Fine-grained Access Control Mechanism of Industrial Internet of Things Based on DAG Blockchain

Fei Tang[1,2], Zhangtao Ye[1], Kung Dong[1], and Dong Huang[3]
(Corresponding author: Zhangtao Ye)

College of Computer Science and Technology & Chongqing University of Posts and Telecommunications[1]
Chongqing 400065, China
Email: S190201057@stu.cqupt.edu.cn
School of Cyber Security and Information Law & Chongqing University of Posts and Telecommunications[2]
Key Laboratory of Advanced Manufacturing Technology of the Ministry of Education & Guizhou University[3]
Guizhou 402760, China

(Received Dec. 11, 2021; Revised and Accepted June 14, 2022; First Online July 3, 2022)

Abstract

Numerous researchers from academia and industry exploit the feasibility of the blockchains technique to achieve security in the industrial internet of things applications. Blockchain can provide excellent characteristics like data integrity and non-tamperability, but industrial internet of things applications are still confronted with security issues like the certification of equipment, confidentiality of the data, and access control. Moreover, the limited throughput of blockchain can be one of the bottlenecks of industrial internet of things systems. To settle these problems, we provide a novel industrial internet of things system based on the directed-acyclic-graph blockchain architecture to settle down the above problems. This system adopts identity-based signature and attribute-based encryption techniques to authenticate devices and encrypt data. As a result, industrial internet of things equipment can be certified, and encrypted data can be kept confidential even if the storage node is not the data owner. We also achieve fine-grained access control in the industrial internet of things applications. We conduct a security and privacy analysis and a performance evaluation. We prove our scheme is safe from industrial internet of things applications, and throughput is not excessively low in practical usage.

Keywords: Attribute Based Encryption; Blockchain; Fine-Grained Access Control; Industrial Internet of Things

1 Introduction

The concept of industry 4.0 was first proposed by the German government in 2011 at the Hanover industrial exposition [26]. The basic idea of the Industry 4.0 is built on the integration of information and communication technologies with industrial technology. The construction of the digital and intelligent factory is indispensable with Cyber-Physical System (CPS), which can promote manufacturing to become more digital, information-led, customized, and green [35]. The IIoT will create a variety of information from a variety of resource [1]. As the industrial internet of things (IIoT) develops further, various countries and governments have formulated standards and development plans for the IIoT due to economic benefits [31]. Although IIoT promises significantly advanced in broad application scenarios, there are still several security issues remain unsettled. For instance, data modification, single-node errors, etc.

IIoT equipment is usually exposed to the problem of low battery and low storage capacity [32]. In current IIoT applications, different systems are interacting with the physical world. Systems can be vulnerable when the systems receive data from multiple intermediaries, requiring multilevel security approaches, in addition to link encryption [29]. Specifically, IIoT refers to all interconnected sensors, instruments, and other devices, which in combination with industrial applications [20]. Equipment in IIoT faces plenty of attacks in actual application scenes, such as reverse engineering, injecting crafted packets or input, and brute-force search attacks etc [22]. IIoT applications require massive data sharing operations. The process happens between senders and receivers brings on the security issues of access control. In the actual application scenarios , it is vital to decide which entities are allowed to access and which are not. Malicious entities access the systems will compromise the security of the IIoT systems and ruin the integrity and effectiveness of data.

According to [12], current IIoT systems within partially structured smart factories play a central role in monitoring and supervising natural processes by taking autonomous and decentralized decisions to maximize the production process. That means all the sensitive information is fully exposed to the owner of servers instead of the owner of data. The healthy operation in the systems depends on the conscience and integrity of the owners of servers. For data owners, their ownership of data is incomplete. That is unacceptable for some IIoT scenarios that need strict confidentiality. Besides, centralized IIoT architecture relies on convergent servers to handle the data computation and storage, which may hit a bottleneck as the number of IIoT users explodes.

Our contributions are:

1) We integrate the DAG network into the IIoT application architecture and implement the role division. The clients of the DAG network consist of two types of nodes, which are full nodes and light nodes. The light nodes in the proposed IIoT architecture are responsible for collecting information and package transaction, the full nodes in the proposed IIoT application architecture are responsible for node consensus, data storage, authority authentication, and routing.

2) We implement a fine-grained access control scheme. The equipment in the IIoT scenario generally lacks a security guarantee, and the data is easy to leak. So we integrate an efficient IBS mechanism to achieve equipment certification and a distributed ABE mechanism to ensure data security and fine-grained access control in the IIoT scenario.

