# Blockchain for Large-Scale Internet of Things Data Storage and Protection

Ruinian Li, Tianyi Song, Bo Mei, Hong Li, Xiuzhen Cheng, *Fellow, IEEE*, and Limin Sun, *Member, IEEE*

**Abstract**—With the dramatically increasing deployment of IoT devices, 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 demands on the cloud servers. Meanwhile, the strong dependencies on the centralized servers bring significant trust issues. To mitigate these problems, we propose a distributed data storage scheme employing blockchain and cetrificateless cryptography. Our scheme eliminates the traditional centralized servers by leveraging the blockchain miners who perform "transaction" verifications and records audit with the help of certificateless cryptography. We present a clear definition of the transactions in a non-cryptocurrency system and illustrate how the transactions are processed. To the best of our knowledge, this is the first work designing a secure and accountable IoT storage system using blockchain. Additionally, we extend our scheme to enable data trading and elaborate how data trading can be efficiently and effectively achieved.

**Index Terms**—IoT, blockchain, certificateless cryptography

---

## 1 INTRODUCTION

INTERNET of Things (IoT) is an emerging term that describes the ubiquitous connection of everyday objects [1], [2], [3], [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 was 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 cooperatively 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, which achieves the following 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 accessing and modifying the IoT data, can be recorded by the blockchain. No malicious attempts can be made undetected.

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], [9]. 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

- *R. Li and T. Song are with the Department of Computer Science, Bowling Green State University, Bowling Green, OH 43403.*
  *E-mail: {lir, tsong}@bgsu.edu.*
- *B. Mei is with the Department of Computer Science, Texas Christian University, Fort Worth, TX 76129. E-mail: b.mei@tcu.edu.*
- *H. Li and L. Sun are with the Beijing Key Laboratory of IoT Information Security Technology, Institute of Information Engineering, Chineses Academy of Sciences, Beijing 100093, China and also with the School of Cyber Security, University of Chinese Academy of Sciences, Beijing 101408, China. E-mail: {lihong, sunlimin}@iie.ac.cn.*
- *X. Cheng is with the Department of Computer Science, The George Washington University, Washington, DC 20052. E-mail: cheng@gwu.edu.*
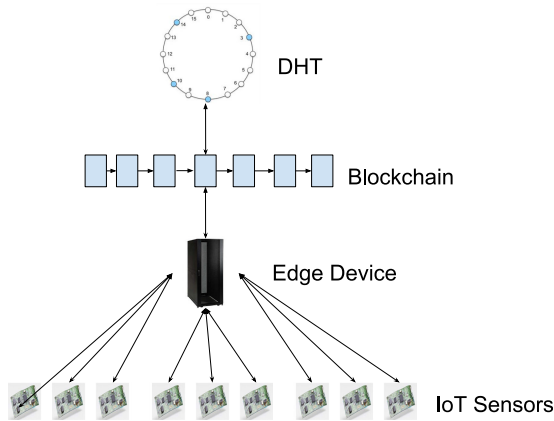
Fig. 1. The structure of data storage scheme with blockchain.

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 "transaction" 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.

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 cryptographic 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 redundancy. 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* [10], [11], [12] steps in. Certificateless 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.

Our contributions are listed as follows:

1) We propose a scheme for large-scale IoT data storage and protection. The scheme eliminates the centralized server and guarantees data protection by letting a large group of blockchain miners control the IoT data.

2) We incorporate edge computing in our proposed scheme in order to perform computations for IoT devices and forward the data to DHT for storage.

3) We propose to utilize certificateless cryptography in the blockchain-based IoT systems. The public ledger of blockchain system offers a convenient way to broadcast any IoT device's public key, which overcomes the drawback of a certificateless cryptography system. In the meantime, certificateless cryptography highly reduces redundancy brought by traditional PKI and offers an efficient way to authenticate an IoT device. To the best of our knowledge, none of previous research has tackled this problem.

The paper is organized as follows. Section 2 introduces blockchain, edge computing and the the preliminary setting of the system. Section 3 discusses how authentication are done in the proposed system. In Section 4, we extend our protocol to enable data trading. In Sections 5, we study the security issues in the system. In Section 6, deployment issues are discussed. In Section 7, the most related works are summarized. We conclude this paper with a future research discussion in Section 8.

