Blockchain For Large-Scale Internet of Things Data Storage and Protection

Seminar Report

submitted in partial fulfillment of the requirement
for award of Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

of

APJ Abdul Kalam Technological University

Submitted by

RASNA K

Department of Computer Science and Engineering
Mar Athanasius College of Engneering
Kothamangalam

Mar Athanasius College of Engineering

KOTHAMANGALAM

Blockchain For Large-Scale Internet of Things Data Storage and Protection

Bonafide record of seminar done by

RASNA K
Register Number: LMAC15CS066

submitted in partial fulfillment of the requirement
for the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

of

APJ Abdul Kalam Technological University

.....
Prof Joby George
Seminar Coordinator

.....
Dr. Surekha Mariam Varghese
Head of the Department

.....
Prof Neethu Subhash
Seminar Coordinator

ACKNOWLEDGEMENT

First and foremost, I sincerely thank the 'God Almighty' for his grace for the successful and timely completion of the seminar.

I express my sincere gratitude and thanks to Dr. Solly George, Principal and Dr. Surekha Mariam Varghese, Head Of the Department for providing the necessary facilities and their encouragement and support.

I owe special thanks to the staff-in-charge Prof. Joby George and Prof. Neethu Subhash for their corrections, suggestions and sincere efforts to co-ordinate the seminar under a tight schedule.

I express my sincere thanks to staff members in the Department of Computer Science and Engineering who have taken sincere efforts in helping me to conduct this seminar.

Finally, I would like to acknowledge the heartfelt efforts, comments, criticisms, co-operation and tremendous support given to me by my dear friends during the preparation of the seminar and also during the presentation without whose support this work would have been all the more difficult to accomplish.

ABSTRACT

Storing and protecting the large volume of IoT data has become a significant issue. Traditional cloud based IoT structures impose extremely high computation and storage on the cloud servers. The centralized server bring significant trust issues. To avoid this problem we propose a distributed data storage scheme employing blockchain. The blockchain offers a convenient platform for distributed data storage and protection. The miners are work cooperately to create block as public ledger that validate and record transactions. The data can be stored in DHT(Distributed Hash Table) . The authentication of the requester is handled by the distributed blockchain miners instead of a trusted centralized server. The main advantages are decentralized storage , no centralized trusted server, traceability and accountability. To get a clear definition of the transaction and data trading , elaborate how data trading can be efficiently and effectively achieved.

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATION

IoT	Internet of Things
DHT	Distributed Hash Tables
KGC	Key Generation Center
ACL	Access Control List
IBE	Identity Based Encryption
P2P	Peer-to-Peer
PoW	Proof-of-work
PKI	Public Key Infrastructure

Introduction

Internet of Things (IoT) is an emerging term that de-scribes the ubiquitous connection of everyday objects[1]-[4] .With the dramatically increasing deployment of IoT devices, tremendous interactions among the physical objects are enabled, which brings improved efficiency, accuracy, and economic benefits while reducing human interventions[5] . It is estimated by Gartner that there will be over 20 billion connected IoT devices all over the world by 2020[6] . The great amount of these devices brings lots of challenges in data storage. How to efficiently store the large-scale IoT data, and how to protect the data are issues of great significance.

IoT applications such as smart grid and implantable medical system, involve tremendous data aggregations. In a traditional cloud-based IoT structure, a centralized cloud server collects and controls all the data, which brings two drawbacks: 1) the cloud server needs very high storage capacity to store the IoT data; 2) sensitive data can be easily leaked from the server. For example, server might trade sensitive data with other entities without notifying the data owner. A decentralized structure will properly handle these issues: Data can be transferred and controlled in a distributed manner as opposed to that in a centralized structure.

Blockchain offers a convenient platform for distributed data storage and protection. In blockchain, a group of users, also known as miners, work cooperately to create blocks as a public ledger that validate and record transactions. In an IoT application such as implantable medical system, data can be stored in Distributed Hash Tables (DHTs) [7] while the pointer to the DHT storage address can be stored in the blockchain. When an entity requests data from the DHT, the blockchain will decide whether the access can be granted or not, i.e., the authentication of the requester is handled by the distributed blockchain miners instead of a trusted centralized server.

IoT devices have low computational power, and they are not capable of conducting complex computations. Edge computing is one way to help mitigate this problem. Edge computing, as opposed to cloud computing, is a method to process data at the network edge, rather than in the remote cloud [8]. Edge computing brings realtime computations and communications by leveraging nearby edge servers. A lot of companies such as Intel, Amazon, and Cisco are developing edge-based services to facilitate IoT development. An edge device could be any computing resource residing between data source and cloud. In our scheme a smartphone or any local computing device can be used as edge server. We assume that the communications between an edge server and IoT devices are secure. This is a reasonable assumption as an edge server is placed in the same local network as the IoT devices, aiming to help the IoT devices perform certain kinds of computations. For example, in implantable medical system, an edge server within physical proximity of a patient can collect health data from the implanted sensors. In our scheme, an edge server has two duties: 1) helping the IoT

devices perform cryptographic computations; and 2) collecting data from IoT devices and forwarding data to the DHT.

Fig. 1 illustrates the proposed structure for IoT data storage using blockchain. In this structure, a group of IoT sensors send realtime data to an edge device that manages the data storage in DHT through the blockchain. Blockchain works as a "trusted third party" in the following two ways:

1) Before an edge device forwards data to DHT, it posts a "transaction" to the blockchain, announcing that the data belonging to certain IoT device will be stored in an address of DHT. Blockchain verifies the transaction and records the identity of the IoT device and the storage address. In this way, blockchain helps manage data storage.

2) When an IoT device requests data from DHT, it posts a "transactions" to the blockchain. The blockchain will work as a "trusted third party" to authenticate the requester. If the transaction is validated and written into a block, the DHT node storing the data will send data to the requester. Therefore, authentication is performed through the blockchain, without a trusted server.