3) We design a new DAG consensus algorithm. In this algorithm, we adopt Lamport timestamp algorithm and PBFT consensus algorithm. We implement this consensus algorithm and analyzed the efficiency of this algorithm.

4) Based on the existing schemes, we analyze the security of the two algorithms in this scheme and implement a performance test.

2 Related Works

In 2008 Satoshi Nakamoto invented the basis for blockchain-based distributed ledgers [21]. This unique technology is fully decentralized and provided several excellent characteristics like non-tamperability and undeniable, which make blockchain technology quite suitable for IIoT scenarios. But the throughput of the current blockchain system is seriously limited which would affect the performance of the system. The tremendous waste of the current blockchain is unsuitable for the real industrial production environment either [19].

A brand new blockchain architecture named directed-acyclic-graph (DAG) appeared in 2015, which the initial version of DAG-based blockchain was called DAG-coin [14]. It is the first proposed concept of the DAG-based blockchain, though it is just a prototype. Based on this concept, there is an application named IOTA [24]. Afterward, the academia proposed several DAG-based blockchains [6, 7, 11, 16, 27]. The biggest distinguish between DAG-based blockchain and the original blockchain is the underlying data structure is no longer a chain, but a graph instead. Besides, the consensus algorithm in DAG is quite different from the original blockchain. The intention of the creation of the DAG-based blockchain is scalability and efficiency. The biggest feature of this kind of blockchain is allowing forks, greater throughput, and better concurrent processing capabilities than the original blockchain. These characteristics are suitable for the unpredictable information interaction and the huge amount of information in the IIoT scenario.

Many papers and studies are focusing on data security on the IIoT based on blockchain. For example, Wan *et al.* [30] proposed a layered architecture of IIoT integrated blockchain. This architecture is divided into five layers, namely, application layer, storage layer, gateway layer, firmware layer, and perception layer. The equipment need not do a lot of calculations work by connecting itself to a microchip or a microcomputer. In addition, the article also considers access control issues in the data interaction process between equipment. As for the access control structure, the paper divides all roles in the system according to the BLP model and Biba model. Besides that, this paper creates a mechanism similar to a whitelist for nodes to achieve access control functions. However, the whitelist mechanism cannot achieve fine-grained access control and aggravates the system to a certain extent after the system scale is expanded.

Tang *et al.* [28] proposed a new electronic health records(EHRs) paradigm which can help in dealing with the centralized problem of cloudbased EHRs. The paper presented an authentication scheme for blockchain-based EHRs with IBS using multiple authorities. Accordiong to the paper, the scheme can resist collusion attack out of N from $N-1$ authorities. The scheme is also provably secure in the random oracle model and has more efficient signing and verification algorithms than existing authentication schemes of blockchain-based EHRs.

Zhang *et al.* [34] deliver a blockchain architecture using ABS and CP-ABE for data sharing in the Internet of Things(IoT) is proposed. The ABS algorithm is used in the system to authenticate the equipment to solve the equipment security problem. The information collected by the IIoT equipment should be uploaded to the cloud for processing to enhance decision-making and facilitate business activities. While the cloud is untrustworthy, the system uses the fine-grained access control mechanism of the ABE algorithm to control the information interaction in the system. To solve the performance problem of the blockchain, the system uses the PBFT algorithm instead of POW.

Cui *et al.* [8] proposed a DAG architecture worked on the IIoT. In this architecture, the client role is divided into miners, gateways, and nodes. The miner is a full-node client with strong computing power, and the gateway is a light-node client with large bandwidth capability. The node is a light node client and needs to initiate transactions to the gateway. However, this paper does not consider the security issues on the IIoT, the lack of computing power and storage capacity of the IIoT equipment, or divide the IIoT equipment into entities.

3 Preliminaries

3.1 Composite Order Bilinear Pairings

The scheme construct ABE mechanism using composite order bilinear groups, which were first introduced in [3]. Let us define a group generator $\mathcal{G}$, an algorithm which takes a security parameter λ as input and outputs a description of a bilinear group G. For our purposes, we will have $\mathcal{G}$ output $(p_1, p_2, p_3, G, G_T, e)$ where p_1, p_2, p_3 are distinct primesm, G and G_T are cyclic groups of order $N = p_1p_2p_3$, and $e : G^2 \rightarrow G_T$ is a map such that:

1) (Bilinear) $\forall g, h \in G,\ a, b \in \mathbb{Z}_N, e(g^a, g^b) = e(g, h)^{ab}$;

2) (Non-degenerate) $\exists g \in G$ such that $e(g, g)$ has order n in G_T.

