A

Seminar Report

Securing IOT devices using Blockchain

Submitted in partial fulfillment of the requirements for the Award of the Degree

of

Master of Computer Applications

of

APJ Abdul Kalam Technological University

Submitted by

RAFSAL RAHIM

RegNo: TVE16MCA41

Department of Computer Applications

COLLEGE OF ENGINEERING TRIVANDRUM

AUGUST 2018

DEPARTMENT OF COMPUTER APPLICATIONS

COLLEGE OF ENGINEERING TRIVANDRUM

CERTIFICATE

Certified that this Seminar report entitled, ***"Securing IOT devices using Blockchain "*** *is the paper presented by* ***" Rafsal Rahim "(Reg No: TVE16MCA14)*** *in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications of APJ Abdul Kalam Technological University during the year 2018.*

Prof. Baby Syla L.

Co-ordinator

Prof. Jose T Joseph.

Head of the Department

Acknowledgement

First and for most I thank **GOD** almighty and to my parents for the success of this seminar. I owe a sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my seminar.

I would like to thank **Dr.Hari V.S**, Principal, College of Engineering Trivandrum, who helped me during the entire process of work.

I am extremely grateful to **Prof.Jose T Joseph**, HOD, Dept of Computer Applications, for providing me with best facilities and atmosphere for the creative work guidance and encouragement.

I would like to thank my coordinator, **Prof. Baby Syla**, Dept of Computer Applications, who motivated me throughout the work of my seminar.

I profusely thank other Asst. Professors in the department and all other staffs of CET, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this seminar. No words can express my humble gratitude to my beloved parents and relatives who have been guiding me in all walks of my journey.

Rafsal Rahim

Abstract

Internet of Things (IoT) plays an important role in the development of various fields. The increasing scale and scope of applications make a great demand of IoT data exchange in recent years. Meanwhile, a number of IoT data exchange platforms which dedicated to connecting various and distributed data sources are emerging. In such a platform, service providers can search and exchange the data sets that they need. However, the centralized infrastructure cannot provide enough trust as the third-party intermediaries for data exchange. As a result, most platforms unable to satisfy the complex requirements due to few institutions and individuals are willing to share their IoT data sets in such an untrustworthy environment. This paper proposes a decentralized solution based on the blockchain for IoT data trusted exchange. Specifically, In this paper, the basic principles of blockchain and corresponding key technologies are expounded through in-depth analysis of three main reliable requirements in IoT data exchange. Besides, this paper provides an architecture of above solution and detailed design of its main trust component. Finally, it realizes a prototype by using Ethereum blockchain and smart contracts and presents its auditable, transparent, decentralized features visually.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the past decade, benefiting from the rapid development of wireless communication technology, sensing technology and the improvement of big data analysis capacity [1], the Internet of Things (IoT) is growing with incredible speed in most areas, especially in healthcare [2], smart city [3] and autonomous vehicles [4]. As the essential elements of the IoT world, data collected from various devices can be applied in a wide range of areas after being analyzed and processed. Combining with advanced technologies such as big data and artificial intelligence [5], IoT service based on data not only reduces the cost of industry and agriculture and makes the devices around people more intelligent, but also repeatedly optimizes the IoT ecosystem (security, equipment management and the standardization [6]) itself. However, due to the limited high maintenance and management cost [7], restricted collecting scope in privacy data, data exchange between individual and organizations become irreversible trend when they realize that connection is more important than possession [8]. Therefore, a lot of data exchange or sharing platform hava emerged in the past years. Such as crawdad for archiving wireless data [9], data science central provided industrys online resources for big data practitioners, data.gov as the home of the US governments open data platform, the development of digital coast met the unique needs of the coastal management community [10,11]. Most of them concentrated on data of the same kind or a specific field and led by government or the unions of large institutions. However, the data sets on such centralized platform cannot meet the publics diversified demands, largely due to they cannot provide enough trust to guarantee the transparency, auditable, immutable in data exchange process.

It has been watched by researchers for a long time, as the issue of trust kills the enthusiasm to share data actively and seriously hampers the development of the data industry [12]. Although current platforms integrate a lot of confidentiality mechanisms (access control, authorization privacy) and propose some trust model for data sharing in IoT, they are not break away from the third party [13].

In order to guarantee IoT data exchange in a completely trusted, transparent environment, we propose a decentralized solution based-on blockchain. As the core technology inBitcoin, blockchain soon came to be widely attracted attention and application for its trust property[14].

The core advantage of blockchain is decentralization. It consists of data encryption, timestamp, distributed consensus algorithm, economic incentive mechanism and other technology. It is applicable to the point-to-point transaction based on decentralized credit in distributed systems without mutual trust. Sequentially, this technology solves the prevailing high cost, low efficiency and uneasiness of data storage in current central institutions. This paper utilizes the feature of data is not tampered and completely transparent, combines with the time stamp and the transaction details in the process of storage and trading, so that it can be trusted by many parties. Its mixed encryption technology based on asymmetric encryption makes the user's privacy information secure with the public and private key as the only identifier of transaction subject. The second generation blockchain introduces intelligent contract, which makes the blockchain easier to use distributed application programming and speed up transaction speed. The Ethereum intelligent contract combined with capability-based access control method makes the data provider can completely control their own data sharing permissions quickly and efficiently and completely solve the credible issues of original system in the data.

Chapter 2

Little about the core technologies

2.1 Internet of Things(IOT)

2.1.1 what do IoT really Means?