## 2 APPLYING BLOCKCHAIN AND EDGE COMPUTING IN IoT APPLICATIONS

### 2.1 Blockchain Description

Blockchain works as a peer-to-peer network without a trusted third party. Fig. 2 illustrates a typical blockchain network. 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
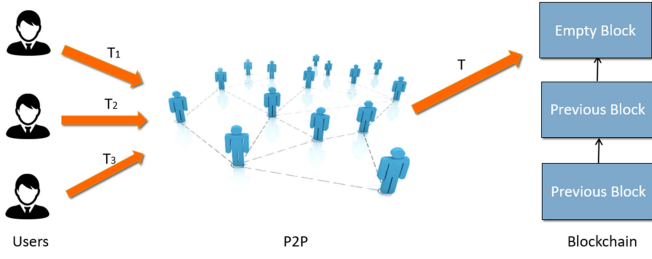
Fig. 2. Blockchain network.

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) [13], [14], [15].

The early version of blockchain, such as Bitcoin, is implemented by a clever mechanism called Proof of Work (PoW). In such a system, each block $B_n$ 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 $B_n$ such that $Hash(B_{n-1}, T, salt)$ 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 $B_n$ 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.

To tackle the power-consumption issues of the Bitcoin, other mechanisms have been explored to replace PoW such as Proof of Stake (PoS) [17], Proof of Space (PoSpace)[18], and Rem (Intel SGX) [19], 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" 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.

## 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: $T = (ID_A, Timestamp, Action = store\ data\ in\ Addr)$. 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: $T = (ID_B, ID_A, Timestamp, Action = access\ data\ in\ Addr)$. 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.

## 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 mechanism 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.

## 2.4 Edge Computing

An edge device plays a significant role in the proposed blockchain based IoT storage scheme. It not only relays messages/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 determines the DHT address to store the data and sends the encrypted data to the designated address.

## 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 AUTHENTICATION OF BLOCKCHAIN TRANSACTIONS

Establishing the asymmetric key pairs using traditional PKI [20] or IBE [21], [22], [23] 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 $SK_A$, and appends its public key $PK_A$ 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 applications. 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 $H(H(PK_i)) = Addr$ [24], [25]. This brings lots of convenience compared to traditional PKI 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 [26] 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 certificateless 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 computations of an IoT device are performed at the edge device.

## 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 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 $\lambda$ 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 $ID_A \in \{0,1\}^*$, 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 generation 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 algorithm 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 encrypt, decrypt, sign and verify are described as follows.

1) $Encrypt\ (K, M \in \mathcal{M}, ID_A, PK_A) \rightarrow C \in \mathbb{C} \vee \perp$
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) $Decrypt\ (C \in \mathbb{C}, SK_A), \rightarrow M \in \mathcal{M} \vee \perp$
This algorithm takes a ciphertext $C$ and user's private key $SK_A$, and outputs a message $M$.

3) $Sign\ (K, M \in \mathcal{M}, SK_A) \rightarrow Sig$
This algorithm takes system parameters $K$, a message $M$ and user's private key $SK_A$, and outputs a signature.

4) $Ver\ (M \in \mathcal{M}, Sig, ID_A, PK_A) \rightarrow Valid \vee Invalid \vee \perp$
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 The readers can refer to *Al-Riyami*'s scheme as an example for details [10], 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 $Ver_{PK}(Sig)$.

## 3.2 How Blockchain Transactions Work in IoT Applications

In this section, we give details how transactions are processed 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.

### 3.2.1 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 $ID_A$ 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 $PSK_A$, $X_A$ 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$(PSK_A, Sign_{SK_K}(ID_A))$
5:     $V \leftarrow Ver(ID_A, Sign_{SK_K}(ID_A), SK_K)$
6:     **if** $V = Valid$ **then**:
7:         $SK_A \leftarrow SKeyGen(K, PSK_A, X_A)$
8:         $PK_A \leftarrow PKeyGen(K, X_A)$
9:     **end if**
        return
10: **end procedure**