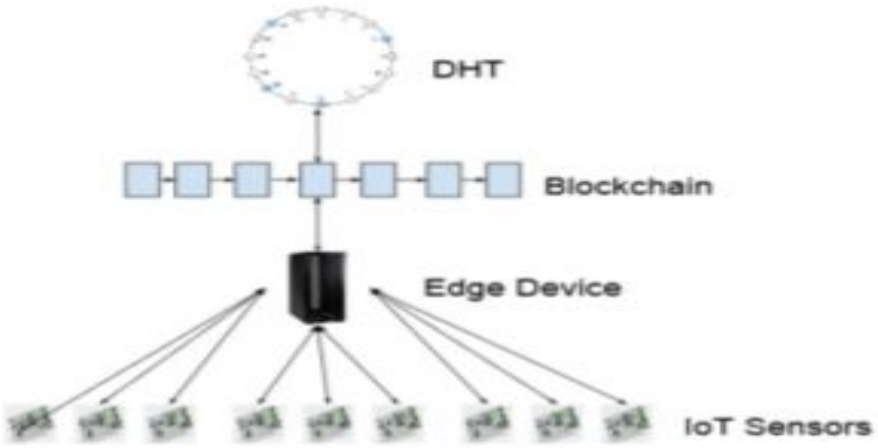

Figure 1.1: The structure of data storage scheme with blockchain

Applying blockchain in large-scale IoT data storage and protection is challenging. The most significant issue is how to manage the identities of IoT devices so that authentication can be easily done through blockchain. Note that the miners take charge of authentication when an entity requests to access the data. However, the miners should not have any knowledge of the credentials to perform authentication. This implies that the system must have some cryptography mechanism that allows an IoT device to be identified and verified by other parties without utilizing a secret value such as password.

Traditional Public Key Infrastructure (PKI) with certificate introduces too much redundency. Identity Based Encryption (IBE) is one alternative that enables a user's public key to be created using his identity, so that other entities can verify the user's identity through the public key. However, IBE suffers from the key Escrow problem: the Key Generation Center (KGC) is aware of the user's private key, and there is no way to authenticate a user unless we assume the KGC is completely trusted. This is where Certificateless Cryptography steps

in. Certificate-less cryptography is different than IBE as a user's public key is generated by both the user's identity and some secret of which the KGC is not aware. Therefore, KGC has no knowledge of the user's private key, while a public key can be verified whether it belongs to certain user or not. The only drawback of certificateless cryptography compared to IBE is that the public key of a user, even though can be verified, needs to be pre-broadcasted. The good thing is, blockchain offers a platform to share public information, which means that the user's public key can be shared via the blockchain. For example, an IoT device can append its public key to his data access request and sends the request to the blockchain where the blockchain miners are able to verify the public key of the device.

Related Works

The rapid growth of Internet of Things promotes the sharp growth of data, and brings lots of challenges in big data storage, analytic and management. For data storage, most researchers focus on building databased management models to mitigate the massive IoT data storage problem. NoSQL and Hadoop databases are attracting most of the attention. Besides, the authors in proposed a new IoT data storage platform based on combined multiple database models. Our work is distinct from these researches as we focus on constructing a novel distributed data storage system based on blockchain technology. To process the massive IoT data, edge computing has become a popular enabled technology.

Blockchain, as a decentralized cryptocurrency mechanism, has received lots of attention. Most research are focused on the blockchain mechanism themselves, aiming to solve the scalability problem or reducing the power consumption of a PoW. There are some research utilizing blockchain on practical applications. In 2014, for the first time, Bitcoin is proposed to form a lottery protocol as a multiparty computation method. The authors demon-strated that Bitcoin provides an attractive way to construct a "timed commitments" and a multiparty protocol can be constructed by letting the miners emulating a trusted server. Bitcoin was also proposed as an incentive mechanism for distributed P2P application [48], in which the relay of the network helps to transfer messages and gets reward from the Bitcoin system. In 2016, Christidis et al. discussed how blockchain could possibly work in the IoT domain. They illustrated several IoT applications that blockchain might fit in, but did not give detailed solutions. The most related work is, in which Zyskind et al. proposed a decentral-ized personal data management system using blockchain and illustrated how blockchains could become a vital resource in trusted computing. This work assumed the existence of a secure channel through which the data requester shared secret keys with the data owner. The authentication was processed through the blockchain, and only the users holding the secret keys are able to access data. The work has its limitations as 1) the assumption of such a secure channel is not practical; 2) the secret keys will be leaked to the blockchain miners who perform the authentication. In our research, we take advantage of certificateless cryptography to achieve effective and efficient authentications without a secure channel.

Certificateless cryptography was proposed in 2003 to overcome the key escrow problem in identity based encryption. Compared to traditional PKI, certificateless cryptography does not require certificates to guarantee the authenticity of public keys. Followed by this pioneer research, Seo et al. proposed a more efficient way to construct certificateless cryptography withouting using paring. The most Efficient Certificateless signature scheme to date was proposed that involve only simple hash operations for signing. Also, certificateless signcryption algorithm was proposed in order to reduce the computation cost and encryption and signing, such that it can be easily carried by the lightweight IoT devices. The above proposed protocols suffer from one problem: it is hard to broadcast the public key of a user. Our proposed scheme with blockchain overcomes this problem as blockchain is a perfect media for

spreading public keys to all users in one network.

The Proposed Method

. In a traditional cloud-based IoT structure, a centralized cloud server collects and controls all the data, which brings two drawbacks first one is the cloud server needs very high storage capacity to store the IoT data and second one is sensitive data can be easily leaked from the server. To overcome this problem we proposed blockchain, it offers a convenient platform for distributed data storage and protection. In blockchain, a group of users, also known as miners, work cooperately to create blocks as a public ledger that validate and record transactions. In an IoT application such as implantable medical system, data can be stored in Distributed Hash Tables (DHTs) [7] while the pointer to the DHT storage address can be stored in the blockchain. When an entity requests data from the DHT, the blockchain will decide whether the access can be granted or not, i.e., the authentication of the requester is handled by the distributed blockchain miners instead of a trusted centralized server.