Definition - What does Internet of Things (IoT) mean? The internet of things (IoT) is a computing concept that describes the idea of everyday physical objects being connected to the internet and being able to identify themselves to other devices. The term is closely identified with RFID as the method of communication, although it also may include other sensor technologies, wireless technologies or QR codes.

The IoT is significant because an object that can represent itself digitally becomes something greater than the object by itself. No longer does the object relate just to its user, but it is now connected to surrounding objects and database data. When many objects act in unison, they are known as having "ambient intelligence."

The internet of things is a difficult concept to define precisely. In fact, there are many different groups that have defined the term, although its initial use has been attributed to Kevin Ashton, an expert on digital innovation. Each definition shares the idea that the first version of the internet was about data created by people, while the next version is about data created by things. In 1999, Ashton said it best in this quote from an article in the RFID Journal:
"If we had computers that knew everything there was to know about things using data they gathered without any help from us we would be able to track and count everything, and greatly

reduce waste, loss and cost. We would know when things needed replacing, repairing or recalling, and whether they were fresh or past their best."

Most people think about being connected in terms of computers, tablets and smartphones. IoT describes a world where just about anything can be connected and communicate in an intelligent fashion. In other words, with the internet of things, the physical world is becoming one big information system.

2.1.2 How it works?

An IoT ecosystem consists of web-enabled smart devices that use embedded processors, sensors and communication hardware to collect, send and act on data they acquire from their environments. IoT devices share the sensor data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analyzed or analyzed locally. Sometimes, these devices communicate with other related devices and act on the information they get from one another. The devices do most of the work without human intervention, although people can interact with the devices – for instance, to set them up, give them instructions or access the data.

The connectivity, networking and communication protocols used with these web-enabled devices largely depend on the specific IoT applications deployed.

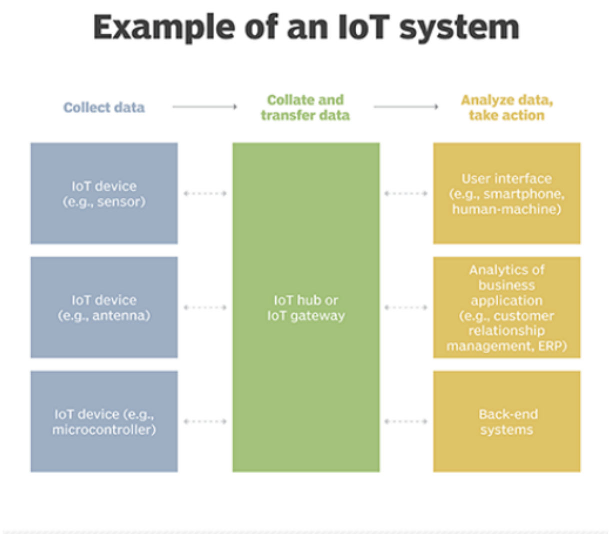

Figure 2.1: IoT System

2.1.3 Consumer and enterprise IoT applications

There are numerous real-world applications of the internet of things, ranging from consumer IoT and enterprise IoT to manufacturing and industrial IoT (IIoT). IoT applications span numerous verticals, including automotive, telco, energy and more.

In the consumer segment, for example, smart homes that are equipped with smart thermostats, smart appliances and connected heating, lighting and electronic devices can be controlled remotely via computers, smartphones or other mobile devices.

Wearable devices with sensors and software can collect and analyze user data, sending messages to other technologies about the users with the aim of making users' lives easier and more comfortable. Wearable devices are also used for public safety – for example, improving first responders' response times during emergencies by providing optimized routes to a location or by tracking construction workers' or firefighters' vital signs at life-threatening sites.

In healthcare, IoT offers many benefits, including the ability to monitor patients more closely to use the data that's generated and analyze it. Hospitals often use IoT systems to complete tasks such as inventory management, for both pharmaceuticals and medical instruments.

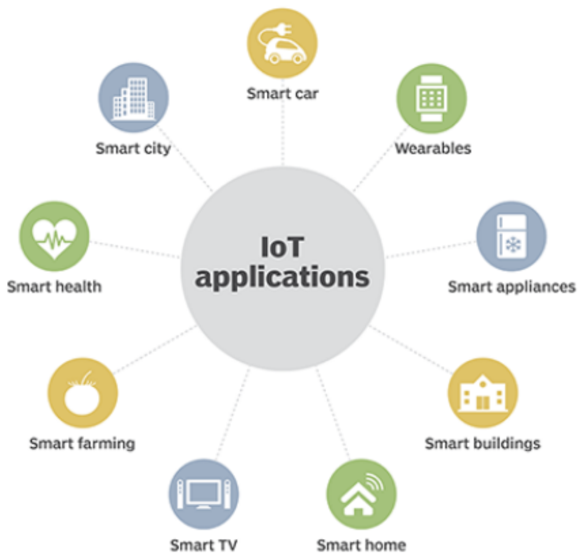

Figure 2.2: IoT Applications

Smart buildings can, for instance, reduce energy costs using sensors that detect how many occupants are in a room. The temperature can adjust automatically – for example, turning the air conditioner on if sensors detect a conference room is full or turning the heat down if everyone in the office has gone home.

In agriculture, IoT-based smart farming systems can help monitor, for instance, light, temperature, humidity and soil moisture of crop fields using connected sensors. IoT is also instrumental in automating irrigation systems.

In a smart city, IoT sensors and deployments, such as smart streetlights and smart meters, can help alleviate traffic, conserve energy, monitor and address environmental concerns, and improve sanitation.

Components of IoT:

1. **Sensors/Devices:**
   First, sensors or devices collect data from their environment. This could be as simple as a temperature reading or as complex as a full video feed.