---

### 3.2.2 Transactions Description and Verification

After successfully 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 $T_A$, 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 $T_A$ is created from $ID_A$ 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:** $T_A, \sigma_{T_A}$
**Output:** a verified $T_A$
1: **procedure** VerTrans $(T_A, \sigma_{T_A})$
2:     $s \leftarrow 0$
3:     $V_1 \leftarrow VerID(ID_A, PK_A, K)$
4:     **if** $V_1 = Valid$ **then**
5:         $V_2 \leftarrow Ver(T_A, \sigma_{T_A}, ID_A, PK_A)$
6:     **else** Abort
7:         **if** $V_2 = Valid$ **then**
8:             $s \leftarrow 1$
9:         **else** Abort
10:        **end if**
11:    **end if**
        return
12: **end procedure**

---

### 3.2.3 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 $\sigma_A$ appended to the transaction, and publishes $T_A$ 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: **procedure** SetACL $ACL$
 2:     Create $T_A = (PK_A, ID_A, ACL, Addr)$
 3:     Broadcast $(T_A, \sigma_A)$
        return
 4: **end procedure**
 5: **procedure** Vertrans $T_A, \sigma_A$          ▷ run by the miners
 6:     $s \leftarrow 0$
 7:     $V_1 \leftarrow VerID(ID_A, PK_A, K)$
 8:     **if** $V_1 = Valid$ **then**
 9:         $V_2 \leftarrow Ver(T_A, \sigma_A, ID_A, PK_A)$
10:     **else** Abort
11:         **if** $V_2 = Valid$ **then**
12:             $s \leftarrow 1$
13:         **else** Abort
14:         **end if**
15:     **end if**
         return
16: **end procedure**

---

The miners in the blockchain system, once received the transaction $T_A$, 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.

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.

---

**Algorithm 4.** Access Data

---

**Input:** $ID_A \| Addr, ID_B$
**Output:** a verified $T_B$
 1: **procedure** RequestData $(ID_B, ID_A \| Addr)$
 2:     Create $T_B = (ID_B, ID_A \| Addr)$
 3:     $\sigma_{T_B} \leftarrow Sign(K, T_B, Sk_B)$
 4:     Broadcast $(T_B, \sigma_{T_B})$
        return
 5: **end procedure**
 6: **procedure** VerTrans $(T_B, \sigma_{T_B})$          ▷ run by the miners
 7:     $s \leftarrow 0$
 8:     $V_1 \leftarrow VerID(ID_B, PK_B, K)$
 9:     **if** $V_1 = Valid$ **then**
10:         $V_2 \leftarrow Ver(T_B, \sigma_{T_B}, ID_B, PK_B)$
11:     **else** Abort
12:         **if** $V_2 = Valid$ **then**
13:             **if** $ID_B \in ACL$ **then**
14:                 $s \leftarrow 1$
15:             **else** Abort
16:             **end if**
17:         **end if**
18:     **end if**
         return
19: **end procedure**

---

## 4 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 would like 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.

Fig. 3 illustrates how data trading is processed. Firstly, data buyer posts a transaction $Deposit_A$ with a valid signature $Sig_{sk_A}(Deposit_A)$ 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 $T_B$ 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 $Getpaid_B$ 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 $ID_A \in ACL_B$ 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
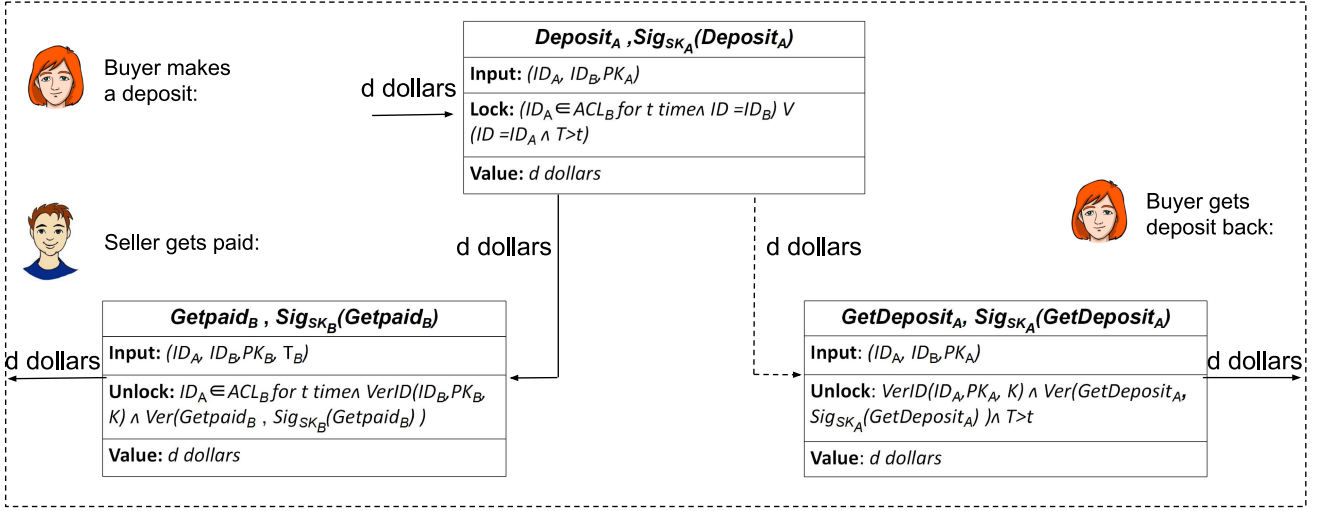
Fig. 3. Data trading with the blockchain.