We assume that the group operations in G and G_T as well as the bilinear map e are computable in polynomial time with respect to λ and that the group description of G and G_T include generators of the respective cyclic groups. We let G_{p_1}, G_{p_2}andG_{p_3} denote the subgroup of orderp_1, p_2 and p_3 in G respectively. We note that when $h_i \in G_{p_i}$ and $h_j \in G_{p_j}$ for $i \neq j$, $e(h_i, h_j)$ is the identity element in G_T. To show this, suppose $h_1 \in G_{p_1}$ and $h_2 \in G_{p_2}$. Let g denote a generator of G. Then, $g^{p_1p_2}$ generates G_{p_3}, $g^{p_1p_3}$ generates G_{p_2}, $g^{p_2p_3}$ generates G_{p_1}. Hence, for some $\alpha_1, \alpha_2, h_1 = (g^{p_2p_3})^{\alpha_1}$ and $h_2 = (g^{p_1p_3})^{\alpha_2}$. Then:

$$\begin{aligned} e(h_1, h_2) &= e(g^{p_2p_3\alpha_1}, g^{p_1p_3\alpha_2}) \\ &= e(g^{\alpha_1}, g^{p_3\alpha_2})^{p_1p_2p_3} \\ &= 1 \end{aligned}$$

3.2 Lagrange Interpolation

Shamirs secret sharing uses the Lagrange interpolation technique to obtain the secret from shared-secrets. Suppose that $p(x) \in \mathbb{Z}_{|}[x]$ is a $(k-1)$ degree polynomial and secret $s = p(0)$. Denote $S = x_1, x_2, \cdots, x_k$ and the Lagrange coefficient for x_i in S as

$$\Delta_{x_i, S(x)} = \prod_{x_j \in S, x_j \neq x_i} \frac{x - x_j}{x_i - x_j} \tag{1}$$

For a given k different number of values $p(x_1), p(x_2), \cdots, p(x_k)$, the polynomial $p(x)$ can be reconstructed as follows:

$$p(x) = \sum_{x_i \in S} p(x_i) \prod_{x_j \in S, x_j \neq x_i} \frac{x - x_j}{x_i - x_j} = \sum_{x_i \in S} p(x_i)\Delta_{x_i, S(x)} \tag{2}$$

hence the secret s can be obtained as:

$$s = p(0) = \sum_{x_i \in S} p(x_i) \prod_{x_j \in S, x_j \neq x_i} \frac{0 - x_i}{x_i - x_j} \tag{3}$$

3.3 Linear Secret-Sharing Schemes

The ABE mechanism in this paper adopts linear secret-sharing schemes (LSSS). The scheme use the defintion from [2]

A secret sharing scheme Π over a set of parties $\mathcal{P}$ is linear (over $\mathbb{Z}_P$) has following properties:

1) The shares for each party form a vector over $\mathbb{Z}_P$;

2) There is a matrix A called share-generating matrix for Π. The matrix has l rows and n cols. For all $x = 1, \ldots, l$, the x^{th} row of A is labeled by a party $\rho(x)$(ρ is function from $\{1, \ldots, l\}$to $\mathcal{P}$). Consider the column vector $v = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_P$ is the secret to be shared and $r_2, \ldots, r_n \in \mathbb{Z}_P$ are randomly chosen, the Av is the vector of l shares of the secret s according to Π. The share $(Av)_x$ belong to party $\rho(x)$.

4 Layered IIoT Architectrue Based on DAG

4.1 Layered IIoT Architectrue

In this paper, the IIoT model is based on decentralized architecture. Certain equipment is organized into an entity according to subordination. All entities and DAG networks together constitute the IIoT system. The system architecture is divided into four layers: perception layer, storage layer, gateway layer, and application layer.

In the perception layer, the composition is diverse. There are many digital and nondigital devices that can act as the equipment in the IIoT system, such as a robotic arm, a computer, or a monitor. The devices are responsible for transfering the data to the system, so it must connect to a client in the DAG network. Both the devices and clients constitute the perception layer. The data in the system need to be persistent to support applications. In other words, the data are supposed to store in a database. All full nodes in the DAG network constitute the storage layer. In this architectrue, the IIoT system is divided into the various entity. Access control issues arise when interaction occurs between different entities. Entities need an identity and access control center which can be represented by the gateway. All gateways form the gateway layer. The last layer of the system is the application layer. After the system collects information, initiates transactions, and forms a consensus, it can develop IIoT applications on this basis, such as energy, manufacturing, automotive, and smart cities (see in Figure 1).