   I use “sensors/devices, because multiple sensors can be bundled together or sensors can be part of a device that does more than just sense things. For example, your phone is a device that has multiple sensors (camera, accelerometer, GPS, etc), but your phone is not just a sensor.

   However, whether it‘s a standalone sensor or a full device, in this first step data is being collected from the environment by something.

2. **Connectivity:**
   Next, that data is sent to the cloud , but it needs a way to get there!

   The sensors/devices can be connected to the cloud through a variety of methods including: cellular, satellite, WiFi, Bluetooth, low-power wide-area networks (LPWAN), or connecting directly to the internet via ethernet.

Each option has tradeoffs between power consumption, range and bandwidth (heres a simple explanation). Choosing which connectivity option is best comes down to the specific IoT application, but they all accomplish the same task: getting data to the cloud.

3. **Data Processing:**
   Once the data gets to the cloud, software performs some kind of processing on it.

   This could be very simple, such as checking that the temperature reading is within an acceptable range. Or it could also be very complex, such as using computer vision on video to identify objects (such as intruders in your house).

   But what happens when the temperature is too high or if there is an intruder in your house? Thats where the user comes in.

4. **User Interface:**
   Next, the information is made useful to the end-user in some way. This could be via an alert to the user (email, text, notification, etc). For example, a text alert when the temperature is too high in the companys cold storage.

   Also, a user might have an interface that allows them to proactively check in on the system. For example, a user might want to check the video feeds in their house via a phone app or a web browser.

   However, its not always a one-way street. Depending on the IoT application, the user may also be able to perform an action and affect the system. For example, the user might remotely adjust the temperature in the cold storage via an app on their phone.

   And some actions are performed automatically. Rather than waiting for you to adjust the temperature, the system could do it automatically via predefined rules. And rather than just call you to alert you of an intruder, the IoT system could also automatically notify relevant authorities.

2.2 Security Issue and vulnerabilities of IoT

One of the security weakness of IoT is that it increases the number of devices behind networks firewall. As Based on the review in, ten years ago, there was a major concern about protecting computers, five years ago, the concern was about protecting smartphones. Now we have to worry about protecting our car, our home appliances, our wearables, and many other IoT devices. Computers also have security problems but with automatic and easier updates have helped alleviate this problem. But in case of IoT devices, manufacturers are pressured to get their devices in the market , there by ending up on compromising the security. Even if they may offer firmware upgrades for a time, they often stop when they focus on constructing the next device, leaving customers with slightly outdated hardware that can become a security risk.

IoT devices have security concerns, as these devices can easily get attacked by hackers, the data will be hacked and these devices may get controlled or accessed by the hackers. The point is that we have to think about what a hacker could do with a device if he can break through its security. A strong cryptographic algorithms are required to secure IoT devices and to provide secure channel. The following lists different kinds of attacks which have been observed recently and discussed in.

2.2.1 DDoS attacks

In 2016, the Mirai botnet launched one of the biggest DDoS attacks ever recorded. More than 1 terabyte per second flooded the network of Dyn, a major DNS provider, and brought down sites such as Reddit and Airnbnb. But what made this attack so special was that it was the first to be carried out with IoT devices. Nearly 150,000 compromised smart cameras, routers and other devices all enslaved into a single botnet, focused on a single target.

Manufacturers usually use a handful of default password and usernames to protect an IoT device. So there will be a few hundreds/thousands of password combinations to protect tens of millions of smart devices. All it took were a few simple lines of code, designed to test each of those default passwords. A device could be hacked and enslaved within a few seconds, so long as the user didnt change the standard login information.

2.2.2 Unsecure car apps

As Internet connected cars are coming around, it has been observed that hackers are trying to take control of the onboard software , trying to access and open the car locks. Unsecure car apps can allow malicious hackers to control ones car.

2.2.3 Insecure default login credentials

In practice, manufacturers might hide the Change password/Username options deep in the UI, out of sight for most users. If each Internet of Things device had a randomized username and password, Mirai might not have happened in the first place.

2.2.4 Poor software updates

Many Internet of Things manufacturers dont even patch or update the software that came on their devices. If a device has software vulnerability, theres little one can done to prevent an attacker from exploiting it without help from the manufacturer.

2.2.5 The communication isnt encrypted

Many IoT devices lack basic encryption to hide the data sent between the device and the central server. This can potentially expose the users personal information, if a malicious hacker can snoop in on his personal information.

Another thing that Internet of Things devices do, is that some of them ask for more permissions than they need to.One time, numerous Amazon Echo users were surprised to see their device ordering dollhouses after a TV anchor said the phrase Alexa ordered me a dollhouse.In that case, the device had permission to do a purchase all by itself. Each extra permission in an IoT device adds another vulnerability layer which can be exploited. The fewer permissions, the more secure your device is.

2.2.6 Insecure user interface

A devices user interface is usually the first thing a malicious hacker will look into for any vulnerabilities. For instance, he might try to manipulate the I forgot my password, in order to reset it or at least find out your username or email.

A properly designed device should also lock out a user from attempting to login too many times. This stops dictionary and brute force attacks that target passwords, and greatly secures your device credentials. In other cases, the password might be sent from the device to the central server in plain text, meaning it isnt encrypted.

2.2.7 Poor privacy protection

Internet connected devices are data-hungry beasts, but some of them have a greater appetite than others. The less information they have on you, the better, since it limits how much a cybercriminal can learn about you if he hacks the device. As a rule, try to look into what type of data a device will store about you. Be critical of those that harvest data they dont need, such as coffee machines storing users location information.