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 accountable. 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.

## 5   SECURITY

### 5.1   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 $1/2 + \epsilon(k)$, 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 $ID_A$ is also not possible, as the fake key cannot pass the verification function $VerID$.                                      □

### 5.2   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 [27]. 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.

### 5.3   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.

### 5.4   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. Furthermore, a single blockchain can work for various IoT applications. 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.

## 6 DISCUSSIONS

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 [14], [15], [28], [29], [30], and two most studied mechanisms to solve the scalability problem are Sharding [31], [32] and sidechains [33]. 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 [34] could process thousands of transactions per second, but the design is partially decentralized. Decentralized designs such as Ethereum [35] 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.

## 7 RELATED WORK

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 [37]. For data storage, most researchers focus on building databased management models to mitigate the massive IoT data storage problem. NoSQL [38], [39] and Hadoop [40] databases are attracting most of the attention. Besides, the authors in [41] 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 [42]. Edge computing and IoT have been extensively studied [8], [43], [44]. The visions and challenges in edge computing are comprehensively studied in [45], and the security and privacy issues in edge computing are extensively discussed in [46].

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 [14], [15], [29] or reducing the power consumption of a PoW [17], [19]. There are some research utilizing blockchain on practical applications. In 2014, for the first time, Bitcoin is proposed to form a lottery protocol as as a multiparty computation method [47]. The authors demonstrated 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 [49]. The most related work is [50], in which Zyskind et al. proposed a decentralized personal data management system using blockchain and illustrated how blockchains could become a vital resource in trusted computing [50]. 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 [10] 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 [51]. The most Efficient Certificateless signature scheme to date was proposed that involve only simple hash operations for signing [52]. 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 [53]. 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.

## 8 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 platform 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," *Int. J. Commun. Syst.*, vol. 25, no. 9, 2012, Art. no. 1101.

[2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, 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 Things J.*, 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*, vol. 4, no. 6, pp. 1844–1852, Dec. 2017.

[5] O. Vermesan and P. Friess, *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*. Gistrup, Denmark: River Publishers, 2013.

[6] Iot devices will outnumber the world's population, 2017. [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 *Proc. Int. Workshop Peer-to-Peer Syst.*, 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 *Proc. 1st Ed. MCC Workshop Mobile Cloud Comput.*, 2012, pp. 13–16.

[9] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, 2014, pp. 1–8.

[10] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Asiacrypt*, vol. 2894. New York, NY, USA: Springer, 2003, pp. 452–473.

[11] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "On the security of certificateless signature schemes from asiacrypt 2003," in *Proc. 4th Int. Conf. Cryptology Netw. Secur.*, 2005, pp. 13–25.

[12] S. S. Chow, C. Boyd, and J. M. G. Nieto, "Security-mediated certificateless cryptography," in *Public key cryptography*, vol. 3958. New York, NY, USA: Springer, 2006, pp. 508–524.

[13] E. Buchman, "Tendermint: Byzantine fault tolerance in the age of blockchains," Ph.D. dissertation, University of Guelph, Guelph, ON, Canada, 2016.

[14] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work versus. bft replication," in *Proc. Int. Workshop Open Problems Netw. Secur.*, 2015, pp. 112–125.