3.1 Internet of Things

The Internet of things (IoT) is the network of devices, vehicles, and home appliances that contain electronics, software, actuators, and connectivity which allows these things to connect, interact and exchange data.

IoT involves extending Internet connectivity beyond standard devices, such as desktops, laptops, smartphones and tablets, to any range of traditionally dumb or non-internet-enabled physical devices and everyday objects. Embedded with technology, these devices can communicate and interact over the Internet, and they can be remotely monitored and controlled.

3.1.1 Three Tier Architecture

The IoT technology stack consists of three tiers: sensor devices, gateways, and the data center or cloud IoT platform. As explained in the paper, “a typical IoT solution is characterized by many devices (i.e., things) that may use some form of gateway to communicate through a network to an enterprise back-end server that is running an IoT platform that helps integrate the IoT information into the existing enterprise.”

The device tier focuses on information gathering via sensors. Because sensors are so tiny and inexpensive, they can be embedded in many different types of devices, including mobile computing devices, wearable technology, and autonomous machines and appliances. They capture information about the physical environment, such as humidity, light, pressure, vibration and chemistry. Standards-based wired and wireless networking protocols are used to transmit the telemetry data northbound from the device to the gateway. Northbound data, if you remember from my previous post, is data going from the device through the gateway

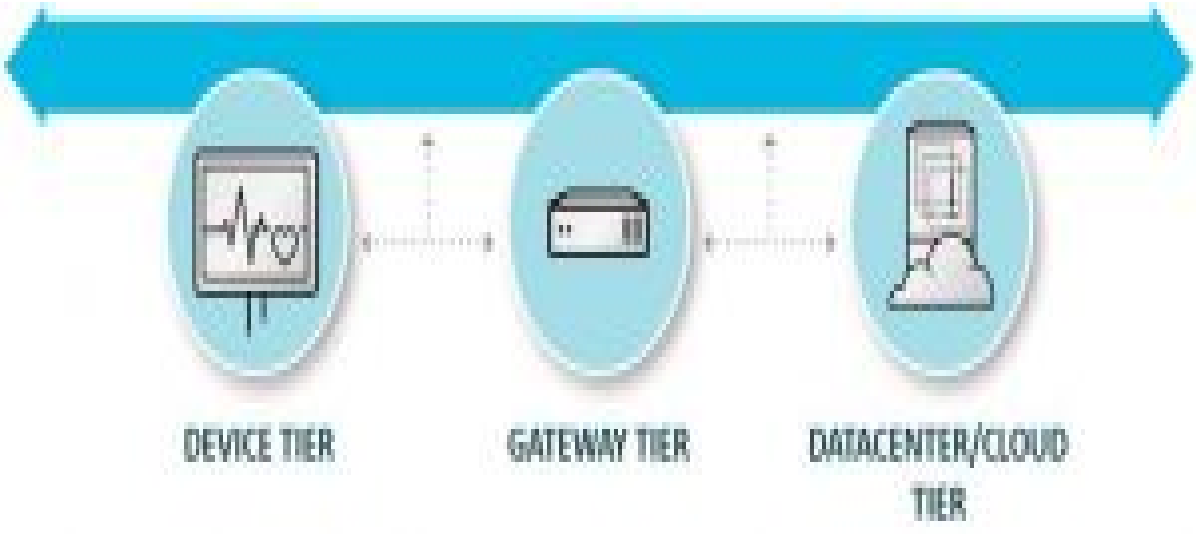

Figure 3.1: Three Tier Architecture

up to the cloud. It is typically telemetry data, but can be command and control requests. Southbound data, on the other hand, is generally command-and-control data that goes from the cloud to the gateway or from the cloud, through the gateway, to the device.

The gateway sometimes referred to as the control tier, acts as an intermediary that facilitates communications, offloads processing functions and drives action. Because some sensors generate tens of thousands of data points per second, the gateway provides a place to preprocess the data locally before sending on to the data center/cloud tier. When data is aggregated at the gateway, summarized and tactically analyzed, it can minimize the volume of unnecessary data forwarded on. Minimizing the amount of data can have a big impact on network transmission costs, especially over cellular networks. It also allows for critical business rules to be applied based on data coming in. The control tier is bidirectional. It can issue control information southbound, such as configuration changes. At the same time, it can respond to northbound device command-and-control requests, such as a security request for authentication.

The data center/cloud tier performs large-scale data computation to produce insights that generate business value. It offers the back-end business analytics to execute complex event processing, such as analyzing the data to create and adapt business rules based on historical trends, and then disseminates the business rules downstream (southbound). It needs to scale both horizontally (to support an ever growing number of connected devices) as well as vertically (to address a variety of different IoT solutions). Core functions of an IoT data center/cloud platform include connectivity and message routing, device management, data storage, event processing and analysis, and application integration and enablement.

3.2 Applying Blockchain and Edge computing in IoT Applications

3.2.1 Blockchain Description

Blockchain works as a peer-to-peer network without a trusted third party. Fig.2 illustrates a typical blockchain net-work. In this network, a user creates a transaction and sends it to a peer-to-peer (P2P) network. The workers in the P2P network will utilize one agreed algorithm to determine how to write transactions into an empty block, and the transactions will only be validated after they are written into a block. As time goes on, there will be more and more blocks forming a blockchain. Blockchain can be viewed as a continuously growing list of records that can be seen by any user in the network. However, Blockchain is not merely a data structure, but more generally considered as the technology that enables a large group of users agreeing on writing transactions into the blocks. The key point of blockchain technology is how to design an algorithm of consensus, and most designs are based on Byzantine Fault Tolerance (BFT) .