2.3 Blockchain

2.3.1 Structure of a block

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 (generally represented as a merkle tree root hash). A block is a container data structure, which brings together transactions for inclusion in the public ledger, known as the blockchain. The block is made up of a header; containing metadata, followed by a long list of transactions. A block can be identified in two ways, either by referencing the block hash, or through referencing the block height. The block header consists of three sets of block metadata. Metadata is data that provides information about other data. Firstly, there is a reference to a previous block hash, which connects this block to the previous block, lying in the blockchain. The second set of metadata relates to the mining competition; namely the difficulty, timestamp and nonce. Lastly, the third piece of metadata is the Merkle Tree root; a data structure used to summarize all the transactions in the block in an efficient manner.

Block headers can be regarded as an example of a dynamic membership multi-party signature (DMSS). A DMSS is a digital signature formed by a set of signers which has no fixed size (Back, Corallo, Dashjr, Friedenbach, 2014).Bitcoins block headers are DMSS because their proof of work has the property that anyone can contribute without undergoing an enrolment process. Furthermore, contribution is weighted by proportional computational power rather than one threshold signature contribution per party (Back, Corallo, Dashjr, Friedenbach, 2014).

This allows anonymous membership without risk of a Sybil attack. A Sybil attack is when one party joins many times and has an uneven, disproportionate input into the signature. Since the blocks are chained together, Bitcoins DMSS is cumulative. A chain of block headers is also a DMSS on its first block, with computational strength equivalent to the sum of the computational strengths of the composing DMSS . Therefore, the key innovation in Blockchain is a signature of computational power, rather than the typical signature of knowledge.

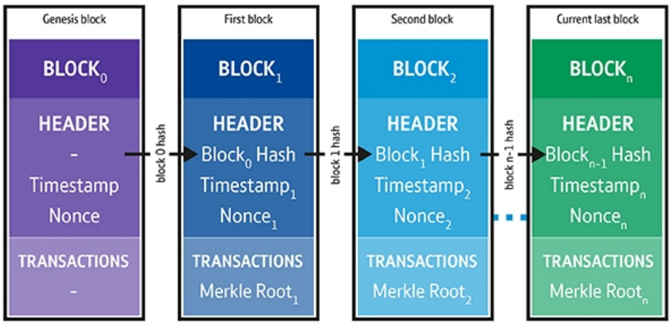

Figure 2.3: Blockchain structure

2.3.2 Block header hash and nodes

Here I have am providing an example, the block hash of the first Bitcoin block ever created will be like 000000000019d7789c085ae165831e934gf763ae46a4a6c172b3f1b60a8ce26f. The block hash identifies a block uniquely, and can be independently derived by any node simply by hashing the block header. A node is a full client. A full client is a client that owns the block chains and is sharing blocks and transactions across the blockchain network. A node is considered to be part of the blockchain infrastructure, and does not necessarily have to be a miner. Each node keeps a complete copy of a totally ordered sequence of events in the form of a blockchain . The blocks hash is computed by each node, as the block is received from the network. The block hash may be stored in a separate database table as part of the blocks metadata, to facilitate indexing and faster retrieval of blocks from disk.

2.3.3 Block height

Block height is another method to identify a block, this time through its position in the blockchain. The first block ever created is at block height 0 (zero), and in the case of Bit- coin, is the same block that was referenced by the block hash of the above block which is 000000000019d7789c085ae165831e9 .Each subsequent block added on top of that first block is one position higher in the blockchain, like boxes stacked one on top of the other.Block height does not always identify a particular singular block. It is possible for two or more blocks may have the same block height, both competing for the same position in the blockchain.

2.3.4 Genesis Block

The first block in any blockchain is termed the genesis block. If you start at any block and follow the chain backwards chronologically, you will arrive at the genesis block. The genesis block is statically encoded within the client software, that it cannot be changed. Every node can identify the genesis blocks hash and structure, the fixed time of creation, and the single transactions within. Thus every node has a secure root from which is possible to build a trusted blockchain on.

2.3.5 Proof of Work

In Proof of Work, in order for an actor to be elected as a leader and choose the next block to be added to the blockchain they have to find a solution to a particular mathematical problem. Given that the hash function used is cryptographically secure, the only way to find a solution to that problem is by bruteforce (trying all possible combinations). In other words, probabilistically speaking, the actor who will solve the aforementioned problem first the majority of the time is the one who has access to the most computing power. These actors are also called miners. It has been widely successful primarily due to its following properties:

1. It is hard to find a solution for that given problem

2. When given a solution to that problem it is easy to verify that it is correct.

Whenever a new block is mined, that miner gets rewarded with some currency (block reward, transaction fees) and thus are incentivized to keep mining. In Proof of Work, other nodes verify the validity of the block by checking that the hash of the data of the block is less than a preset number.Due to the limited supply of computational power, miners are also incentivized not to cheat. Attacking the network would cost a lot because of the high cost of hardware, energy, and potential mining profits missed.

2.3.6 Linking blocks in the blockchain

Nodes maintain a copy of the blockchain locally, starting from the genesis block. The local copy of the blockchain constantly updates as new blocks are discovered and subsequently built on the chain. As a node receives information of incoming blocks from the network, it will validate these blocks first, then link them to the existing blockchain.

The process to establish a link is as follows; a node will examine the incoming block header and look for the previous block hash. Looking at this incoming block, the node finds the previous block hash field, which contains the hash of its parent block. This hash is known to the node previously. Therefore, the node reasons that this new block is a child of the last block on the chain, and is the legitimate extension of the chain. The node adds this new block to the end of the chain, making the blockchain longer with a new height of the incoming block, now validated.