[15] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-ng: A scalable blockchain protocol," in *Proc. 13th Usenix Conf. Netw. Syst. Des. Implementation*, 2016, pp. 45–59.

[16] Botcoin wiki scalability, 2017. [Online]. Available: https://en.bitcoin.it/wiki/Scalability

[17] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Proc. Annu. Int. Cryptology Conf.*, 2017, pp. 357–388.

[18] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak, "Proofs of space," in *Proc. Annu. Cryptology Conf.*, 2015, pp. 585–605.

[19] F. Zhang, I. Eyal, R. Escriva, A. Juels, and R. van Renesse, "Rem: Resource-efficient mining for blockchains," *IACR Cryptology ePrint Archive*, vol. 2017, 2017, Art. no. 179.

[20] D. R. Kuhn, V. C. Hu, W. T. Polk, and S.-J. Chang, "Introduction to public key technology and the federal pki infrastructure," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. SP 800–32, 2001.

[21] S. Chatterjee and P. Sarkar, *Identity-Based Encryption*. New York, NY, USA: Springer Science & Business Media, 2011.

[22] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Proc. Annu. Int. Cryptology Conf.*, 2001, pp. 213–229.

[23] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Adv Cryptology*, 2005, pp. 457–473.

[24] S. Goldfeder, R. Gennaro, H. Kalodner, J. Bonneau, J. A. Kroll, E. W. Felten, and A. Narayanan, "Securing bitcoin wallets via a new dsa/ecdsa threshold signature scheme," 2015, http://stevengoldfeder.com/papers/threshold_sigs.pdf

[25] R. Gennaro, S. Goldfeder, and A. Narayanan, "Threshold-optimal dsa/ecdsa signatures and an application to bitcoin wallet security," in *Proc. Int. Conf. Appl. Cryptography Netw. Secur.*, 2016, pp. 156–174.

[26] C. Rackoff and D. R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," in *Proc. Annu. Int. Cryptology Conf.*, 1991, pp. 433–444.

[27] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[28] J. Herrera-Joancomartí and C. Pérez-Solà, "Privacy in bitcoin transactions: new challenges from blockchain scalability solutions," in *Proc. Modeling Decisions Artif. Intell.*, 2016, pp. 26–44.

[29] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, et al., "On scaling decentralized blockchains," in *Proc. Int. Conf. Financial Cryptography Data Secur.*, 2016, pp. 106–125.

[30] G. Karame, "On the security and scalability of bitcoin's blockchain," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1861–1862.

[31] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 17–30.

[32] A. E. Gencer, R. van Renesse, and E. G. Sirer, "Short paper: Service-oriented sharding for blockchains," in *Proc. Int. Conf. Financial Cryptography Data Secur.*, 2017, pp. 393–401.

[33] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, and P. Wuille, "Enabling blockchain innovations with pegged sidechains," 2014. [Online]. Available: http://www. opensciencereview.com/papers/123/enablingblockchain-innovations-with-pegged-sidechains

[34] L. Lee, "New kids on the blockchain: How bitcoin's technology could reinvent the stock market," *Hastings Bus. LJ*, vol. 12, 2015, Art. no. 12.

[35] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, 2014.

[36] Tomochain, 2018. [Online]. Available: https://tomochain.com/

[37] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 171–209, 2014.

[38] T. Li, Y. Liu, Y. Tian, S. Shen, and W. Mao, "A storage solution for massive iot data based on nosql," in *Proc. IEEE Int. Conf. Green Comput. Commun.*, 2012, pp. 50–57.

[39] J. Guo, L. Da Xu, G. Xiao, and Z. Gong, "Improving multilingual semantic interoperation in cross-organizational enterprise systems through concept disambiguation," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 647–658, Aug. 2012.

[40] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Proc. IEEE 26th Symp. Mass Storage Syst. Technol.*, 2010, pp. 1–10.

[41] L. Jiang, L. Da Xu, H. Cai, Z. Jiang, F. Bu, and B. Xu, "An IoT-oriented data storage framework in cloud computing platform," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1443–1451, May 2014.

[42] W. Shi and S. Dustdar, "The promise of edge computing," *Comput.*, vol. 49, no. 5, pp. 78–81, 2016.

[43] M. Satyanarayanan, "The emergence of edge computing," *Comput.*, vol. 50, no. 1, pp. 30–39, 2017.