The early version of blockchain, such as Bitcoin, is imple-mented by a clever mechanism called Proof of Work (PoW). In such a system, each block Bn contains a list T of transactions, a random salt, and the hash of the previous block. A new block can be created only if a random salt is found for a block Bn such that Hash(Bn 1; T; salt) starts with a certain number of 0s. Finding a valid salt is a crypto puzzle that needs lots of computational power, and the computing process is called Mining. Once a crypto puzzle is solved, a new block Bn is created and the transactions T can be written into this new block. The smart idea behind PoW is that the transactions in the system are performed and audited by a large group of miners who take great efforts to run this system. Therefore, it is reasonable to assume that the system is secure as long as the majority of miners are honest. However, this scheme does not fit our structure. For example, the most famous cryptocurrency system using PoW – the Bitcoin, is only able to process seven transactions per second [16]. Furthermore, PoW consumes too much computation power without creating true financial welfare, which renders the system too costly for IoT systems.

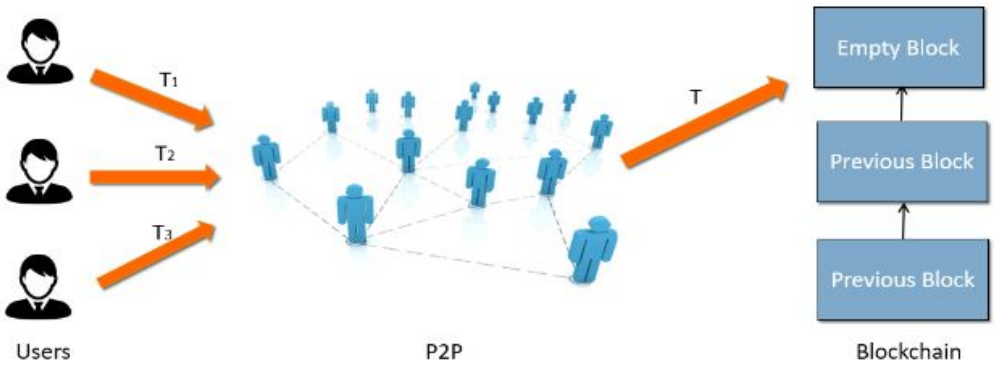

Figure 3.2: Blockchain Network

To tackle the power-consumption issues of the Bitcoin, other mechanisms have been explored to replace PoW such as Proof of Stake (PoS), Proof of Space (PoSpace) [18], and Rem (Intel SGX) etc. We found that Rem is the most suitable mechanism among them. The idea of "Rem" is to replace the power-consuming "Proof of Work (PoW)" with "Proof of Useful Work (PoUW)" where the miners provide trustworthy reports on CPU cycles devoted to inherently useful workloads. PoUW is achieved by adopting Intel's Software Guard extensions "SGX", which permits trustworthy code to be executed in an isolated and tamper-free environment, and SGX can prove remotely the result of such executions. To put it simply, this smart idea is to let the miners compute useful work for Intel, and in return Intel provides workers with a proof of their work so that the workers can build a block. Generally speaking, any company can construct similar structures to the Intel "SGX" to outsource its work to the miners. Such a mechanism is practical and is suitable for applications that utilize blockchain.

3.2.2 Blockchain Transactions

We are using blockchain as a platform to serve IoT data storage and protection, rather than utilizing blockchain as a cryptocurrency. Therefore, "transactions" in our scheme are different from those in the cryptocurrency schemes such as Bitcoin and Ethereum. A transaction in our scheme is any request sent from an IoT device asking for services of data storage or data access. For example, a medical sensor A sends a transaction claiming that it stores data in a certain address Addr in DHT. The transaction can be written in the following form: T = (IDA; T imestamp; Action = store data in Addr). When a doctor's implantable medical device (IMD) programmer, which we shall call it B, requests data from A, it will post a transaction to the blockchain, requesting sensor A's data stored in DHT in the following form: T = (IDB; IDA; T imestamp; Action = access data in Addr). Note that a DHT node in "Addr" does not send data to a requester until the DHT node confirms that the transaction of the request has been verified and written into the blockchain.

3.2.3 Miner Awards

The proposed system is built upon the blockchain run by a large group of miners. One critical question is how to incentivize miners to work for IoT applications. We believe there is sufficient income for mining the blockchain in the proposed scheme from the following three aspects:

1) The proposed structure eliminates the centralized server, and the service fee from a traditional server should be transferred to the miners in the blockchain. This can be done by depositing transaction fees in the blockchain transactions. Once a transaction is done, the miners can get the transaction fee immediately.

2) As we discussed before, a practical blockchain mecha-nism for IoT applications, such as Rem, utilizes miners to compute useful work for large companies, such as Intel. In return, these companies will pay back miners for their work.

3) Blockchain itself is a cryptocurrency that creates block awards all the time. Though we are utilizing blockchain for data storage services, the operations of blockchain will inevitability create block awards that can be split among the miners as their rewards.

3.2.4 Edge Computing

An edge device plays a significant role in the proposed blockchain based IoT storage scheme. It not only relays mes-sages/transactions for the IoT devices, but also helps manage data storage and perform computations. We list the roles of an edge device as follows:

1) Manage the identities of IoT devices. An edge server stores a copy of identities of all nearby IoT devices and helps each device build a pair of keys for authentication through a KGC.

2) Create transactions for IoT devices. A valid transaction should include the signature of an IoT device, and the signing process should be conducted by the edge server. Also, sensitive data should be encrypted before sending out for storage, and edge server can help handle the computations.

3) Collect and forward data to DHT. The edge server continuously collects data from nearby devices. It de-termines the DHT address to store the data and sends the encrypted data to the designated address.

3.2.5 Security Model

In our proposed scheme, we assume the communications between an edge device and an IoT device are secure. All cryptographic computations are processed at the edge device to reduce the workload of an IoT device. A KGC is utilized to help establish keys for IoT devices, and this KGC is semi-honest, which means that KGC follows the rule to perform computations, but tries to infer sensitive information collected by the IoT devices. In our design with certificateless cryptography, KGC is not able to obtain any user's private key. Data storage and protection are performed solely by the blockchain, without intervention of any other entity. Therefore, the security of our scheme is based on the security of the blockchain mechanism.