2.3.7 Cryptographic Hash Algorithm

The blockchain data structure is a back-linked list of blocks of transactions, which is ordered. It can be stored as a flat file or in a simple database. Each block is identifiable by a hash, generated using the SHA256 cryptographic hash algorithm on the header of the block. Each block references a previous block, also known as the parent block, in the previous block hash field, in the block header.A hash, also known in long form as cryptographic hash function, is a mathematical algorithm that maps data of arbitrary size to a bit string of a fixed size. In the case of SHA 256, the result is a string of 32 bytes. The resultant 32 bytes makes it effectively impossible to reverse the output, since the function was designed to be a one-way function .

The idea behind a hash functions use is to facilitate a thorough means for searching for data in a dataset. The most basic form of hash function is any function that can be used to map data of arbitrary size to data of fixed size. This output is a bit-string known as the hash value, hash sum or hash code. The hash values can be stored in a tabular form known as a hash table and is an efficient indexing mechanism; especially useful in search performance. Hash functions are collision-free too. That means its impossible to find two messages that hash to the same hash value. Therefore, when given a compact hash, one can confirm that it matches a particular input datum. Blocks can be identified from their hash, serving two purposes; identification and integrity verification.

Bitcoin hashing function makes use of the SHA 256, applied twice. It generates an almost-unique, fixed size 256-bit (32-byte) hash security. Large classes of hash functions are based on a building block of a compression function .Each block contains the hash of its parent inside its own header. There lays a chain going all the way back to the first block created, also known as the genesis block, linked together by a sequence of hashes. The previous block hash field is inside the block header and thereby the current block hash is dependent on the parent block hash. The childs own identity changes if the parents identity changes. When the parent is modified in any way, the parents hash changes. The parents changed hash necessitates an alteration in the previous block hash pointer of the child. This in turn causes the childs hash to mutate, which requires a change in the pointer of the grandchild, which in turn alters the grandchild and so on.

This cascading effect ensures that, once a block has many generations succeeding it, it cannot be changed without consequently forcing a recalculation of all the subsequent blocks. Because such a recalculation would require an enormous amount of computation, the existence of a long chain of blocks fortifies the Blockchains deep history to be immutable; a key feature of blockchain technology security.

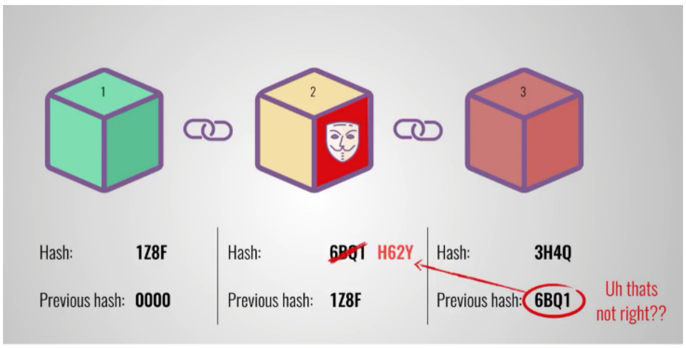

Figure 2.4: Denial on Data Modification

2.3.8 Modification of Data

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 perma- nent 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 al- teration 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.

Blockchains are incredibly popular nowadays.Here, I have a chain of three blocks. As you can see, each block has a hash and a hash of the previous block.So block 3 points to block 2 and block 2 points to the block 1. Now the block 1 is special. It is the Genesis block and it cannot point to any other block. Now if block 2 is tampered, it changes the hash of the block 2. Now the hash in the block 3 becomes invalid as it does not match with the hash in the previous block. Computers are very fast and they can calculate hash at a very high speed.

Chapter 3

Existing System Approach

It relies on centralised, brokered communication models (also known as, client-server paradigm). All devices are identified, authenticated, and connected through cloud servers that sport huge processing and storage capacities. Connection among devices happen exclusively over the internet even if theyre a few feet apart. Also, machine-to-machine (M2M) communication is difficult because there is no single platform that connects all devices no guarantee that cloud services offered by different manufacturers are interoperable and compatible.

The centralised architecture poses challenges to secure IoT deployments. Handling the enormous volume of existing and projected data is daunting. Managing the inevitable complexities of connecting to a seemingly unlimited list of devices is complicated. And the goal of turning the deluge of data into valuable actions seems impossible because of the many challenges. The existing security technologies will play a role in mitigating IoT risks but they are not enough. The goal is to get data securely to the right place, at the right time, in the right format; its easier said than done for many reasons.

The centralised security model common in the enterprise today will struggle to scale up to meet the demands of the internet of things, or IoT.

3.1 Disadvantages

As most IoT networks are currently configured, data isnt be considered trustworthy until it is vetted and allowed to pass through a single controlling security gate. As we saw with the

Mirai incidents, a botnet barrage can focus efforts on compromising this lone point through a Distributed Denial of Service (DDNS) attack. Once through the centralized security gate, a hacker can access the resources of the entire IoT network. For those familiar with blockchain technology (which well discuss in more detail in relation to the IoT shortly), the disadvantages of the present centralized security approach are obvious, especially when compared to the more secure distributed model found in blockchain technology.

Online security: Through something as ubiquitous as a coffee maker (or any of dozens of other devices) attached to a home network, a hacker could enter the system and gain access to ALL the information on ANY network connected device. That means passwords, checking or savings accounts all the worst parts of identity theft. Following that thread of thought, it would be relatively easy to introduce ransomware or tell devices to stop functioning entirely.

