collectively adhering to a protocol for inter-node communication and validating new blocks. Once recorded, the data in any given block cannot be altered retroactively without alteration of all subsequent blocks, which requires consensus of the network majority. Although blockchain records are not unalterable, Blockchains may be considered secure by design and exemplify a distributed computing system with high Byzantine fault tolerance. Decentralized consensus has therefore been claimed with a blockchain. Blockchain was invented by a person (or group of people) using the name Satoshi Nakamoto in 2008 to serve as the public transaction ledger of the cryptocurrency bitcoin [10].

The invention of the blockchain for bitcoin made it the first digital currency to solve the double-spending problem without the need of a trusted authority or central server. The bitcoin design has inspired other applications, and Blockchains which are readable by the public are widely used by cryptocurrencies. Blockchain is considered a type of payment rail. Private Blockchains have been proposed for business use. Sources such as Computerworld called the marketing of such Blockchains without a proper security model "snake oil". By allowing digital information to be distributed but not copied, blockchain technology created the backbone of a new type of internet. Originally devised for the digital currency, Bitcoin, (Buy Bitcoin) the tech community has now found other potential uses for the technology. A blockchain carries no transaction cost. (An infrastructure cost yes, but no transaction cost.) The blockchain is a simple yet ingenious way of passing information from A to B in a fully automated and safe manner. One party to a transaction initiates the process by creating a block. This block is verified by thousands, perhaps millions of computers distributed around the net. The verified block is added to a chain, which is stored across the net, creating not just a unique record, but a unique record with a unique history. Falsifying a single record would mean falsifying the entire chain in millions of instances. That is virtually impossible. Bitcoin uses this model for monetary transactions, but it can be deployed in many others ways [11]. Think of a railway company. We buy tickets on an app or the web. The credit card company takes a cut for processing the transaction. With blockchain, not only can the railway operator save on credit card processing fees, it can move the entire ticketing process to the blockchain. The two parties in the transaction are the railway company and the passenger. The ticket is a block, which will be added to a ticket blockchain. Just

as a monetary transaction on blockchain is a unique, independently verifiable and unfalsifiable record (like Bitcoin), so can your ticket be. Incidentally, the final ticket blockchain is also a record of all transactions for, say, a certain train route, or even the entire train network, comprising every ticket ever sold, every journey ever taken. But the key here is this: it's free. Not only can the blockchain transfer and store money, but it can also replace all processes and business models which rely on charging a small fee for a transaction or any other transaction between two parties [12]. Most blockchain peers utilize local databases to manage ledger data. The Linux Foundation's Hyper-ledger Fabric has a pluggable architecture and currently supports both Couch DB ad Level DB for the State DB. Additionally, Hyper-ledger Fabric has built-in support for managing off-chain transactional data within the protocol, called private data collections. To date, enterprises have deployed nodes (peers) and their supporting data primarily in the public cloud. We can learn more about this from IBM Blockchain Services. Increasingly, firms are deploying peers and managing supporting data on-premises, as part of their blockchain service and hybrid cloud deployment models. IBM Storage underpins on-premises distributed peer on-chain and off-chain data, as well as public cloud peers with the IBM Blockchain as a Service. IBM Storage Solutions for IBM Blockchain supports unstructured off-chain data with the IBM Spectrum Scale high-performance scale out filesystem, structured off-chain and on-chain data with the NVMe-accelerated Flash System 9100, and support with IBM Cloud Private or bare-metal deployments with IBM ZLinux. IBM also enables storage and data protection with IBM Cloud Object Storage, backup-and-recovery via snapshot or continuous data synchronization with native-storage snapshots and with IBM Spectrum Protect Plus.

2.2.1 History of Blockchain

Nowadays cryptocurrency has become a buzzword in both industry and academia. As one of the most successful cryptocurrency, Bitcoin has enjoyed a huge success with its capital market reaching 10 billion dollars in 2016 [13]. With a specially designed data storage structure, transactions in Bitcoin network could happen without any third party and the core technology to build Bitcoin is blockchain, which was first proposed in 2008 and implemented in 2009 [14]. Blockchain could be regarded as a public ledger and all committed transactions are stored in a list of blocks. This chain grows as new blocks are appended to it continuously. Asymmetric cryptography and distributed

consensus algorithms have been implemented for user security and ledger consistency. The blockchain technology generally has key characteristics of decentralization, persistency, anonymity and auditability. With these traits, blockchain can greatly save the cost and improve the efficiency.

Since it allows payment to be finished without any bank or any intermediary, blockchain can be used in various financial services such as digital assets, remittance and online payment [15]. Additionally, it can also be applied into other fields including smart contracts, public services Internet of Things (IoT), reputation systems and security services. Those fields favor blockchain in multiple ways. First of all, blockchain is immutable. Transaction cannot be tampered once it is packed into the blockchain. Businesses that require high reliability and honesty can use blockchain to attract customers. Besides, blockchain is distributed and can avoid the single point of failure situation. As for smart contracts, the contract could be executed by miners automatically once the contract has been deployed on the blockchain.

Although the blockchain technology has great potential for the construction of the future Internet systems, it is facing a number of technical challenges. Firstly, scalability is a huge concern. Bitcoin block size is limited to 1 MB now while a block is mined about every ten minutes. Subsequently, the Bitcoin network is restricted to a rate of 7 transactions per second, which is incapable of dealing with high frequency trading. However, larger blocks means larger storage space and slower propagation in the network. This will lead to centralization gradually as less users would like to maintain such a large blockchain. Therefore the tradeoff between block size and security has been a tough challenge. Secondly, it has been proved that miners could achieve larger revenue than their fair share through selfish mining strategy. Miners hide their mined blocks for more revenue in the future. In that way, branches could take place frequently, which hinders blockchain development. Hence some solutions need to be put forward to fix this problem. Moreover, it has been shown that privacy leakage could also happen in blockchain even users only make transactions with their public key and private key [16]. Furthermore, current consensus algorithms like proof of work or proof of stake are facing some serious problems. For example, proof of work wastes too much electricity energy while the phenomenon that the rich get richer could appear in the proof of stake consensus process.

There is a lot of literature on blockchain from various sources, such as blogs, wikis, forum posts, codes, conference proceedings and journal articles. Tschorsch et al made a technical survey about decentralized digital currencies including Bitcoin. Compared to, our paper focuses on blockchain technology instead of digital currencies. Nomura Research Institut made a technical report about blockchain. Contrast to our paper focuses on state-of-art blockchain researches including recent advances and future trends. The rest of this paper is organized as follows. Section II introduces blockchain architecture. Section III shows typical consensus algorithms used in blockchain. Section IV summarizes the technical challenges and the recent advances in this area. Section V discusses some possible future directions and section VI concludes the paper.

Blockchain is a sequence of blocks, which holds a complete list of transaction records like conventional public ledger [17] .With a previous block hash contained in the block header, a block has only one parent block. It is worth noting that uncle blocks (children of the block's ancestors) hashes would also be stored in Ethereum blockchain. The first block of a blockchain is called genesis block which has no parent block. We then explain the internals of blockchain in details.

2.2.2 Block

A block consists of the block header and the block body as shown in Figure 2.1. In particular, the block header includes:

1) Block version: indicates which set of block validation rules to follow.
2) Merkle tree root hash: the hash value of all the transactions in the block.
3) Timestamp: current time as seconds in universal time since January 1, 1970 [18].
4) nBits: target threshold of a valid block hash.
5) Nonce: an 4-byte field, which usually starts with 0 and increases for every hash calculation,
6) Parent block hash: a 256-bit hash value that points to the previous block [19].

The block body is composed of a transaction counter and transactions. The maximum number of transactions that a block can contain depends on the block size and the size of each transaction. Blockchain uses symmetric cryptography mechanism to validate the authentication of transactions [20].

2.2.3 Digital Signature

Each user owns a pair of private key and public key. The private key that shall be kept in confidentiality is used to sign the transactions. The digital signed transactions are broadcasted throughout the whole network. The typical digital signature is involved with two phases: signing phase and verification phase. For instance, a user Alice wants to send another user Bob a message.

(1) In the signing phase, Alice encrypts her data with her private key and sends Bob the encrypted result and original data.

(2) In the verification phase, Bob validates the value with Alice's public key. In that way, Bob could easily check if the data has been tampered or not.

The typical digital signature algorithm used in Blockchains is the elliptic curve digital signature algorithm (ECDSA).

2.2.4 Key Characteristics of Blockchain

In summary, blockchain has following key characteristics.

- **Decentralization:** In conventional centralized transaction systems, each transaction needs to be validated through the central trusted agency (e.g., the central bank), inevitably resulting to the cost and the performance bottlenecks at the central servers. Contrast to the centralized mode, third party is no longer needed in blockchain. Consensus algorithms in blockchain are used to maintain data consistency in distributed network.

- **Persistency:** Transactions can be validated quickly and invalid transactions would not be admitted by honest miners. It is nearly impossible to delete or rollback transactions once they are included in the blockchain. Blocks that contain invalid transactions could be discovered immediately.

- **Anonymity:** Each user can interact with the blockchain with a generated address, which does not reveal the real identity of the user. Note that blockchain cannot guarantee the perfect privacy preservation due to the intrinsic constraint.

- **Auditability:** Bitcoin blockchain stores data about user balances based on the Unspent Transaction Output (UTXO) model: Any transaction has to refer to some previous unspent transactions. Once the current transaction is recorded into the blockchain, the state of those referred unspent transactions switch from unspent to spend. So transactions could be easily verified and tracked.