3.3 Authentication of Blockcain Transactions

Establishing the asymmetric key pairs using traditional PKI or IBE both have their limitations, as we discussed before. Efficient authentication for IoT devices should be enforced. With certificateless cryptography, an IoT device can be easily verified. For example, an IoT device posts a transaction signed with its private key $\mathrm{SK}_A, and appends its public key \mathrm{PK}_A$ and ID $ID_A$to the transaction. A miner is able to check that: 1) this transaction is indeed signed with a private key associated with PK_A, and 2) PK_A does belong to ID_A. In this way, it can be easily verified that whether the transaction was created by an IoT device with ID_A.

A simpler mechanism adopted in current cryptocurrency also achieves the above result, but it does not fit in our applica-tions. For example, Bitcoin lets each user create a pair of keys (PK_i, SK_i) based on Elliptic-Curve Cryptography (ECC), and a transaction is verified by a signature scheme based on Elliptic Curve Digital Signature Algorithm (ECDSA). The public key PK_i is hashed twice as an address of the user Addr. Therefore, other users are able to check if a public key PK_i belongs to an address by checking if $H(H(PK_i)) = Addr$. This brings lots of convenience compared to traditional PK_I as it eliminates the burden of traditional digital certificates and all users are able to verify a user's public key. However, this system does not achieve user authentication or accountability. Firstly, constantly changing identities will make authentication hard. In a system with full anonymity, the only way to check whether an IoT device has rights to access certain data or not is to verify some credentials only known to this device. However, blockchain is an open ledger and devices' credentials will be exposed. Zero-knowledge Proof is a theoretical solution but hard to deploy. Secondly, a system with full anonymity is not accountable. Therefore, assigning IoT devices with unique identities is necessary to maintain a secure and accountable storage system.

We proposed to utilize certificateless cryptography to achieve both authentication and accountability. In certificate-less cryptography, each IoT device's unique identity can be used to create the public key, and all other users are able to verify that the public key belongs to this unique identity. Also, if an IoT device revokes the old key, it is able to create a new public key pair using its unique identity. The new key pair is different from the old one, but it is still bounded with the unique ID. In the following part, we give a detailed description of certificateless cryptography, and how it can be used in our proposed scheme. To make it simple, we omit the description of an edge device in our algorithms. However, all the compu-tations of an IoT device are performed at the edge device

3.3.1 Certificateless Cryptography

Certificateless cryptography was derived from the IBE in order to solve the key escrow problem. In certificateless cryptography, a key generation center (KGC) creates a partial private key based on a user's identity; the user utilizes the partial private key and its own secret value to establish a private key. Since the secret value is only known to the user, KGC is not able compute the private key. Therefore, key escrow problem in IBE is avoided. The user also creates the public key based on the secret value and makes it public. In detail, there are five general steps for establishing keys PK_A and SK_A for a user A.

1). Setup$(1^\lambda) \rightarrow (K, MSK)$: The setup algorithm takes security parameter and returns the system parameters K and a secret master key MSK. This algorithm is run by KGC and only KGC knows the value of MSK.

2) PSkeyGen$(K, ID_A; MSK) \rightarrow (PSK_A)$: The partial private key generation algorithm takes the system parameters K, a user A's identity $ID_A \epsilon 0, 1$, and the master key MSK, and outputs a partial private key PSK_A. This algorithm is run by KGC and the output will be transported to entity A.

3) SValGen $(K, ID_A) \rightarrow (X_A)$: The secret value genera-tion algorithm takes the system parameters K and user A's identity ID_A, and outputs secret value X_A. This algorithm is run

by the user and X_A will be used to transform the partial private key to a private key. This algorithm is run by the user.

4) SKeyGen$(K, PSK_A, X_A) \rightarrow (SK_A)$. The private key generation algorithm takes as input the system parameters K, the partial private key PSK_A and the secret value X_A, and returns the private key SK_A. This algorithm is run the the user and only this user knows his private key.

5) PKeyGen $(K, X_A) \rightarrow PK_A$: The public key generation algorithm takes the system parameter K and the secret value X_A to construct the public key PK_A. This algo-rithm is run by the user and PK_A will be broadcast to the public.

The above steps illustrate how to generate the public key pairs using certificateless cryptography. The function of en-crypt, decrypt, sign and verify are described as follows.

1) Encrypt $(K, M\epsilon M, ID_A, PK_A) \rightarrow C\epsilon C \vee \perp$

This algorithm takes the system parameters K, a message M in finite message space, user's identity and public key, and outputs a ciphertext C in ciphertext space.

2) Decrypt$(C\epsilon C, SK_A) \rightarrow M\epsilon M$
This algorithm takes a ciphertext C and user's private key SK_A, and outputs a message M.

3) Sign $(K, M\epsilon M, SK_A) \rightarrow Sig$

This algorithm takes system parameters K, a message M and user's private key SKA, and outputs a signature.

4) Ver $(M\epsilon M, Sig, ID_A, PK_A) \rightarrow Valid_Invalid \vee \perp$
This algorithm takes a message M, a signature Sig, user's ID ID_A and user's private key SK_A, and outputs if the signature is valid or not.

There are many ways to achieve certificateless cryptography, with both lightweight encrypting and signing features achieved and we do not give complete constructions of the cryptography scheme here to save space. We also give a definition of the verification function that is used to check whether PK_A belongs to ID_A or not: VerID$(ID_A, PK_A, K) \rightarrow Valid \vee Invalid \vee \perp$

To make the description of our algorithms clear, we generally denote the encryption to ciphertext C of data M with public key PK by $C = E_{PK}(M)$, and decryption of ciphertext C with private key SK by $M = D_{SK}(C)$. Similarly, we denote the signing of a message M by $Sign_{SK}(M)$, and verification of a signature Sig by $VerP_K(Sig)$.

3.3.2 Advantages

1) Decentralized Storage: The IoT data is stored off-chain in a distributed way, and an entity can easily find the storage address through the blockchain.