Surveillance: Got security cameras? Many homeowners and businesses do. One glaring weakness of a centralized server is how easy it is for hackers to get into the system and use security cameras to surveil the surroundings. What better way to have your privacy invaded or to case the place for a burglary? Even the remote camera on your vacuum cleaner could be hijacked by those with ill intent. Were getting into creepy and legitimately dangerous territory here now.

A large enterprise nightmare: Weve already mentioned Mirai. One offshoot, Persirai, has been trained to infect 1,250 different types of security cameras. As in the Mirai attack, the resources of a companys network can be conscripted in support of a global DDNS attack. Since resources are involved in doing the hackers criminal bidding, there will be little bandwidth and processing left for legitimate work, slowing operations to a crawl and, oh, by the way, leaving protected company data wide open for stealing.

Shut it down: Sometimes a hacker has no other end in mind than the mischief involved in making another persons life miserable. There are so many ways to penetrate what passes for security in todays IoT networks Man in the Middle, spoofing, cloning, data sinkholes that a bad actor can sit back and take his or her pick. Once inside, its an easy matter to shut down or destroy devices, infect them with malicious code, and make a network permanently unusable. When you might have spent thousands of dollars to create this smart home network in the first place, losing it all just because some tech whiz was bored on a Sunday afternoon is likely not your definition of a good time.

Chapter 4

Proposed System

Blockchain technology can be used in tracking billions of connected devices, enable the processing of transactions and coordination between devices; allow for significant savings to IoT industry manufacturers. This decentralised approach would eliminate single points of failure, creating a more resilient ecosystem for devices to run on. The cryptographic algorithms used by blockchains, would make consumer data more private.

The ledger is tamper-proof and cannot be manipulated by malicious actors because it doesnt exist in any single location, and man-in-the-middle attacks cannot be staged because there is no single thread of communication that can be intercepted. Blockchain makes trustless, peer-to-peer messaging possible and has already proven its worth in the world of financial services through cryptocurrencies such as Bitcoin, providing guaranteed peer-to-peer payment services without the need for third-party brokers.

The decentralized, autonomous, and trustless capabilities of the blockchain make it an ideal component to become a fundamental element of IoT solutions. It is not a surprise that enterprise IoT technologies have quickly become one of the early adopters of blockchain technologies.

In an IoT network, the blockchain can keep an immutable record of the history of smart devices. This feature enables the autonomous functioning of smart devices without the need for centralized authority. As a result, the blockchain opens the door to a series of IoT scenarios that were remarkably difficult, or even impossible to implement without it.

By leveraging the blockchain, IoT solutions can enable secure, trustless messaging between devices in an IoT network. In this model, the blockchain will treat message exchanges between devices similar to financial transactions in a bitcoin network. To enable message exchanges, devices will leverage smart contracts which then model the agreement between the two parties.

One of the most exciting capabilities of the blockchain is the ability to maintain a duly decentralised, trusted ledger of all transactions occurring in a network. This capability is essential to enable the many compliance and regulatory requirements of industrial IoT applications without the need to rely on a centralised model.

4.1 Combining blockchain and IoT

There are several clear advantages to the idea of building smart machines able to communicate and operate via blockchain.

Firstly, there is the issue of oversight. With data transactions taking place between multiple networks owned and administered by multiple organizations, a permanent, immutable record means custodianship can be tracked as data, or physical goods, pass between points in the supply chain. Blockchain records are by their very nature transparentactivity can be tracked and analyzed by anyone authorized to connect to the network.

If something goes wrong, breakages occur, data leaks where it shouldnt, then the blockchain record makes it simple to identify the weak link and, hopefully, take remedial action.

Secondly, the use of encryption and distributed storage means that data can be trusted by all parties involved in the supply chain. Machines will record, securely, details of transactions that take place between themselves, with no human oversight.

Without the private keys giving write-access to the blockchain (which in this case would be held by machines), no human will be able to overwrite the record with inaccurate information.

Thirdly, the smart contract facilities provided by some blockchain networks, such as Ethereum, allow the creation of agreements which will be executed when conditions are met. This is likely to be highly useful when it comes to, for example, authorizing one system to make a payment, when conditions indicate that delivery of a service has been provided.

Fourthly, blockchain offers the potential of greatly improving the overall security of the IoT environment. Much of the data generated by IoT is highly personalfor example, smart home devices have access to intimate details about our lives and daily routines. This is data that needs to be shared with other machines and services in order to be useful to us. But it also means there are far more openings for hackers to potentially attack us. Business and governments invested in IoT also have to contend with this increased scope for a data breach by criminals, rivals or foreign enemies.

Above the fact that it will offer new opportunities, it is even possible that blockchain and IoT convergence will become a necessity at some point. If the current IoT paradigmdevices connected via a centralized cloud storage and processing servicecontinues, then systems are likely to become increasingly bloated, as data volumes, as well as the number of connected devices, continue to increase.

These cloud services are likely to become bottlenecks as the amount of data pumped through them increases. Blockchains can remedy this thanks to their distributed nature. Rather than an expensive, centralized data center, a blockchain data storage network is duplicated across the hundreds (or potentially thousands, or millions) of computers and devices which make up the network. This huge amount of redundancy means data will always be close at hand when its needed, cutting down transfer times and meaning one server failure will be of no consequence to business activity.