2.2.5 Taxonomy of blockchain systems

Current blockchain systems are categorized roughly into three types: public blockchain, private blockchain and consortium blockchain [21]. In public blockchain, all records are visible to the public and everyone could take part in the consensus process. Differently, only a group of pre-selected nodes would participate in the consensus process of a consortium blockchain. As for private blockchain, only those nodes that come from one specific organization would be allowed to join the consensus process.

A private blockchain is regarded as a centralized network since it is fully controlled by one organization. The consortium blockchain constructed by several organizations is partially decentralized since only a small portion of nodes would be selected to determine the consensus. The comparison among the three types of blockchain is discussed below.

- **Consensus determination:** In public blockchain, each node could take part in the consensus process. And only a selected set of nodes are responsible for validating the block in consortium blockchain. As for private chain, it is fully controlled by one organization and the organization could determine the final consensus.

- **Read permission:** Transactions in a public blockchain are visible to the public while it depends when it comes to a private blockchain or a consortium blockchain.

- **Immutability:** Since records are stored on a large number of participants, it is nearly impossible to tamper transactions in a public blockchain. Differently, transactions in a private blockchain or a consortium blockchain could be tampered easily as there are only limited number of participants.

- **Efficiency:** It takes plenty of time to propagate transactions and blocks as there are a large number of nodes on public blockchain network. As a result, transaction throughput is limited and the latency is high. With fewer validators, consortium blockchain and private blockchain could be more efficient.

- **Centralized:** The main difference among the three types of Blockchains is that public blockchain is decentralized, consortium blockchain is partially centralized and private blockchain is fully centralized as it is controlled by a single group.

- **Consensus process:** Everyone in the world could join the consensus process of the public blockchain. Different from public blockchain, both consortium blockchain and private blockchain are permissioned.

Since public blockchain is open to the world, it can attract many users and communities are active. Many public Blockchains emerge day by day. As for consortium blockchain, it could be applied into many business applications. Currently Hyper ledger is developing business consortium blockchain frameworks [22]. Ethereum also has provided tools for building consortium Blockchains.

2.2.6 Core Components

A blockchain consists of blocks, which are linked between them and collections of timestamped transactions. In this technology that allows nodes to exchange data by constructs a transaction. Each transaction depends on another transaction where one transaction outputs are referred in another transaction as inputs thus creating chain among them [23]. The first block is called generic block and some extraordinary block in the network known as miners try to solve a cryptographic puzzle named Proof of Work.

Thus participating nodes build a trusted network over untrusted participants in the network. New transactions are verified by all participating nodes which omit the necessity of the central dependency and propose distributed management system. Each block contains the hash of its previous block which ensures constancy of the whole transaction thus alternation of any block from the network is unattainable.

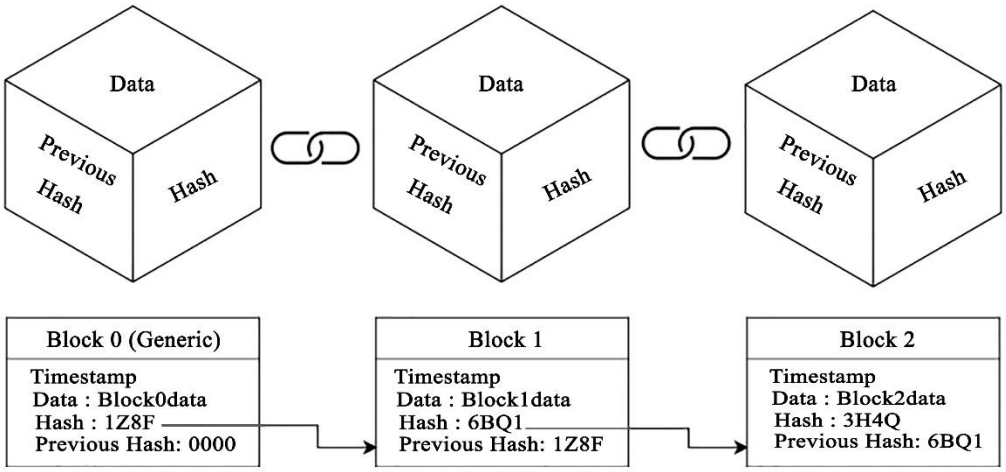

Figure-2.1 Structure of Blockchain

If one transaction is valid then the transaction are continuously stored in the public unchanging blockchain network which can be accessed by any node. All transactions among this network are signature using public-key cryptography thus their authenticity nature is accomplished [24].

2.2.7 Transaction

The smart devices may communicate directly with each other or with entities external to the smart home. Each device inside the home may request data from another internal device to offer certain services, e.g., the light bulb requests data from the motion sensor to turn on the lights automatically when someone enters the room. To achieve user control over smart university transactions, a shared key should be allocated by the miner to devices which need to directly communicate with each other. To allocate the key, the miner checks the policy header or asks for permission from the owner and then distributes a shared key between devices. After receiving the key, devices communicate directly as long as their key is valid. To deny the grant permission, the miner marks the distributed key as invalid by sending a control message to devices. The benefits of this method is twofold: on one hand, the miner (and so the owner) has a list of devices that share data, and on the other, the communications between devices are secured with a shared key [25].

Storing data on the local storage by devices is the other possible transaction flow inside the university network. To store data locally, each device needs to be authenticated to the storage that is done using a shared key. To grant the key, the device needs to send a request for the miner and if it has storing permission, the miner generates a shared key and sends the key for the device and the storage. By receiving the key, the local storage generates a starting point that contains the shared key. Having the shared key, the device can store data directly in the local storage.

The devices may demand to store data on the cloud storage that is known as store transaction. Storing data in the cloud is an anonymous process that is discussed in [26]. To store data the requester needs a starting point that contains a block-number and a hash used for anonymous authentication purpose. The cloud storage may be either owned and managed by the SP (e.g. Nest thermostat) or paid for and managed by the network owner (e.g. Dropbox). In the former instance, the miner requests for the starting point by generating a signed transaction with the device key. In the latter case, payment is done through Bitcoin. In either storage type, after receiving a request the storage creates a starting point and sends it to the miner. When a device needs to store data on the cloud storage, it sends data and the request to the miner. By receiving the request, the miner authorizes the device for storing data on the cloud storage. If the device has been authorized, the miner extracts the last block-number and hash from the local BC, and creates a store transaction and sends it along with the data to the storage. After storing data, the cloud storage returns the new block-number to the miner that is used for further storing transactions.

The other possible transactions are access and monitor transactions. These transactions are mainly generated by either the network owner to monitor the network when he is outside or by SPs to process devices data for personalized services. By receiving an access transaction from nodes in the overlay, the miner checks whether the requested data is on the local or the cloud storage. If data is stored in the local storage, the miner requests data from the local storage and sends it to the requester. On the other hand, if the data is stored in the cloud, the miner either requests data from the cloud storage and sends it to the requester, or sends the last block-number and hash to the requester [27]. The latter scenario empowers the requester to read entire data stored by the device in cloud storage and is suitable when the stored data are for a unique device.

Otherwise, the user's privacy might be endangered as part of a linking attack which is discussed later.

By receiving a monitor transaction, the miner sends current data of the requested device to the requester. If a requester is allowed to receive data for a period of time then the miner sends data periodically until the requester sends a close request to the miner and abolish the transaction. The monitor transaction enables home owners to watch cameras or other devices in which send periodic data. In order to avoid overhead or possible attacks, the owner should define a threshold in minutes for the periodic data. If the time in which the miner is sending data for the requester reaches to the threshold, then the connection is terminated by the miner.

2.2.8 Local BC

In each smart university, there is a local private BC that keeps track of transactions and has a policy header to enforce user's policy for incoming and outgoing transactions. Starting from the genesis transaction, each devices transactions are chained together as an immutable ledger in the BC. Each block in the local BC contains two headers that are block header and policy header. The block header has the hash of the previous block to keep the BC immutable [28].

The policy header is used for authorizing devices and enforcing owners control policy over his home. As shown in the top right corner of Figure 2.2, the policy header has four parameters. The Requester parameter refers to the requester PK in the received overlay transaction. For local devices, this field is equal to the Device ID as shown in the fourth row of the proposed policy header in Figure 2.2. The second column in the policy header, indicates the requested action in the transaction, which can be: store to store data locally, store cloud to store data on the cloud storage, access to access stored data of a device, and monitor to access real-time data of a particular device. The third column in the policy header is the ID of a device inside the smart university, and finally, the last column indicates the action that should be done for the transaction that matches with the previous properties. Besides the headers, each block contains a number of transactions. For each transaction five parameters are stored in the local BC as shown in the top left corner of the Figure 2.2. The first two parameters are used to chain

transactions of the same device to each other and identify each transaction uniquely in the BC.