2) No Centralized Trusted Server: The access to IoT data is controlled by the majority of the blockchain miners, without any intervention from a trusted server. Users do not need to worry about unauthorized access to his/her data.

3) Traceability and Accountability: Activities such as ac-cessing and modifying the IoT data, can be recorded by the blockchain. No malicious attempts can be made undetected.

3.4 How Blockchain Transactions Work in IoT Applications

In this section, we give details how transactions are pro-cessed in the proposed scheme. To start, an IoT device needs to register in the blockchain network by contacting KGC for establishing its keys. After successfully obtaining a pair of keys, an IoT device is able to post transactions that can be verified by the blockchain.

3.4.1 Registration

To start, KGC broadcasts the system parameters K and keeps a secret master key MSK. All IoT devices should have the knowledge of K. Meantime, an IoT device creates a secret value X_A, and generates its public key using X_A and the system parameter K. When an IoT device would like to register in a blockchain, it will contact the KGC with its ID ID_A. Upon receiving the request, KGC will generate a partial private key PSK_A for this IoT device, sign IDA and PSK_A with his private key SK_K , and sends the signed message back to the IoT device. The IoT device will verify if the message comes from the KGC, and if yes, it will generate private key using P SKA, X_A and K. Note that only this IoT device is able to create the private key because it is the only entity that knows X_A.

Algorithm 1 illustrates an IoT devices' operations in the registration process. SendRequest() and RecvReq() denote the functions of sending and receiving messages, respectively

Algorithm 1 Device Registration

Input: ID_A
Output: PK_A, SK_A
1.procedure KEYGEN$(1^\lambda, ID_A)$
2.X$_A \leftarrow SValGen(K, ID_A)$
3.$SendRequest(ID_A)$
4.$RecvReq(PSK_A, Sign_{SKK}(ID_A))$
5.$V \leftarrow Ver(ID_A, Sign_{SKK}(ID_A), SK_K)$
6.$if V = Validthen:$
7.$SK_A \leftarrow SKeyGen(K, PSK_A, X_A)$
8.$PK_AKeyGen(K, X_A)$
9.$endifreturn$
10.$endprocedure.$

3.4.2 Transactions Description and Verification

After suc-cessfully registered at the blockchain, an IoT device is able to take advantage of the blockchain to store data. Generally, a transaction includes the identity of an IoT device, a timestamp and an action, for example, $T_A = (ID_A, Timestamp, Action)$. An action can

be a claim to store data at a DHT address, a request to access data, or a request to update data. To save space, we use Addr to represent actions on the DHT address. To verify a transaction TA, the miners have to check the following requirements:

1) If the the public key PK_A is derived from the identity ID_A associated with it.
2) If the signed transaction can be verified with the public key PK_A.

By checking the two requirements above, the miners are able to verify whether a transaction TA is created from IDA or not. Algorithm 2 illustrates how to verify a transaction with a procedure VerTrans(). When a transaction is posted, it is automatically marked with a flag s = 0. The flag of successfully verified transactions will be set to "1", and then the verified transactions will be written into a block.

Algorithm 2 Verify A Transaction
Input: T_A, σ_{TA}
Output: a verified T_A
1.procedure VERTRANS(T_A, σ_{TA})
2.$s \leftarrow 0$
3.$V_1 \leftarrow VerID(ID_A, PK_A, K)$
4.$if V_1 = Validthen$
5.$V_2 \leftarrow Ver(T_A, \sigma_{TA}, ID_A, P_{KA})$
6.$elseAbort$
7.$if V_2 = Validthen$
8.$s \leftarrow 1$
9.$elseAbort$
10.$endif$
11.$endif$
12.$return$
13.$endprocedure$

3.4.3 IoT Data Storage and Protection

We take implantable medical applications as an example. The proposed scheme works for various IoT applications such as smart grid, smart home etc. For example, in smart grid, smart meters send data via edge devices to blockchain storage and server/aggregator can access data on billing date. In smart home, personal data generated by IoT devices can be sent to the blockchain storage and the data owner/authorized user is able to access data.

Firstly, a medical device creates an access control list through the edge server, which specifies clearly who can access his data.
Then it creates a transaction $T_A = (PK_A, ID_A, ACL, Addr)$ with its public key, identity, access control list and storage address, creates a validation A appended to the transaction, and publishes TA to the blockchain system. The process is illustrate in Algorithm 3

Algorithm 3 Store Data
Input: ID_A, ACL
Output: a verified T_A
1.$procedureSETACL(ACL)$
2.$CreateT_A = (PK_A, ID_A, ACL, Addr)$

$3.Broadcast(T_A, \sigma_A)$
$4.return$
$5.endprocedure$
$6.procedureVERTRANS(T_A, \sigma_A)$
$7.s \leftarrow 0$
$8.V1 \leftarrow VerID(ID_A, PK_A, K)$
$9.ifV_1 = Validthen$
$10.V2 \leftarrow Ver(T_A, \sigma A, ID_A, P_{KA})$
$11.elseAbort$
$12.ifV2 = Validthen$
$13.s \leftarrow 1$
$14.elseAbort$
$15.endif$
$16.endif$
$17.return$
$18.endprocedure$

3.4.4 Access Data

The miners in the blockchain system, once received the transaction TA, will have to verify the validation of the transaction by checking if the signature is valid, and if the message is signed with the transaction creator's public key. If the verification passes, the flag on T_A will be set to "1," and T_A will be written into new block. If a doctor's device with ID_B would like to access the data, it can create a transaction $T_B = (ID_B, ID_A||Addr)$, signs T_B with its private key SK_B and appends the signature to T_B. The miners in the system will verify firstly if T_B is validated, and secondly if the identity of ID_B belongs to the access control list. Both the two verification need to be passed in order to validate this transaction. The process is illustrated in Algorithm 4.