[44] P. Corcoran and S. K. Datta, "Mobile-edge computing and the internet of things for consumers: Extending cloud computing and services to the edge of the network," *IEEE Consumer Electron. Mag.*, vol. 5, no. 4, pp. 73–74, Oct. 2016.

[45] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[46] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Comput. Syst.*, vol. 78, pp. 680–698, 2018.

[47] M. Andrychowicz, S. Dziembowski, D. Malinowski, and L. Mazurek, "Secure multiparty computations on bitcoin," in *Proc. IEEE Symp. Secur. Privacy*, 2014, pp. 443–458.

[48] Y. He, H. Li, X. Cheng, Y. Liu, and L. Sun, "A bitcoin based incentive mechanism for distributed p2p applications," in *Proc. Int. Conf. Wireless Algorithms Syst. Appl.*, 2017, pp. 457–468.

[49] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, May 2016.

[50] G. Zyskind, O. Nathan, et al., "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Secur. Privacy Workshops*, 2015, pp. 180–184.

[51] S.-H. Seo, M. Nabeel, X. Ding, and E. Bertino, "An efficient certificateless cryptography scheme without pairing," in *Proc. 3rd ACM Conf. Data Appl. Secur. Privacy*, 2013, pp. 181–184.

[52] L. Zhang, J. Liu, H. Wu, and R. Sun, "An efficient and robust certificateless short signature scheme," in *Proc. Inf. Netw. Secur.*, 2013, pp. 1–7.

[53] F. Li, Y. Han, and C. Jin, "Certificateless online/offline signcryption for the internet of things," *Wireless Netw.*, vol. 23, no. 1, pp. 145–158, 2017.

**Ruinian Li** received the PhD degree in computer science from the George Washington University, in 2018. He is currently an assistant professor with the Department of Computer Science, Bowling Green State University. His research interests include Internet of Things (IoT), network security, applied cryptography, privacy-aware computing, and blockchain technology.
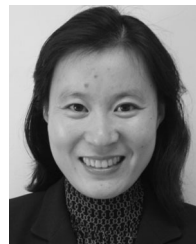
**Tianyi Song** received the PhD degree in computer science from the George Washington University, in 2017. She is a full-time faculty with the Department of Computer Science, Bowling Green State University. Her current research interests include secure and privacy-aware computing for the Internet of Things (IoT) and cyber physical systems (CPS), mobile computing, wireless networking, and blockchain technology.
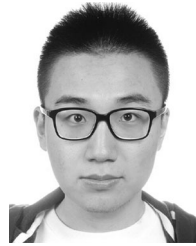
**Bo Mei** received the PhD degree in computer science from the George Washington University, in 2018. He is currently an assistant professor with the Department of Computer Science, Texas Christian University. He began his research focusing on mobile computing since 2014, and has conducted extensive study on system applications for IoT devices. Currently, his research spans the broad area of mobile computing, IoT, and social network inference attacks.

**Hong Li** received the BA degree from the Xi'an Jiaotong University, and the PhD degree from the University of Chinese Academy of Sciences. He is an assistant professor with the Institute of Information Engineering, Chinese Academy of Sciences. His primary research interests include IoT security, privacy-aware computing and blockchain.

**Xiuzhen Cheng** received the MS and PhD degrees in computer science from the University of Minnesota Twin Cities, in 2000 and 2002, respectively. She is a professor with the Department of Computer Science, The George Washington University, Washington, DC. Her current research interests include privacy-aware computing, wireless and mobile security, cyber physical systems, mobile computing, and algorithm design and analysis. She has served on the editorial boards of several technical journals and the technical program committees of various professional conferences/workshops. She also has chaired several international conferences. She worked as a program director for the US National Science Foundation (NSF) from April to October in 2006 (full time), and from April 2008 to May 2010 (part time). She received the NSF CAREER Award in 2004. She is a member of the ACM, and a fellow of the IEEE.

**Limin Sun** received the BS, MS, and PhD degrees from the College of Computer, National University of Defense Technology, in 1988, 1995, and 1998, respectively. Currently, he is working as a professor with the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include IoT security and privacy, ICS security, wireless sensor networks, and internet of things. He is also a member of the IEEE and a senior member of the China Computer Federation (CCF).