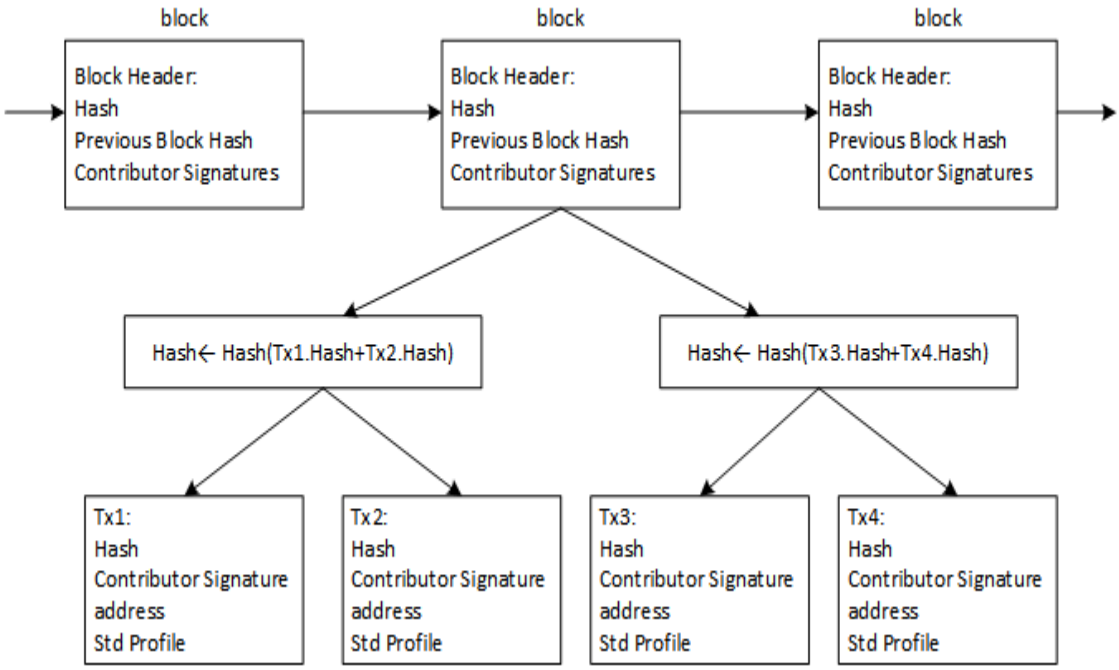

Figure-2.2 Structure of transaction

The transactions corresponding device ID is inserted on the third field. Transaction type refers to the type of transaction that can be genesis, access, store, or monitor transactions. The transaction is stored on the fifth field if it comes from the overlay network, otherwise, this filed is kept blank. The local BC is kept and managed by a local miner.

2.2.9 Miner

Miner is a device that centrally processes incoming and outgoing transactions to and from the smart university network. In addition the miner also accomplishes the following additional functions: generating genesis transactions, distributing and updating keys, changing the transactions structure, and forming and managing the cluster [29]. The miner collects all transactions into a block and appends the full block to the BC. To provide additional capacity, the miner manages a local storage.

2.2.10 Local Storage

Local storage is a storing device e.g. backup drive that is used by devices to store data locally. This storage can be integrated with the miner or it can be a separate device. The storage uses a First-in-First-out (FIFO) method to store data and stores each devices data as a ledger chained to the devices starting point.

2.3 Ethereum

Ethereum is the blockchain-based distributed infrastructure for computing or compiles code fragments functionality that may interact with each other [30]. Ethereum enables to build decentralized applications where Ethereum wallet used as a gateway [31]. Ethereum helps to write, deploy and use smart contracts. Ethereum enhances the scripting abilities of programming language aiming to create a programming environment. Thus, Ethereum becomes to be a distributed application platform to take advantage of blockchain technology where users can choose to customized format for transactions. Ethereum smart contracts programming are executed in a virtual machine which is called Ethereum Virtual Machine (EVM) [32]. Then the customized function can be accessed using the address of the contract and its application binary interface (ABI) file. And Etherscan is the prime block searcher application program interface for the Ethereum Blockchain. A block explorer is materially a search engine that allows users to easily find, ensure and verify each transaction that has taken place on the Ethereum Blockchain.

At its simplest, Ethereum is an open software platform based on blockchain technology that enables developers to build and deploy decentralized applications. Like Bitcoin, Ethereum is a distributed public blockchain network. Although there are some significant technical differences between the two, the most important distinction to note is that Bitcoin and Ethereum differ substantially in purpose and capability. Bitcoin offers one particular application of blockchain technology, a peer to peer electronic cash system that enables online Bitcoin payments. While the Bitcoin blockchain is used to track ownership of digital currency (bitcoins), the Ethereum blockchain focuses on running the programming code of any decentralized application. In the Ethereum blockchain, instead of mining for bitcoin, miners work to earn Ether, a type of crypto token that fuels the network.

Beyond a tradeable cryptocurrency, Ether is also used by application developers to pay for transaction fees and services on the Ethereum network. Because decentralized applications run on the blockchain, they benefit from all of its properties. Mutability – A third party cannot make any changes to data. Corruption & tamper proof – Apps are based on a network form end around the principle of consensus, making censorship impossible. Secure – With no central point of failure and secured using cryptography, applications are well protected against hacking attacks and fraudulent activities. Zero downtime – Apps never go down and can never be switched off.

2.4 Etherscan

Etherscan is the leading Block Explorer for the Ethereum Blockchain. A Block Explorer is basically a search engine that allows users to easily lookup, confirm and validate transactions that have taken place on the Ethereum Blockchain. We are independently operated and developed by a team of individuals who are truly passionate and excited about the kinds of decentralized information and infrastructure applications that Ethereum makes possible. Etherscan is a block explorer and analytics platform for Ethereum, a decentralized smart contracts platform. Etherscan hosts a collection of web-based tools for exploring the public Ethereum network, based on the transactions that have been confirmed on the Ethereum blockchain.

As a blockchain explorer, Etherscan can only provide and display information on transactions that occur on the Ethereum blockchain. Etherscan does not process transactions and is unable to troubleshoot transaction failures. Etherscan uses 11 technology products and services including Google Analytics, Google Tag Manager, and Bootstrap.

2.5 Smart Contact

Smart contract is just a phrase used to describe a computer code that can facilitate the exchange of money, content, property, shares, or anything of value. When running on the blockchain a smart contract becomes like a self-operating computer program that automatically executes when specific conditions are met. Because smart contracts run on the blockchain, they run exactly as programmed without any possibility of censorship, downtime, and fraud or third-party interference. The aim of smart contracts is to provide security that is superior to traditional contract law and to reduce other transaction costs associated with contracting.

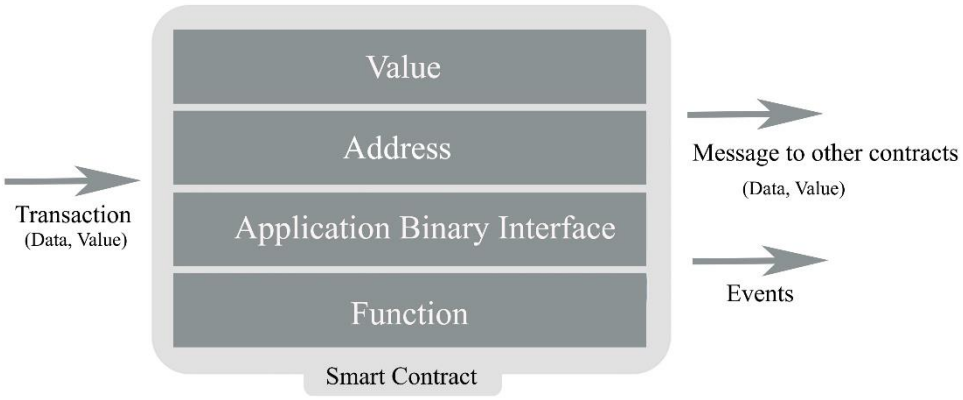

Figure-2.3 Structure of Smart Contact

A blockchain-based smart contract is visible to all users of said blockchain. However, this leads to a situation where bugs, including security holes, are visible to all yet may not be quickly fixed. Smart contracts are often written in a programming language called “Solidity”, a language similar to JavaScript and C++. Other languages for writing smart contracts include Viper and Bamboo. Before Solidity was released, other languages like Serpent (deprecated) and Mutan (deprecated) were used.

2.6 Ethereum Virtual Machine

Ethereum core innovation, the Ethereum Virtual Machine (EVM) is a Turing complete software that runs on the Ethereum network. It enables anyone to run any program, regardless of the programming language given enough time and memory. The

Ethereum Virtual Machine makes the process of creating blockchain applications much easier and efficient than ever before. Instead of having to build an entirely original blockchain for each new application, Ethereum enables the development of potentially thousands of different applications all on one platform. The Ethereum Virtual Machine possesses its own programming language, known as the 'EVM bytecode' [33]. When code is written in higher-level programming languages such as Ethereum's contract-orientated language Solidity, this code can then be compiled to the EVM bytecode, so that the Ethereum Virtual Machine can understand what has been written.

2.7 Challenges & Recent Advances

Despite the great potential of blockchain, it faces numerous challenges, which limit the wide usage of blockchain. We enumerate some major challenges and recent advances as follows.

2.7.1 Scalability

With the amount of transactions increasing day by day, the blockchain becomes bulky. Each node has to store all transactions to validate them on the blockchain because they have to check if the source of the current transaction is unspent or not. Besides, due to the original restriction of block size and the time interval used to generate a new block, the Bitcoin blockchain can only process nearly 7 transactions per second, which cannot fulfill the requirement of processing millions of transactions in real-time fashion. Meanwhile, as the capacity of blocks is very small, many small transactions might be delayed since miners prefer those transactions with high transaction fee.

There are a number of efforts proposed to address the scalability problem of blockchain, which could be categorized into two types:

- **Storage optimization of blockchain**: Since it is harder for node to operate full copy of ledger, Bruce proposed a novel cryptocurrency scheme, in which the old transaction records are removed (or forgotten) by the network. A database named account tree is used to hold the balance of all non-empty addresses. Besides lightweight client could also help fix this problem. A. Biryukob was proposed to provide another way allowing lightweight clients to exist [34].

- **Redesigning blockchain:** In Bitcoin-NG (Next Generation) was proposed by F. Tschorsch and B. Scheuermann [35]. The main idea of Bitcoin-NG is to decouple conventional block into two parts: key block for leader election and micro block to store transactions. The protocol divides time into epochs. In each epoch, miners have to hash to generate a key block. Once the key block is generated, the node becomes the leader who is responsible for generating micro blocks. Bitcoin-NG also extended the heaviest (longest) chain strategy in which micro blocks carry no weight. In this way, blockchain is redesigned and the tradeoff between block size and network security has been addressed.