Algorithm 4 Access Data
Input:$ID_A||Addr, ID_B$
Output: $averifiedT_B$
1: procedure REQUESTDATA(ID$_B$, $ID_A||Addr$)
$2 : CreateT_B = (ID_B, ID_A||Addr)$
$3 : \sigma_{T_B} \leftarrow Sign(K, T_B, SK_B)$
$4 : Broadcast(T_B, \sigma_{T_B})$
$return$
$5 : endprocedure$
$6 : procedureVERTRANS(T_B; \sigma_{T_B})$
$7 : s \leftarrow 0$
$8 : V_1 VerID(ID_B, PK_B, K)$
$9 : ifV_1 = Validthen$
$10 : V_2 \leftarrow Ver(T_B, \sigma_{T_B} ID_B, PK_B)$
$11 : elseAbort$
$12 : ifV_2 = Validthen$
$13 : ifID_B \epsilon ACLthen$

$14 : s \leftarrow 1$
$15 : elseAbort$
$16 : endif$
$17 : endif$
$18 : endif$
$return$
$19 : endprocedure$

In the proposed scheme, only transactions passing access control can be written into a block. A DHT node that stores data will check if the transaction requesting data exists in the blockchain before it can send data to the requester. If an unauthorized IoT device tries to access sensitive data of a patient, it will be blocked by the blockchain as it can not pass the verification. Note that the access control list is determined by the data owner himself and nobody is able to modify it. Therefore, no entity is able to access data without the data owner's permission. Furthermore, the security of such a scheme is based on the majority of miners, which guarantees the protection of sensitive medical data.

3.5 Extension to Data Trading

With the proposed scheme, any user can easily trade his data through the blockchain. Data trading can be done in a transparent and accountable way. For example, a user wouldlike to sell his IoT data collected by electrocardiogram (ECG) sensors for years to some research institutes. This user can post a transaction through the network edge to the blockchain claiming to sell his data. An interested party can post a transaction, in which it commits a deposit to request the data. If the data owner thinks the deposit is enough, he can post a transaction with an updated access control that includes the requester. The blockchain will verify if the requester can access the data, and if yes, the deposit will be sent to the data owner

Figure illustrates how data trading is processed. Firstly, data buyer posts a transaction $Deposit_A$ with a valid signature $Sigs_{kA}(Deposit_A)$ appended to it. The transaction includes an input of the buyer's identity ID_A, the public key of ID_A, and the seller's identity ID_B. The buyer will make a commitment of d dollars, which is locked until certain situations are satisfied: 1) the data seller publishes a transaction TB that writes the buyer's identity into the access control list of requested data, then the d dollars will be sent to the seller; or 2) after t time, the seller does not sell the data, then the buyer can get his deposit back.

If the seller would like to sell his data to the buyer, he can post a transaction T_B through the edge device to update his access control list by including the buyer's identity inside the access control list for a period of t time. Then the seller is able to post a transaction GetpaidB that unlocks the d dollars in $Deposit_A$. Note that only the buyer is able to unlock the d because the blockchain will strictly check the following two requirements: 1) if T_B specifies that $ID_A \epsilon ACL_B$for t time; and 2) if $Getpaid_B$ is indeed created by the device ID_B. If the seller does not sell his data within t time, buyer is able to redeem his money by posting a transaction $GetDepoosit_A$. The blockchain will verify 1) if it is the buyer himself redeeming

the money; and 2) if t time has passed since the post of $Deposit_A$. If the verification succeeds, then $GetDepoosit_A$ will be validated and written into blocks, such that the buyer can get his deposit back.

In the above example, data trading is achieved in an effective and efficient way. The trading process is controlled and audited by the blockchain, which makes trading transparent and ac-countable. Furthermore, through this mechanism, a seller can quickly obtain trading fee from the buyer by redeeming the deposit transaction, without waiting for a period of time.

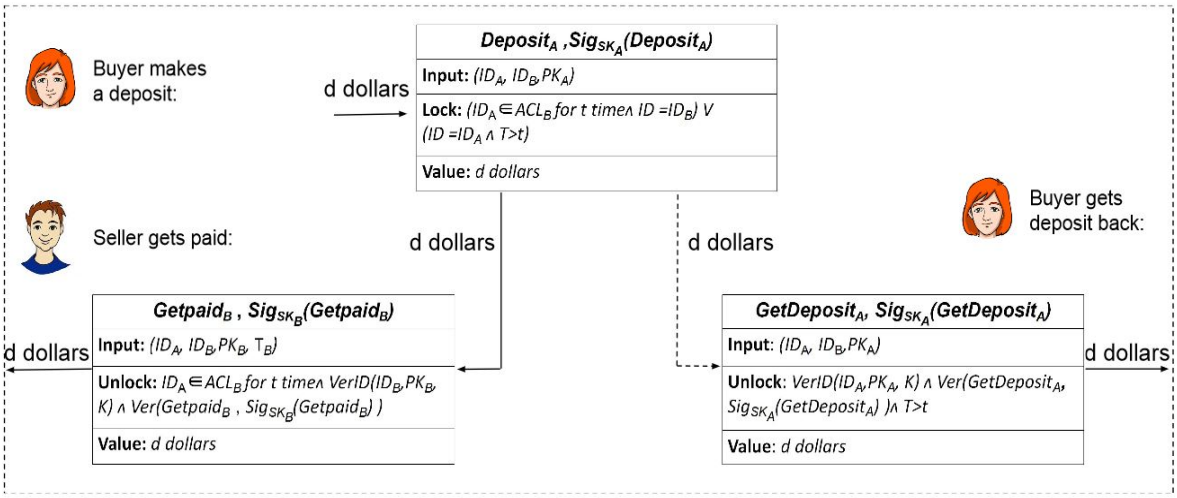

Figure 3.3: Data trading with the blockchain

3.6 Security

3.6.1 Protocol Security

Theorem 1: Algorithms 1 to 4 form a secure protocol in authentication under the assumption that the adopted certificateless cryptography algorithm is secure.