4.2 Advantages

Four ways IoT can exploit blockchain technology:

1. Trust building
2. Cost reduction
3. Accelerated data exchanges
4. Scaled security

Here are some ways the distributed architecture of blockchain can help solve many of these security and trust challenges:

- Blockchain can be used to track the sensor data measurements and prevent duplication with any other malicious data.
- Deployments of IoT devices can be complex, and a distributed ledger is well suited to provide IoT device identification, authentication and seamless secure data transfer.
- Instead of going through a third party for establishing trust, IoT sensors can exchange data through a blockchain.

- A distributed ledger eliminates a single source of failure within the ecosystem, protecting a IOT device data from tampering.
- Blockchain enables device autonomy (smart contract), individual identity, integrity of data and supports peer to peer communication by removing technical bottlenecks and inefficiencies.
- The deployment and operation costs of IoT can be reduced through blockchain since there is no intermediary.
- IoT devices are directly addressable with blockchain, providing a history of connected devices for troubleshooting purposes.

Chapter 5

DROPS System Model

A cloud that consists of M nodes, each with its own storage capacity. Let Si represents the name of ith node and si denotes total storage capacity of Si Communication time between Si and Sj is the total time of all of the links within a selected path from Si to Sj represented by c(i, j). Consider N number of file fragments such that Ok denotes k -th fragment of a file while ok represents the size of k-th fragment. Pk denote the primary node that stores the primary copy of Ok, replication scheme for Ok denoted by Rk is also stored at Pk and Whenever there is an update in not as an independent document. Please do not revise any of the current designations Ok , the updated version is sent to Pk that broadcasts the updated version to all of the nodes in Rk. Let colSi store the value of assigned color to Si. The colSi can have one out of two values, namely: open color and close color. The value open color represents that the node is available for storing the file fragment. The value close color shows that the node cannot store the file fragment The set T is used to restrict the node selection to those nodes that are at hop-distances not belonging to T.

In the DROPS methodology, not to store the entire file at a single node. The DROPS methodology fragments the file and makes use of the cloud for replication. The fragments are distributed such that no node in a cloud holds more than single fragment, so that even a successful attack on the node leaks no significant information.

Symbols	Meanings
M	Total number of nodes in the cloud
N	Total number of fragments to be placed
O_k	k-th fragment of the file
o_k	Size of O_k
S^i	i-th node
s_i	Size of S^i
cen_i	Centrality measure of S^i
col_{S^i}	Color assigned to S^i
r_k^i	Number of reads for O_k from S^i
R_k^i	Aggregate read cost of r_k^i
w_k^i	Number of writes for O_k from S^i
W_k^i	Aggregate write cost of w_k^i
NN_k^i	Nearest neighbour of S^i holdingO_k
c(i,j)	Communication cost between S^i and S^j
P_k	Primary node for O_k
R_k	Replication schema for O_k
RT	Replication Time

Table 5.1: Notations and their meanings.

In the DROPS methodology, user sends the data file to cloud. The cloud manager system(a user facing server in the cloud that entertains users requests) upon receiving the file performs:

1. fragmentation

2. first cycle of nodes selection and stores one fragment over each of the selected node and

3. second cycle of nodes selection for fragments replication. The cloud manager keeps record of the fragment placement and is assumed to be a secure entity.

Chapter 6

DROPS Architecture

This system mainly consist of 3 modules:

6.1 Cloud Client

Cloud client should be Data owner or Data user.

6.1.1 Data Owner

Data owner is responsible for uploading file on cloud as well as view files uploaded by him or others. Data owner has information about the placed fragment and its replicas with their node numbers in cloud.

6.1.2 Data User

Data user is the one who is responsible for downloading files or view files uploaded by others. To download file from cloud he has to be authenticated user otherwise he will be considered as attacker.

6.2 Cloud Server

6.2.1 Fragmentation

This approach is used for fragmenting the file for security purpose at sever side. This approach runs the Fragmentation algorithm. It has file as input and produces the file fragments as output.

6.2.2 Replication

This approach creates replicas (duplicate copy) of fragments. These replicas are useful when one of fragment is corrupted by attacker then to provide file for user admin replaces its replica at that place and combine all fragments and send file to authenticated user or data owner. To make replicas of file fragments this approach runs replication algorithm which takes input as fragments and produces its replicas as output.

6.2.3 Allocation

After the file is spitted and replicas are generated then allocate that fragments at cloud server for storing data. While storing or allocating that fragments, consider security issues. So the proposed method using T- Coloring Graph concept for placing fragments at different nodes on cloud server. This approach runs Fragment allocation algorithm which takes input as fragments and produces the output as fragments allocated with node numbers.

6.3 Admin

Admin is an authorized person who has rights to validate authorized data owner and user. He is also responsible for allocation of block and maintains information and authentication.

6.4 Fragment Placement

In the DROPS methodology, not to store the entire le at a single node. The DROPS methodology fragments the le and makes use of the cloud for replication. The fragments are distributed such that no node in a cloud holds more than a single fragment, so that even a successful attack on the node leaks no signicant information. The DROPS methodology uses controlled replication where each of the fragments is replicated only once in the cloud to improve the security. Although, the controlled replication does not improve the retrieval time to the level of full-scale replication, it signicantly improves the security.The fragment placement strategy is presented in the algorithm given below.