2.7.2 Privacy Leakage

Blockchain can preserve a certain amount of privacy through the public key and private key. Users transact with their private key and public key without any real identity exposure. However, it is shown in that blockchain cannot guarantee the transactional privacy since the values of all transactions and balances for each public key are publicly visible [36]. Besides, the recent study has shown that a user's Bitcoin transactions can be linked to reveal user's information [37]. Moreover, D. Johnson et al. presented a method to link user pseudonyms to IP addresses even when users are behind Network Address Translation (NAT) or firewalls [38]. In each client can be uniquely identified by a set of nodes it connects to [39]. However, this set can be learned and used to find the origin of a transaction. Multiple methods have been proposed to improve anonymity of blockchain, which could be roughly categorized into two types:

- **Mixing**: In blockchain, user's addresses are pseudonymous. But it is still possible to link addresses to user real identity as many users make transactions with the same address frequently. Mixing service is a kind of service which provides anonymity by transferring funds from multiple input addresses to multiple output addresses. For example, user Alice with address A wants to send some funds to Bob with address B. If Alice directly makes a transaction with input address A and

output address B, relationship between Alice and Bob might be revealed. So Alice could send funds to a trusted intermediary Carol. Then Carol transfer funds to Bob with multiple inputs c1, c2, c3, etc., and multiple output d1, d2, B, d3, etc. Bob's address B is also contained in the output addresses. So it becomes harder to reveal relationship between Alice and Bob. However, the intermediary could be dishonest and reveal Alice and Bob's private information on purpose. It is also possible that Carol transfers Alice's funds to her own address instead of Bob's address. In Hyper ledger Project provides a simple method to avoid dishonest behavior's [40]. The intermediary encrypts users' requirements including funds amount and transfer date with its private key. Then if the intermediary did not transfer the money, anybody could verify that the intermediary cheated. However, theft is detected but still not prevented. Coin join depends on a central mixing server to shuffle output addresses to prevent theft [41]. And inspired by Coin join, Coin Shuffle uses decryption mix nets for address shuffling.

- **Anonymous:** In Zero coin, zero-knowledge proof is used. Miners do not have to validate a transaction with digital signature but to validate coins belong to a list of valid coins [42]. Payment's origin are unlinked from transactions to prevent transaction graph analyses. But it still reveals payments' destination and amounts. S. King and S. Nadal was proposed to address this problem [43]. In Zero cash, zero-knowledge Succinct Non-interactive Arguments of Knowledge (zk-SNARKs) is leveraged. Transaction amounts and the values of coins held by users are hidden.

2.7.3 Selfish Mining

Blockchain is susceptible to attacks of colluding selfish miners. In particular, Bitshares [online portal] showed that the network is vulnerable even if only a small portion of the hashing power is used to cheat [44]. In selfish mining strategy, selfish miners keep their mined blocks without broadcasting and the private branch would be revealed to the public only if some requirements are satisfied. As the private branch is longer than the current public chain, it would be admitted by all miners. Before the private blockchain publishment, honest miners are wasting their resources on a useless branch while selfish miners are mining their private chain without competitors.

In miners could amplify its gain by non-trivially composing mining attacks with network-level eclipse attacks. The trail-stubbornness is one of the stubborn strategy that miners still mine the blocks even if the private chain is left behind. Yet in some cases, it can result in 13% gains in comparison with a non-trail-stubborn counterpart shows that there are selfish mining strategies that earn more money and are profitable for smaller miners compared to simple selfish mining [45]. But the gains are relatively small. Furthermore, it shows that attackers with less than 25% of the computational resources can still gain from selfish mining. To help fix the selfish mining problem, J. kwon presented a novel approach for honest miners to choose which branch to follow [46]. With random beacons and timestamps, honest miners would select more fresh blocks [47]. However, is vulnerable to forgeable timestamps. ZeroBlock builds on the simple scheme: Each block must be generated and accepted by the network within a maximum time interval [48]. Within ZeroBlock, selfish miners cannot achieve more than its expected reward.

2.8 Possible Future Directions

Blockchain has shown its potential in industry and academia. We discuss possible future directions with respect to four areas: blockchain testing, stop the tendency to centralization, big data analytics and blockchain application.

2.8.1 Blockchain testing

Recently different kinds of Blockchains appear and over 700 cryptocurrencies are listed up to now [49]. However, some developers might falsify their blockchain performance to attract investors driven by the huge profit. Besides that, when users want to combine blockchain into business, they have to know which blockchain fits their requirements. So blockchain testing mechanism needs to be in place to test different Blockchains. Blockchain testing could be separated into two phases: standardization phase and testing phase. In standardization phase, all criteria have to be made and agreed. When a blockchain is born, it could be tested with the agreed criteria to valid if the blockchain works fine as developers claim. As for testing phase, blockchain testing needs to be performed with different criteria.

For example, an user who is in charge of online retail business cares about the throughput of the blockchain, so the examination needs to test the average time from a user send a transaction to the transaction is packed into the blockchain, capacity for a blockchain block and etc.

2.8.2 Stop the tendency to centralization

Blockchain is designed as a decentralized system. However, there is a trend that miners are centralized in the mining pool. Up to now, the top 5 mining pools together owns larger than 51% of the total hash power in the Bitcoin network [50]. Apart from that, selfish mining strategy showed that pools with over 25% of total computing power could get more revenue than fair share [51]. Rational miners would be attracted into the selfish pool and finally the pool could easily exceed 51% of the total power. As the blockchain is not intended to serve a few organizations, some methods should be proposed to solve this problem.

2.8.3 Big data analytics

Blockchain could be well combined with big data. Here we roughly categorized the combination into two types: data management and data analytics. As for data management, blockchain could be used to store important data as it is distributed and secure. Blockchain could also ensure the data is original. For example, if blockchain is used to store patient's health information, the information could not be tampered and it is hard to steal those private information. When it comes to data analytics, transactions on blockchain could be used for big data analytics. For example, user trading patterns might be extracted. Users can predict their potential partners' trading behaviors with the analysis.

2.8.4 Blockchain applications

Currently most Blockchains are used in the financial domain, more and more applications for different fields are appearing [52]. Traditional industries could take blockchain into consideration and apply blockchain into their fields to enhance their systems. For example, user reputations could be stored on blockchain. At the same time, the up-and-coming industry could make use of blockchain to improve

performance. For example, Arcade City, a ridesharing startup offers an open marketplace where riders connect directly with drivers by leveraging blockchain technology.

A smart contract is a computerized transaction protocol that executes the terms of a contract [53]. It has been proposed for long time and now this concept can be implemented with blockchain. In blockchain, smart contract is a code fragment that could be executed by miners automatically. Smart contract has transformative potential in various fields like financial services and IoT.

2.9 Conclusion

Blockchain is a growing list of records, called blocks, which are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data. Blockchain has been in a lot of buzz these days. And that is mainly because it is backbone of the very famous cryptocurrency in the world - the Bitcoin. Many Governments and leading Banks have decided to bring many of their conventional transactions based on Blockchain concept. The applications and potential of this framework is huge and is considered to be changing the way transactions are made in various domains. So it is very much helpful in the field of IoT environment. The next chapter provides a complete technique of how IoT and Blockchain can be integrated in a body.

Chapter 3

Integration Method

3.1 Introduction

The aim of this chapter is to provide a brief overview of IoT-Blockchain Integration technique with a focus on their enabling technologies, application areas, structures and architectures. Our goal is not to give a point by point clarification of every subject, but to give the reader the basic principles and a brief overview of every subject, as well as the bibliography to be checked in case someone wishes to deepen on some aspects of the subject.

3.2 An Overview of our Method

Terminal devices and gateways with blockchain technology can be integrated into different process depending on the memory capabilities and power consumption. The terminal devices are always-on or battery powered which are always communicating with a gateway connected to the internet which is also always-on. Now, we discuss one of the following integration techniques.

3.3 Integration Technique

Now, we are going to control the terminal device with my Ethereum wallet. Firstly, use a smart contract to control terminal devices which are posting data onto the blockchain network. Gateway is enabled that is always on and it is connected via Wi-Fi to the internet and this gateway is running an instance to go Ethereum node and so this chip is interacting with that node and sending RPC calls to that Ethereum network.

We used the Rinkeby Etherscan Network which is the test Ethereum network. We actually prefer Rinkeby network for proof of authority private networks and so that's one of the things if we were to interact with this it's one of the first things that we want for IoT security and privacy. How to interact with contracts and the contract that will be interacting with terminal devices for this we need a smart contract address and

an API in order to interact with it which is provided by Etherscan. Then the user should be able to access select functions to control terminal devices.