Proof The security of authentication in the proposed scheme is based on the certificateless cryptography. There exists lots of schemes that are IND-CCA secure. In such a scheme, an adversary has a negligible ”advantage” to distinguish two distinct plaintext from a ciphertext with probability 1=2+ (k), where k is a negligible function in the security parameter k.That is to say, an adversary is not able to guess the private key of a user even from large number of ciphertext in the IoT system. Therefore, it is impossible for an adversary to forge a digital signature $\sigma(T)$ for any transactions T published in the system. Furthermore, to forge a public key PK_A of a user ID_Aification function VerID.

3.6.2 Privacy

To protect sensitive IoT data, necessary encryption needs to be carried. Data sent out from an edge device to the final storage can be sent in an encrypted form such that eavesdroppers in the network cannot get sensitive data. The data can be encrypted under the IoT device's public key or under a specified entity's public key. When sharing data to another entity, the data owner could choose to encrypt the data under the entity's public key, or he could choose to use re-encryption . Re-encryption is a useful cryptography primitive that enables data encrypted under one public key to be transformed to data under another public key, without decrypting the message. Data sharing will become easy with the incorporation of re-encryption.

If the IoT data is encrypted when stored, data trading can become more complicated than the example we gave, as the buyer has to make sure that the requested data has been transformed to ciphertext under his own key before making payments. Therefore, re-encryption has to be performed by the DHT node that holds that data, instead of the data owner himself. Particularly, one more step should be added: when the data owner posts a transaction to sell the data, the DHT node re-encrypts the data and posts a transaction to the blockchain claiming that the data is now encrypted under the buyer device's public key. Only after this step, data owner is able to get payment from the buyer's deposit.

3.6.3 Traceability and Accountability

The proposed scheme brings traceability and accountability. Any IoT device accessing data in a certain DHT address will be recorded and there is no way to deny this operation. Data owner will know which entity accessed his data, and he is able to make sure none unauthorized entity has accessed his data. Also, malicious attempts to access data will be recorded and detected, which could largely mitigate Denial of Service attacks. When a malicious device constantly challenges the system to access the data, this device can be easily detected and recorded, and will be blocked permanently.

3.6.4 Blockchain Security

It is clear that the security of the proposed scheme is based on the security of blockchain and certificateless cryptography. The security of blockchain lies on the hardness of preventing sibyl attacks. In a blockchain, if an adversary is powerful enough to take over the majority of the nodes, then it can perform arbitrary malicious operations on the transactions. Therefore, whether a blockchain system is secure enough depends on if we can incentivize sufficient large number of miners in the blockchain. As discussed before, our system brings mining award from transaction fees of data storage and protection, apart from block award in a traditional blockchain cryptocurrency system. Therefore, miners will be attracted in building such a storage system for IoT applications. Further-more, a single blockchain can work for various IoT applica-tions. The more IoT applications adopt blockchain, the more transaction fee there will be, and the more miners will be attracted to the blockchain. Therefore, the blockchain platform serving IoT data storage and protection will be secure.

3.7 Implementation

The deployment of the proposed system relies heavily on the design of blockchain system. Without highly scalable blockchain system, it is hard to deploy the IoT storage system. This is the trade-off brought by eliminating the centralized server. Scalability is a major problem in blockchain's design. There are a lot of ongoing research working on it, and two most studied mechanisms to solve the scalability problem are Sharding and sidechains. To find the most scalable system that exists currently, we have to evaluate the blockchain systems that have already been deployed in very large networks. Some very new blockchain technologies such as Ripple could process thousands of transactions per second, but the design is partially decentral-ized. Decentralized designs such as Ethereum could take only thirteen transactions per second. However, there exists new-emerging blockchain design built based on Ethereum that have much better scalability. For example, Tomochain is supposed to process up to one thousand transactions per second [36]. Supposing a blockchain mechanism is adopted that could process around 1000 transactions per second, then 10; 000 transactions could be processed in roughly 10 seconds in the IoT storage system. One good thing is that, with the development of blockchain technology, it can be foreseen that more and more blockchain designs suitable for the proposed IoT storage system will come out.

Conclusion

In this paper, we propose a secure scheme for IoT data storage and protection based on blockchain. Edge computing is incorporated to help manage data storage and small IoT devices perform computations. Certificateless cryptography is adopted to set up a convenient authentication system for the blockchain-based IoT applications, and blockchain overcomes the drawback of certificatess cryptography by offering a plat-form for broadcasting the public key of a user. We give detailed algorithms on how to process transactions in such a system and how to achieve authentication and accountability. To the best of our knowledge, this is the first paper tackling the problem of building a secure and accountable storage system for large-scale IoT data, and the first to combine edge computing, certificateless cryptography, and blockchain as a whole to serve IoT applications.

Our future work lies on improving our authentication scheme for blockchain based system. In our current scheme, authentication is done by verifying the identity of the data requester. For a system with complicated access control policies, more comprehensive designs need to be explored.

REFERENCES

[1] F. Xia, L. T. Yang, L. Wang, and A. Vinel, "Internet of things," International Journal of Communication Systems, vol. 25, no. 9, p. 1101, 2012.

[2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Computer networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] R. Li, T. Song, N. Capurso, J. Yu, J. Couture, and X. Cheng, "Iot applications on secure smart shopping system," IEEE Internet of Things Journal, vol. 4, no. 6, pp. 1945–1954, 2017.

[4] T. Song, R. Li, B. Mei, J. Yu, X. Xing, and X. Cheng, "A privacy preserving communication protocol for iot applications in smart homes," IEEE Internet of Things Journal, 2017.

[5] O. Vermesan and P. Friess, Internet of things: converging technologies for smart environments and integrated ecosystems. River Publishers, 2013.

[6] (2017) Iot devices will outnumber the world's population. [Online]. Available: https://www.zdnet.com/article/iot-devices-will-outnumber-the-worlds-population-this-year-for-the-first-time/

[7] M. F. Kaashoek and D. R. Karger, "Koorde: A simple degree-optimal distributed hash table," in International Workshop on Peer-to-Peer Sys-tems. Springer, 2003, pp. 98–107.

[8] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, 2012, pp. 13–16.