Inputs and initializations:
$O = \{O_1, O_2, ..., O_N\}$
$o = \{sizeof(O_1), sizeof(O_2), ..., sizeof(O_N)\}$
$col = \{open\_color, close\_color\}$
$cen = \{cen_1, cen_2, ..., cen_M\}$
$col \leftarrow open\_color \forall$ i
$cen \leftarrow cen_i \forall$ i
Compute:
for each $O_k \in O$ **do**
  select $S^i \mid S^i \leftarrow$ indexof(max(cen_i))
  if $col_{S^i} = open\_color$ and $s_i >= o_k$ then
    $S^i \leftarrow O_k$
    $s_i \leftarrow s_i - o_k$
    $col_{S^i} \leftarrow close\_color$
    $S^{i\prime} \leftarrow distance(S^i, T)$ ▷ /*returns all nodes at distance T from S^i and stores in temporary set $S^{i\prime}$*/
    $col_{S^{i\prime}} \leftarrow close\_color$
  end if
end for

Figure 6.1: Algorithm for Fragment Placement

Initially, all of the nodes are given the open color. Once a fragment is placed on the node, all of the nodes within the neighborhood at a distance belonging to T are assigned close color. In the aforesaid process, lose some of the central nodes that may increase the retrieval time but achieve a higher security level. If somehow the intruder compromises a node and obtains a fragment, then the location of the other fragments cannot be de- termined. The attacker can only keep on guessing the location of the other fragments. However,the probability of a successful coordinated attack is extremely minute. The pro- cess is repeated until all of the fragments are placed at the nodes. The above algorithm represents the fragment placement methodology.

6.5 Fragment Replication

In addition to placing the fragments on the central nodes, this approach also per- form a controlled replication to increase the data availability, reliability, and improve data retrieval time. Place the fragment on the node that provides the decreased access cost with an objective to improve retrieval time for accessing the fragments for reconstruction of original le. While replicating the fragment, the separation of fragments as explained in the placement technique through T- coloring, is also taken care off.

```
for each O_k in O do
    select S^i that has max(R^i_k + W^i_k)
    if col_{S^i} = open_color and s_i >= o_k then
        S^i ← O_k
        s_i ← s_i − o_k
        col_{S^i} ← close_color
        S^i' ← distance(S^i, T)      ▷ /*returns all nodes at
        distance T from S^i and stores in temporary set S^i'*/
        col_{S^i'} ← close_color
    end if
end for
```

Figure 6.2: Algorithm for Fragment Replication

In case of a large number of fragments or small number of nodes, it is also possible that some of the fragments are left without being replicated because of the T-coloring. As discussed previously, T-coloring prohibits to store the fragment in neighborhood of a node storing a fragment, resulting in the elimination of a number of nodes to be used for storage. In such a case, only for the remaining fragments, the nodes that are not holding any fragment are selected for storage randomly. The replication strategy is presented in the algorithm given below.

6.6 Replication Time

Aim is to minimize the overall total network transfer time or replication time (RT) or also termed as replication cost (RC). The RT is composed of two factors: (a) time due to read requests and (b) time due to write requests.

- The total read time of Ok by Si from NNi k is denoted by Ri k and is given by:

$$R^i_k = r^i_k o_k c(i, NN^i_k)$$

- The total time due to the writing of Ok by Si ad- dressed to the Pk is represented as Wi k and is given:

$$W^i_k = w^i_k o_k (c(i, P_k) + \sum_{(j \in \neq R_k), j \neq i} c(P_k, j))$$

- The overall RT is represented by:

$$RT = \sum_{i=1}^{M} \sum_{k=1}^{N} (R^i_k + W^i_k)$$

- The storage capacity constraint states that a file fragment can only be assigned to a node, if storage capacity of the node is greater or equal to the size of fragment.
- To handle the download request from user, the cloud manager collects all the fragments from the nodes and reassemble them into a single file. Afterwards, the file is sent to the user.

6.7 Features of DROPS

- It ensures that even in case of a successful attack, no meaningful information is revealed to the attacker.
- A successful attack on a single node must not reveal the locations of other fragments within the cloud.
- The nodes storing the fragments, are separated with certain distance by means of graph T-coloring to prohibit an attacker of guessing the locations of the fragments.
- Does not rely on the traditional cryptographic techniques for the data security.
- Faster
- Higher level of security with slight performance overhead.

Chapter 7

Conclusion and future scope

The DROPS methodology is a cloud storage security scheme that jointly deals with the security and performance in terms of retrieval time. The data file was fragmented and the fragments are scattered over multiple nodes. The nodes were separated by means of T-coloring. The fragmentation and dispersal make sure that no significant information was obtainable by an antagonist in case of a successful attack. No node in the cloud, stored more than a single fragment of the same file.

The scope of this work is divide a file into fragments, and replicate the fragmented data over the cloud nodes. Each of the nodes stores only a single fragment of a particular data file that ensures that even in case of a successful attack, no meaningful information is revealed to the attacker.

Bibliography

[1] Lade Mayuri S.Lolage Pranali S. Nagare Mayuri S.Sadaphal Priyanka V. Prof.Jorwekar Y.S, *"An Study on-Cloud Security Using DROPS Technique"*, Vol. 2 Issue. 6, 2016.

[2] Priya Patni, Dr. S.N Kakarwal , *"Optimal performance and security by division and replication of data in cloud"*, Global Journal of Engineering Science and Research Management, Nov. 2015.

[3] Nidhi Jain, Archana jadhav, *"A Survey Paper on Drops: Division and Replication of Data in Cloud for Optimal Performance and Security"* , International Journal of Innovative Research in Computer and Communication Engineering, vol. 4, Oct. 2016

[4] Mazhar Ali, K. Bilal, S. U. Khan, B. Veeravalli, K. Li, Albert Y. Zomaya and A. J. R. Neves, *"dropd: Division and Replication of Data in the Cloud for Optimal Performance and Security"*, IEEE transactions of cloud computing., vol. 13, no. 11, pp. 14111418, Nov. 2015.