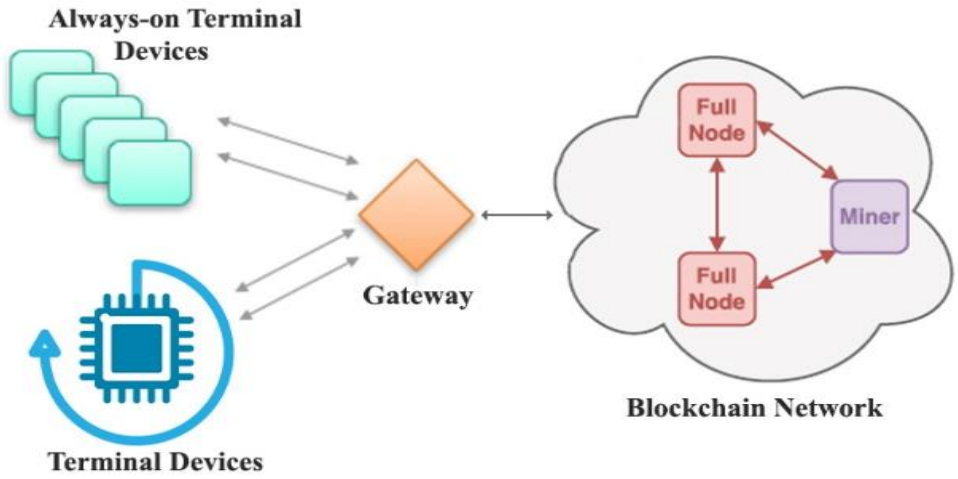

Figure-3.1 IoT and Blockchain Integration Technique

3.4 Gateway as a full blockchain node

Routing data among whole network and verifying integrity together IoT gateway works as a full node because of verifying each block when the transaction is begun. Integration technique is comparatively easy because no changes are required in the communication link when the end devices communicate with Ethereum network. With the gateways total computing power for defending the integrity of the system, it is possible to achieve a trustless IoT infrastructure.

3.5 Terminal device as a thin client

In a thin-client terminal device, all of the application processing and data management is carried out on the Ethereum network. The terminal IoT devices are simply responsible for running the presentation. It could be operated in thin client mode if the terminal devices are not connected with battery powered as well as always off. Integration is relatively easy however the weakness here is that there should be other full nodes to defend the integrity of the system. A trustless infrastructure can still be achieved, but only with full nodes operating at the cloud side.

3.6 Terminal device as a fat client

In this model, the Ethereum network is only responsible for data management. The terminal devices implement the application logic and the interactions. This way, the terminal device will interact with an Ethereum node without any memory capacity or computational functional requirements.

3.7 Conclusion

It is clear that the differences between traditional and blockchain-based IoT integration where every component acts as a part of a trustless peer-to-peer network and contributes to this network as much as its capabilities. This way, data collection and storage may be standardized by using blockchain client protocols. In a sample integration scenario gateways operate as full nodes and various end devices connect to it using different blockchain protocols. The next chapter provides how to proof this work by different methods.

Chapter 4

Proof of Concept

4.1 Introduction

This Chapter provide a details description of how to proof our work that Blockchain can provide privacy as well as security of Internet of Things (IoT). In this work several types of Ethereum transaction can process. We proposed a sequence diagram of this transaction between devices and users. Miner process all the incoming and outgoing transactions to a form a network and can store this information to local or cloud storage.

4.2 Procedure

In previous implementation, from the terminal devices data was sent to an always-on gateway, which then routed this data stream through the official Ethereum client to a private Ethereum network using a smart contract [54]. Without creating a connection to any specific group for the transaction, gateway send their data as block. After data has been received, this data is pushed into the Ethereum network for transaction. Then a file hash is received and these file used to access particular function to control terminal devices. A smart proxy may communicate with the Ethereum network that has been sent from the gateway. Then with the keystore/JSON file, to unlock this transaction with the Ethereum network, however a smart contract should be assigned first for the terminal devices. After being compiled keystore/JSON into bytecode, smart contracts are sent just like any other transaction of the blockchain network, to be mined by miners. When a smart contract address is created, then this smart contact address and Application Binary Interface (ABI) which is created by Etherscan network used to interact with terminal devices. A smart proxy may communicate with the Ethereum client by means of its JSON-RPC interface, however, to enable a real interaction with the Ethereum network, a smart contract should be deployed first. After being compiled into bytecode, smart contracts are sent just like any other transaction, to be mined by miners. When a smart contract is mined, its address and application binary interface (ABI) are used to interact with it.

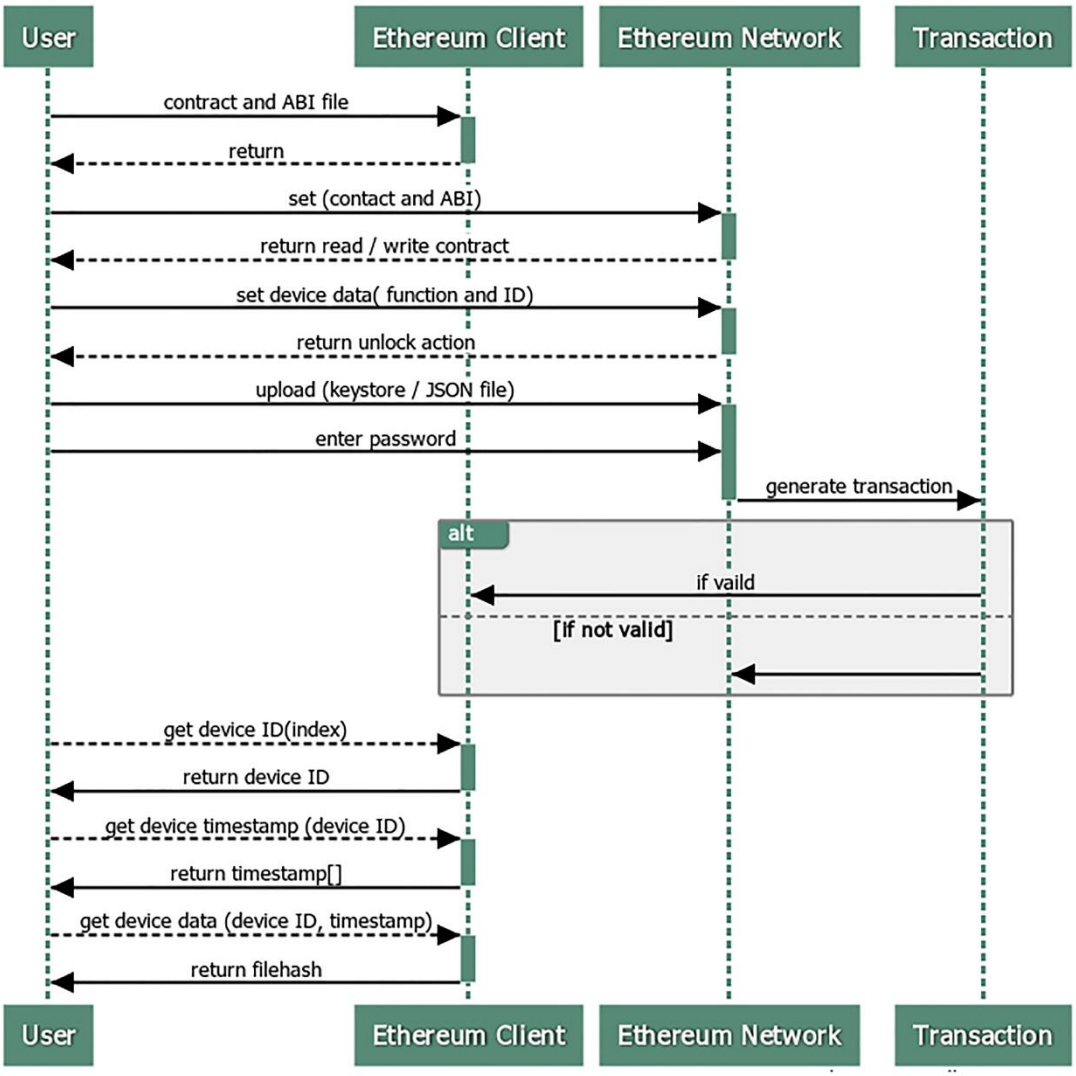

Figure: 4.1 IoT device data and access sequence diagram.

4.3 Application Binary interface

An application binary interface (ABI) is an interface between two binary program modules; often, one of these modules is a library or operating system facility, and the other is a program that is being run by a user. An ABI defines how data structures or computational routines are accessed in machine code, which is a low-level, hardware-dependent format; in contrast, an API defines this access in source code, which is a relatively high-level, hardware-independent, often human-readable format. A common aspect of an ABI is the calling convention, which determines how data is provided as

input to or read as output from computational routines; examples are the x86 calling conventions. Adhering to an ABI (which may or may not be officially standardized) is usually the job of a compiler, operating system, or library author; however, an application programmer may have to deal with an ABI directly when writing a program in a mix of programming languages, which can be achieved by using foreign function calls.

Contact ABI File:


```
[{"constant":true,"inputs":[],"name":"getBalances","outputs":[{"name
":"_addresses","type":"address[]"},{"name":"_balances","type":"uint2
56[]"}],"payable":false,"stateMutability":"view","type":"function"},
{"constant":true,"inputs":[],"name":"name","outputs":[{"name":"","ty
pe":"bytes32"}],"payable":false,"stateMutability":"view","type":"fun
ction"},{"constant":false,"inputs":[{"name":"spender","type":"addres
s"},{"name":"tokens","type":"uint256"}],"name":"approve","outputs":[
{"name":"success","type":"bool"}],"payable":false,"stateMutability":
"nonpayable","type":"function"},{"constant":true,"inputs":[],"name":
"totalSupply","outputs":[{"name":"","type":"uint256"}],"payable":fal
se,"stateMutability":"view","type":"function"},{"constant":false,"in
puts":[{"name":"from","type":"address"},{"name":"to","type":"address
"},{"name":"tokens","type":"uint256"}],"name":"transferFrom","output
s":[{"name":"success","type":"bool"}],"payable":false,"stateMutabili
ty":"nonpayable","type":"function"},{"constant":true,"inputs":[],"na
me":"decimals","outputs":[{"name":"","type":"uint8"}],"payable":fals
e,"stateMutability":"view","type":"function"},{"constant":false,"inp
uts":[{"name":"_addresses","type":"address[]"}],"name":"disableWhite
list","outputs":[{"name":"success","type":"bool"}],"payable":false,"
stateMutability":"nonpayable","type":"function"},{"constant":false,"
inputs":[{"name":"token_amount","type":"uint256"}],"name":"burnToken
s","outputs":[],"payable":false,"stateMutability":"nonpayable","type
":"function"},{"constant":true,"inputs":[{"name":"tokenOwner","type"
:"address"}],"name":"balanceOf","outputs":[{"name":"balance","type":
"uint256"}],"payable":false,"stateMutability":"view","type":"functio
n"},{"constant":true,"inputs":[],"name":"owner","outputs":[{"name":"
","type":"address"}],"payable":false,"stateMutability":"view","type"
:"function"},{"constant":true,"inputs":[],"name":"symbol","outputs":
[{"name":"","type":"bytes32"}],"payable":false,"stateMutability":"vi
ew","type":"function"},{"constant":false,"inputs":[{"name":"token_am
ount","type":"uint256"}],"name":"mintTokens","outputs":[],"payable":
false,"stateMutability":"nonpayable","type":"function"},{"constant":
false,"inputs":[],"name":"closeCrowdsale","outputs":[],"payable":fal
se,"stateMutability":"nonpayable","type":"function"},{"constant":fal
se,"inpayable":false,"stateMutability":"nonpayable","type":"function
"},{"constant":false,"inputs":[],"name":"immediateWithdrawal","outpu
ts":[],"payable":false,"stateMutability":"nonpayable","type":"functi
on"},{"constant":true,"inputs":[],"name":"price","outputs":[{"name":
"","type":"uint256"}],"payable":false,"stateMutability":"view","type
}]
```

4.4 Smart Contract Interface

Step-1: Check if transaction is processed. Then function is transaction (bool status).

Step-2: Check if device is present. Then the function get_device_ID (int index) public constant returns (int device_ID);

Step-3: Get timestamp values containing data (for a specific device) function get_device_timestamps (int device_ID) public constant returns (char [] timestamp);

Step-4: Get stored file hashes (handles) with certain timestamp (for a specific device) function get_device_data (int device_id, char [] timestamp) public constant returns (string filehash);

Step-5: Push file hashes (hashfile) into the chain function set_device_data (int device_id, string filehash) public returns (int index, char [] timestamp);

Step-6: Event to log action event log_action (int device_address, bool transaction, device_ID, int index, char [] timestamp, string filehash);

```
30 struct iot_device_data {
31   bool transaction; // check transaction
32   int index;  // selected devices ID
33   char[] timestamps;  // block chain timestamps
34
35   mapping(uint => string) filehashes;  // map time stamp and hashes file
36  }
37  device_address[] private device_index;  // all device id's
38
39  mapping(device_address => device_data) private device_loging;
40
41  loging_action (device_address, bool transaction,
42              int index,char[] timestamp,string filehash);
```

Figure: 4.2 Smart contract data structure implementation.

Figure 4.2 shows the actual code fragment exposing how terminal device identified timestamp values and file hashes. IoT gateways and their hashes file can be calculated and obtained by using get_device_data () and get_device_timestamp () functions. The actual smart contract function is set_device_data () is worked when a new data is added, then a log_action () event is fired and all peers that are related with that event.

4.5 Corresponding Coding

```
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include <ArduinoJson.h>
#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS 2

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature DS18B20( & oneWire);
int temperatureF;
int temperatureC;
int counter;
int currentTemperature;

void setup() {
  counter = 0;
  pinMode(0, OUTPUT);
  Serial.begin(115200);
  DS18B20.begin();
  WiFi.begin("blockchain", "ethereum");
  while (WiFi.status() != WL_CONNECTED) {

    delay(500);
    Serial.println("Waiting for connection");
  }
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
}

void getTemperature() {
  float tempC;
  float tempF;
  do {
    DS18B20.requestTemperatures();
    tempC = DS18B20.getTempCByIndex(0);
    temperatureC = tempC;
    tempF = DS18B20.getTempFByIndex(0);
    temperatureF = tempF;
    delay(100);
  } while (tempC == 85.0 || tempC == (-127.0));
}

String callGeth(String inputJSON) {
  HTTPClient http;

  http.begin("http://192.168.0.1:8080/");
```

```
http.addHeader("Content-Type", "application/json");

 int httpCode = http.POST(inputJSON);
 String JSONResult = http.getString();
 http.end();
 return JSONResult; }
```

4.6 Conclusion

This provide a details description of how to proof our work that Blockchain can provide privacy as well as security of Internet of Things (IoT) in all sector. In this thesis we try to show several types of Ethereum transaction which can process by this method. We proposed a sequence diagram of this transaction between devices and users that is prove that it is possible to integrated Blockchain with IoT as well as we provide a corresponding coding that also prove that it is possible for next generation to provide secure IoT environment with the help this technology. Next we will discuss about evaluation and analysis.

Chapter 5

Evaluation and Analysis

5.1 Introduction

In this chapter, the main focus is to analyze the obtained result. The result are based on the data obtained which explain in chapter 4. We designed our network and analyze different performance parameters. This chapter provides a complete discussion on the security and performance of the blockchain based IoT devices by exchange of Ethereum.

5.2 Security Analysis

There are three main security requirements that need to be addressed by any security design, namely: Confidentiality, Integrity, and Availability, known as CIA [55]. Confidential information should not be accessible to unauthorized users. Integrity means data may only be modified through an authorized mechanism and authorized users should be able to access data for legitimate purposes as necessary called availability. These three types of security mechanism included our infrastructure when we transaction among IoT devices by Ethereum network. This table summarizes of this infrastructures for security management in the case of Ethereum transaction.

Table: 5.1 Security evaluation

Requirements	Employed safeguard
Confidentiality	Symmetric encryption.
Integrity	Hashing
Availability	Limiting acceptable transactions
User control	Logging with JSON file

5.2.1 Confidentiality

Confidentiality is roughly equivalent to privacy. Measures undertaken to ensure confidentiality are designed to prevent sensitive information from reaching the wrong people, while making sure that the right people can in fact get it: Access must be

restricted to those authorized to view the data in question. It is common, as well, for data to be categorized according to the amount and type of damage that could be done should it fall into unintended hands.

More or less stringent measures can then be implemented according to those categories. Sometimes safeguarding data confidentiality may involve special training for those privy to such documents. Such training would typically include security risks that could threaten this information. Training can help familiarize authorized people with risk factors and how to guard against them. Further aspects of training can include strong passwords and password-related best practices and information about social engineering methods, to prevent them from bending data-handling rules with good intentions and potentially disastrous results.

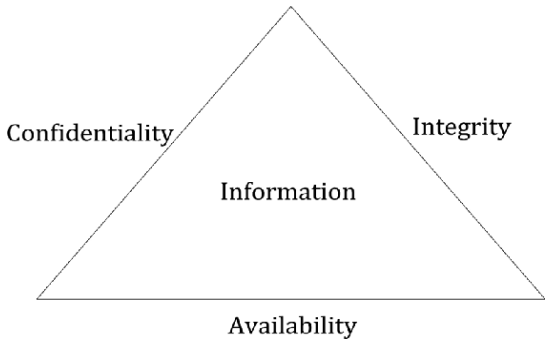

Figure 5.1 The Security Requirements Triad

5.2.2 Integrity

Integrity involves maintaining the consistency, accuracy, and trustworthiness of data over its entire life cycle. Data must not be changed in transit, and steps must be taken to ensure that data cannot be altered by unauthorized people. These measures include file permissions and user access controls. Version control maybe used to prevent erroneous changes or accidental deletion by authorized users becoming a problem. In addition, some means must be in place to detect any changes in data that might occur as a result of non-human-caused events such as an electromagnetic pulse (EMP) or server crash. Some data might include checksums, even cryptographic checksums, for verification of integrity. Backups or redundancies must be available to restore the affected data to its correct state.

5.2.3 Availability

Availability is best ensured by rigorously maintaining all hardware, performing hardware repairs immediately when needed and maintaining a correctly functioning operating system environment that is free of software conflicts. It's also important to keep current with all necessary system upgrades. Providing adequate communication bandwidth and preventing the occurrence of bottlenecks are equally important. Redundancy, failover, RAID even high-availability clusters can mitigate serious consequences when hardware issues do occur. Fast and adaptive disaster recovery is essential for the worst case scenarios; that capacity is reliant on the existence of a comprehensive disaster recovery plan (DRP). Safeguards against data loss or interruptions in connections must include unpredictable events such as natural disasters and fire. To prevent data loss from such occurrences, a backup copy may be stored in a geographically-isolated location, perhaps even in a fireproof, waterproof safe. Extra security equipment or software such as firewalls and proxy servers can guard against downtime and unreachable data due to malicious actions such as denial-of-service (DoS) attacks and network intrusions.

5.2.4 Internet of Things Privacy

Internet of Things privacy is the special considerations required to protect the information of individuals from exposure in the IoT environment, in which almost any physical or logical entity or object can be given a unique identifier and the ability to communicate autonomously over the Internet or a similar network. The data transmitted by a given endpoint might not cause any privacy issues on its own. However, when even fragmented data from multiple endpoints is gathered, collated and analyzed, it can yield sensitive information.

5.2.5 Internet of Things Security

Internet of Things security is also a special challenge because the IoT consists of so many Internet-enabled devices other than computers, which often go unpatched and are often configured with default or weak passwords. Unless adequately protected, IoT things could be used as separate attack vectors or part of a thingbot. In a recent proof-of-concept exploit, for example, researchers demonstrated that a network could be

compromised through a Wi-Fi-enabled light bulb. In December 2013, a researcher at Proof point, an enterprise security firm, discovered that hundreds of thousands of spam emails were being logged through a security gateway. Proof point traced the attacks to a botnet made up of 100,000 hacked appliances. As more and more products are developed with the capacity to be networked, it's important to routinely consider security in product development.

5.2.6 Blockchain Security for IoT

One of the most important emerging trends is the amalgamation of blockchain technology and the Internet of Things. The decentralization of an IoT network will provide it with the ability to solve a lot of its security challenges. Capabilities of the technology, including trustworthiness, decentralization, scalability, and autonomy, make it a potentially essential component of the overall IoT ecosystem In the context of the Internet of Things, blockchain technology can be applied to ensure the successful processing of multiple transactions, the tracking, and coordination of millions of smart devices, etc. Essentially, blockchain technology investment by the IoT industry can ensure the proper management of data at various levels. Also, since blockchain technology is based on cryptography, its integration into IoT network s can provide additional privacy and security.

Also, blockchain technology gets transactions recorded orderly and carefully. This means that the history of connected devices can be recorded. Add this to the fact that blockchain technology works without the necessity of central authority and you will see that integration possibilities and benefits are truly endless.

5.3 Resources Analysis

We configure two different types of Ethereum network connection in our infrastructure which we used Wi-Fi and LoRaWAN as IoT protocol. The two connections among Ethereum network and IoT devices are one-month-old, private Ethereum network (around 100000 blocks) and a public Ethereum network (around 200000 blocks). In this two configurations, peak memory consumption may vary from setup to setup due to synchronization of speed among Ethereum network and IoT devices such as mining full-node, non-mining full-node, non-mining light- node and

mining archive-node. Generally, every node is called a client which is a device that communicates with the Ethereum network. Table 2 shows the statistics of resources of different nodes for a private Ethereum network and for the shareable or public Ethereum network.

Table: 5.2 Resource consumption due to synchronization speed in different nodes.

Properties of different nodes	Mining full node	Non-mining full node	Non-mining light node	Mining archive node
Distributed storage platform user	Active	Active	Active	Active
Verify block transaction	Yes	Yes	No	Yes
Memory use in private Ethereum network	1 GB-1.5GB	0.5GB-0.8GB	Around 0.3 GB	1.2 GB-1.7GB
Memory use in public Ethereum network	At least 4GB	At least 2GB	Around 0.2GB	At least 5GB

The statistics are given below for a private Ethereum network and for the shareable or public Ethereum network. Generally, mining full node is used in powerful servers deployed in the cloud, non-mining full node is used in IoT Vendor Server, Strong IoT Gateway, etc. Non- mining light node is used in IoT gateways and end devices whereas mining archive node is a special case of a full node.

5.3.1 Mining Full Node

With an active Swarm client, these nodes use between 1.2GB and 1.5GB of memory in private Ethereum network. In public network they need at least 4GB of memory; it is

expected that this requirement will go up in time. These nodes are powerful servers deployed in the cloud.

5.3.2 Non-Mining Full Node

With an active Swarm client, these nodes use between 300MB and 400MB of memory in private Ethereum network. In public network they need at least 2GB of memory to properly sync with blockchain. These nodes may be IoT vendor servers, network provider servers or powerful IoT gateways.

5.3.3 Non-Mining Light Node

With an active Swarm client, these nodes use around 300MB of memory, whereas Ethereum client only uses around 50MB (200MB in public Ethereum). Because Swarm has no light client mode to limit bandwidth or memory usage at the moment, the memory benefit is minimal. These nodes may be regular IoT gateways and end devices.

5.4 Data Analysis

Bitcoin offer output limits with its 10-minutes average with the fixed block size of 1MB [52]. But in Ethereum transaction, there is no fixed block size but included an amount of resource to be used by transactions limit for each block called gas limit. Similar to resource analysis, data analysis statistics are given for a private Ethereum blockchain as well as for the shareable Ethereum blockchain.

Table: 5.3 Data analysis in different network.

Properties	Private transaction	Shareable transaction
Default mining strategy of a minimum gas limit	4712388	6718904
Gas price	384.284 Ethereum	384.284 Ethereum
Transaction per block	224 transaction	320 transaction
Average block time	14 sec	30 sec
Output	57600 transaction/hours	38520 transaction/hours

Although this output indicates fewer directions to support full-scale expansion today, but it is necessary to remark that all transactions are created only by IoT gateway and every IoT gateway may serve thousands of terminal devices. But using this capability of Ethereum transaction, it is possible to maintain keep secure billion of IoT devices.

Data throughput in blockchain systems depends on various metrics and varies in different implementations. Our private Ethereum network has a gas limit of 4,712,388 gas/block and the average gas price is 21k gas, therefore, a block may only contain 224 transactions. Considering that the average block time is 14 seconds for the private system, the throughput will be 16 transactions per second (or around 1k transactions per minute).

The public Ethereum is in the middle of a difficulty increase as of September 2017. At the time of writing, the public Ethereum system has a gas limit of 6,718,904 gas/block, an average gas price of 21k gas and an average block time of 30 seconds. Data throughput will in turn be 320 transactions per block, which is 10.6 transactions per second (or 640 transactions per minute).

Though transaction throughput seems low to support a full-scale deployment today, it is imperative to note that transactions are created only by IoT gateways and every gateway may serve hundreds of thousands of end devices. Proposed infrastructure can support tens of thousands of IoT gateways (and millions of end devices) pushing data periodically every 15 minutes.

5.5 Conclusion

Here we mainly focus on to analyze the obtained result by integration of Blockchain with IoT by Ethereum transaction. We try to designed our network and analyze different performance parameters from different measurements. The next chapter provides a complete discussion of which parts of the IoT and blockchain technology can be improved as well as the challenge of how to tackle upcoming various future automation and data exchange manufacturing technology.

Chapter 6

Discussion and Conclusion

6.1 Introduction

For better perception, this section discusses which parts of the IoT and blockchain technology can be improved as well as the challenge of how to tackle upcoming various future automation and data exchange manufacturing technology.

6.2 Incapacity

Ethereum uses proof of work and consensus algorithms that promise that every block is backed by a certain amount of computational work. This way is Inharmonious inefficient because every miner in the blockchain network is doing hard calculations. Integration of IoT with blockchain technology most of time facing, memory limitation and may create monopolies due to the centralization of stake, but in the case of future automation and data exchange manufacturing technologies like the Internet of things, cyber-physical systems, cloud computing, cognitive computing in the era of the fourth industrial revolution this “bug” may be used as a “feature”. Authorization of certain trusted parties like system integrators or regulators may indeed be beneficial.

6.3 Communication link

Blockchain-based systems store transactions that are signed with public-key cryptography. Although gateways are the main point of transmission to the cloud for connected terminal devices. Consider that all gateways connected to a fast communication link otherwise IoT and blockchain integration would be delay although we proposed a less time-consuming technique. When an IoT system is dealing with sensitive data must be encrypted before pushed into the blockchain for accessing terminal device data which is most beneficial for the revolution of industry 4.0 when we exchange data that create a virtual copy of all transaction and make decentralized decisions.

6.4 Real-Time application

Because of their trustless nature, blockchain-based systems may be able to store data only after a certain period of time. In order to support real-time IoT applications, data propagation delay, memory consumption, and computational complexity should be minimized by proposing new types of IoT infrastructures where application development and data processing can be massively conducted by using smart contracts in future heterogynous technologies.

6.5 Future Work

We've reached The Fourth Industrial Revolution, and along the way, it brought the concept of smart cities and homes. Our main target is to integrate technologies in a way that it would perform most of our daily tasks securely which includes cyber-physical systems, cloud computing, cognitive computing, etc.

6.6 Conclusion

In order to deal with the increasing number of IoT devices, it is essential to standardize the method of communication among them. Combination of decentralized, trustless nature, fault-tolerant data storage and DDOS-resistant of blockchain technology a new type of IoT infrastructures may be created. In this way, increasing number of IoT devices may be compact into this novel infrastructure according to their functionality, computational complexity, computing technique, and memory capacities. This achievement will lead to various models where application development and data processing can be guided using this technology. These technologies are highly committed for the future fourth industrial revolution where automation and data exchange among various manufacturing technologies is the main challenge.

Bibliography

[1] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, “Blockchain for IoT security and privacy: The case study of a smart home,” 2017 IEEE Int. Conf. Pervasive Comput. Commun. Work. PerCom Work. 2017, no. March, pp. 618–623, (2017).

[2] S. Ali, I. Quadri, and C. Engineering, “IoT Based Smart Home Automation and Surveillance System,” vol. 4, no. 3, pp. 861–866, (2017).

[3] G. S. Ramachandran and B. Krishnamachari, “The Blockchain technology for the IoT: Opportunities and Challenges,” no. May (2018).

[4] A. Toffler, “Future Shock,” no. October, pp. 98–100, 1990. http://www.forbes.com/sites/josephsteinberg/2014/01/27/thesedevices-may-bespying-on you-even-in-your-own-home (2014).

[5] J. H. Lee, “BIDaaS: Blockchain Based ID as a Service,” IEEE Access, vol. 6, no. XX, pp. 2274–2278, (2017).

[6] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and C. Yang, “Blockchain as a Decentralized Security Framework,” IEEE Consum. Electron. Mag., vol. 7, no. 2, pp. 18–21, 2018.Cook, D., Das, S. (2004). Smart environments: Technology, protocols and applications (Vol. 43). John Wiley, Sons (2004).

[7] A. Dorri, S. S. Kanhere, and R. Jurdak, “The Blockchain in internet of things: Challenges and Solutions,” (2016).

[8] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” Proc. - 2017 IEEE 6th Int. Congr. Big Data, BigData Congr. 2017, no. October, pp. 557–564, (2017).

[9] S. Session and M. C. Conference, “IOT2 . 0 : Internet of Things Based on Blockchain,” pp. 2–3, (2018).

[10] S. Nakamoto.Bitcoin: A peer-to-peer electronic cash system. [Online]. Available: https://bitcoin.org/ bitcoin.pdf (2008).

[11] E. F. Jesus, V. R. L. Chicarino, C. V. N. De Albuquerque, and A. A. D. A. Rocha, "A Survey of How to Use Blockchain to Secure Internet of Things and the Stalker Attack," Secur. Commun. Networks, vol. 2018, (2018).

[12] Foil Arms and Hog, "WTF is Brexit? - Foil Arms and Hog - YouTube," YouTube, pp. 1–9, 2016. Granzer, W., Kastner, W., Neugschwandtner, G., Praus, F. (2006). Security in networked building automation systems. (2016).

[13] O. Eibhlín, "Detecting Patterns in the Ethereum Transactional Data using Unsupervised Learning," no. August, 2018, (2018).

[14] Oliver Dale on November 2, 2017, "How to Make a Paper Ethereum Wallet" Avaliable: https://blockonomi.com/paper-ethereum-wallet, (2017).

[15] B. V. Buterin, "OffsetMapping," no. January, pp. 1–36, 2009. Schiefer, M. (2015, May). Smart Home Definition and Security Threats. In IT Security Incident Management and IT Forensics (IMF), 2015 Ninth International Conference on (pp. 114 118). IEEE, (2009).

[16] E. Hildenbrandt, M. Saxena, X. Zhu, N. Rodrigues, P. Daian, D. Guth, and G. Rosu, "Kevm: A complete semantics of the ethereum virtual machine," pp. 1–33, 2017, (2017).

[17] R. Tonelli, G. Destefanis, M. Marchesi, and M. Ortu, "Smart Contracts Software Metrics: a First Study," no. February, 2018, (2018).

[18] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, and V. Santamaría, "Blockchain and smart contracts for insurance: Is the technology mature enough?," Futur. Internet, vol. 10, no. 2, pp. 8–13, (2018).

[19] N. Komninos, E. Philippou, and A. Pitsillides, "Survey in Smart Grid and Smart Home Security: Issues, Challenges and Countermeasures," Commun. Surv. Tutorials, IEEE, vol. PP, no. 99, p. 1, (2014).

[20] N. Wadhwa, S. Z. Hussain, and S. A. M. Rizvi, “A Combined Method for Confidentiality , Integrity , Availability and Authentication (CMCIAA),” World Congr. Eng., vol. II, pp. 6–9, 2013, (2013)

[21] K. R. Özyılmaz and A. Yurdakul, “Designing a blockchain-based IoT infrastructure with Ethereum, Swarm and LoRa,” no. September, 2018, (2018).

[22] Tong, J., Sun, W., Wang, L. (2013, May). An information flow security model for home area network of smart grid. In Cyber Technology in Automation, Control and Intelligent Systems (CYBER), 2013 IEEE 3rd Annual International Conference on (pp. 456-461). IEEE, (2013).

[23] McCune, J. M., Perrig, A., Reiter, M. K. (2005, May). Seeing-is-believing: Using camera phones for human-verifiable authentication. In Security and privacy, 2005 IEEE symposium on (pp. 110 124). IEEE, (2005).

[24] Zuo, F., De With, P. H. (2005). Real-time embedded face recognition for smart home. Consumer Electronics, IEEE Transactions on, 51(1), 183-190, (2005).

[25] Yoo, D. Y., Shin, J. W., Choi, J. Y. (2007, December). Home-Network Security Model in Ubiquitous Environment. In Proceedings of World Academy of Science, Engineering and Technology (Vol. 26), (2007).

[26] Ricquebourg, V., Menga, D., Durand, D., Marhic, B., Delahoche, L., Loge, C. (2006, December). The smart home concept: our immediate future. In ELearning in Industrial Electronics, 2006 1ST IEEE International Conference on (pp. 23-28). IEEE, (2006).

[27] Dorri, S. S. Kanhere, and R. Jurdak, Blockchain in internet of things: Challenges and solutions, arXiv preprint arXiv:1608.05187, (2016).

[28] G. Foroglou and A.-L. Tsilidou, “Further applications of the blockchain,” 2015. And A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, “Hawk: The blockchain model of cryptography and privacy-preserving smart contracts,” in Proceedings of IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2016, pp. 839–858, (2015)

[29] B. W. Akins, J. L. Chapman, and J. M. Gordon, “A whole new world: Income tax considerations of the bitcoin economy,” 2013. [Online]. Available: https://ssrn.com/abstract=2394738, (2013).

[30] Y. Zhang and J. Wen, “An iot electric business model based on the protocol of bitcoin,” in Proceedings of 18th International Conference on Intelligence in Next Generation Networks (ICIN), Paris, France, 2015, pp. 184–191, (2015)

[31] M. Sharples and J. Domingue, “The blockchain and kudos: A distributed system for educational record, reputation and reward,” in Proceedings of 11th European Conference on Technology Enhanced Learning (EC-TEL 2015), Lyon, France, 2015, pp. 490–496, (2015).

[32] C. Noyes, “Bitav: Fast anti-malware by distributed blockchain consensus and feedforward scanning,” arXiv preprint arXiv:1601.01405, (2016).

[33] I. Eyal and E. G. Sirer, “Majority is not enough: Bitcoin mining is vulnerable,” in Proceedings of International Conference on Financial Cryptography and Data Security, Berlin, Heidelberg, 2014, pp. 436– 454, (2014).

[34] A. Biryukov, D. Khovratovich, and I. Pustogarov, “Deanonymisation of clients in bitcoin p2p network,” in Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 2014, pp. 15–29, (2014).

[35] F. Tschorsch and B. Scheuermann, “Bitcoin and beyond: A technical survey on decentralized digital currencies,” IEEE Communications Surveys Tutorials, vol. 18, no. 3, pp. 2084–2123, (2016).

[36] NRI, “Survey on blockchain technologies and related services,” Tech. Rep., 2015. [Online]. Available: http://www.meti.go.jp/english/press/ 2016/pdf/0531 01f.pdf (2015).

[37] D. Lee Kuo Chuen, Ed., Handbook of Digital Currency, 1st ed. Elsevier, 2015. [Online]. Available: http://EconPapers.repec.org/RePEcTRDcxTR: eee:monogr: 9780128021170 [15] V. Buterin, “A next-generation smart contract and decentralized application platform,” white paper, (2014).

[38] D. Johnson, A. Menezes, and S. Vanstone, “The elliptic curve digital signature algorithm (ecdsa),” International Journal of Information Security, vol. 1, no. 1, pp. 36–63, (2001).

[39] V. Buterin, “On public and private blockchains,” 2015. [Online]. Available: https://blog.ethereum.org/2015/08/07/ on-public-and-private-blockchains, (2015).

[40] “Hyperledger project,” 2015. [Online]. Available: https://www. hyperledger.org, (2015).

[41] “Consortium chain development.” [Online]. Available: https://github. com/ethereum/wiki/wiki/Consortium-Chain-Development, (2018).

[42] L. Lamport, R. Shostak, and M. Pease, “The byzantine generals problem,” ACM Transactions on Programming Languages and Systems (TOPLAS), vol. 4, no. 3, pp. 382–401, (2008).

[43] S. King and S. Nadal, “Ppcoin: Peer-to-peer crypto-currency with proofof- stake,” Self-Published Paper, August, vol. 19, (2012).

[44] “Bitshares - your share in the decentralized exchange.” [Online]. Available: https://bitshares.org/, (2014).

[45] D. Schwartz, N. Youngs, and A. Britto, “The ripple protocol consensus algorithm,” Ripple Labs Inc White Paper, vol. 5, (2014).

[46] J. Kwon, “Tendermint: Consensus without mining,” URL http://tendermint. com/docs/tendermint { } v04. pdf, (2014).

[47] S. King, “Primecoin: Cryptocurrency with prime number proof-ofwork,” July 7th, (2013).

[48] P. Vasin, “Blackcoins proof-of-stake protocol v2,” 2014. [Online]. Available: https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf, (2014).

[49] G. Wood, “Ethereum: A secure decentralised generalised transaction ledger,” Ethereum Project Yellow Paper, (2014).

[50] V. Zamfir, “Introducing casper the friendly ghost,” Ethereum Blog URL: https://blog. ethereum. org/2015/08/01/introducing-casperfriendly- ghost, (2015).

[51] C. Miguel and L. Barbara, “Practical byzantine fault tolerance,” in Proceedings of the Third Symposium on Operating Systems Design and Implementation, vol. 99, New Orleans, USA, 2005, pp. 173–186,(2005).

[52] “State of blockchain q1 2016: Blockchain funding overtakes bitcoin,” 2016. [Online]. Available: http://www.coindesk.com/ state-of-blockchain-q1-2016, (2016).

[53] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, “Zerocash: Decentralized anonymous payments from bitcoin,” in Proceedings of 2014 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2014, pp. 459–474,(2014)

[54] K. Nayak, S. Kumar, A. Miller, and E. Shi, “Stubborn mining: Generalizing selfish mining and combining with an eclipse attack,” in Proceedings of 2016 IEEE European Symposium on Security and Privacy (EuroS&P), Saarbrucken, Germany, 2016, pp. 305–320, (2016)

[55] Wadhwa, N., Hussain, S. Z. & Rizvi, S. A. M. A Combined Method for Confidentiality , Integrity , Availability and Authentication (CMCIAA). World Congr. on Eng. II, 6–9 (2013).