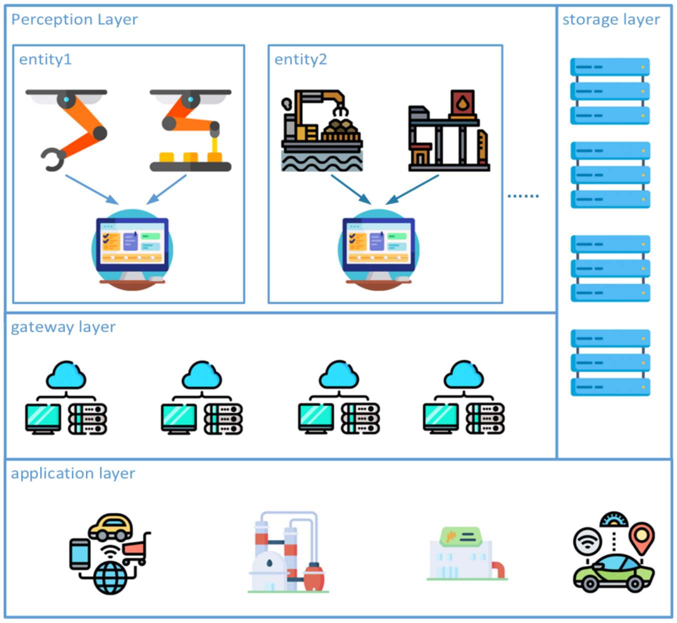

Figure 1: Layered IIoT architectrue

4.2 DAG Network Architectrue

Generally, the original blockchain has low throughput, excessive resource consumption. In addition, the original blockchain can not meet the demand of unpredictability and a huge amount of equipment messages in the process of transporting information in the application scene. In original blockchain systems, all users must solve mathematic hard problems to compete for the opportunity to generate new blocks. Only the user who wins the competition whose calculation is effective. The calculations of users who lost the competition are wasted. The original blockchain can not tolerate fork, the fork due to high concurrency will eventually be eliminated after some time. In IIoT applications, data interaction is often linked to high concurrency. If the blockchain system can not tolerate a fork, this certainly causes a waste of resources and has a bad effect on performance.

In the above chapter, we introduce a brand-new blockchain system which names DAG blockchain. This blockchain system processes properties like fork tolerance, high concurrent processing capability, and higher throughput. The topological of DAG-based blockchain in the system is directed-acyclic-graph(DAG). The difference from the DAG-based blockchain is each block can connect one or more parent blocks. For each node in the system, a chain of its own is generated, and in the subsequent process of the system, a DAG is formed by linking the chain with other nodes through a consensus algorithm. The nodes in the system are connected through the p2p network. After the nodes reach a consensus, a new block is generated. The nodes are going to package transactions from the transaction pool and broadcast the new block to the P2P network.

The entire model is divided into four layers, each of layer is responsible for its corresponding function. In the system, to better describe the operation process, the roles are divided into the following five types from the system level. They are equipment client, storage layer client, gateway client, user client, and entity. According to the role division of Dag network, the roles in the system are divided into the following two types, full nodes, and light nodes. The equipment client refers to the client directly connected to one or more IIoT equipment. The roles of these clients in the blockchain system architecture belong to light nodes. The client does not participate in the system consensus process and does not store all Block. The client is only responsible for the verification and storage related to its transactions. In other words, the clients need not store all blocks, they only need to store all block headers so that the client can verify related transactions. Storage layer clients refer to clients with strong storage capabilities. The roles of these clients in the blockchain system architecture belong to full nodes. These clients participate in the whole process of the consensus algorithm and store all blocks. All blocks are stored in these clients in a distributed manner, and subsequent query history transactions can be obtained from these clients.

Gateway clients refer to clients with strong network bandwidth capabilities . The roles of these clients on the blockchain system belong to light nodes. These clients are not going to participate in the entire process of the consensus algorithm, either store all blocks. Each gateway client corresponds to an entity, and these clients are also responsible for attribute registration, identity registration, and message broadcasting in the system. The user client refers to the client that is not connected to the equipment in the DAG network, and it also belongs to the light node. These clients are mainly responsible for the query of historical transactions, etc., which is convenient for the development of application layer IIoT applications. Entities do not belong to the role division of the blockchain architecture. In the IIoT application scene, most of the equipment is divided into the same organization in space and time. In this model, this organization is called an entity. In the entity, there are multiple devices, light node clients, full node clients, etc. (see in Figure 2).

4.3 DAG Consensus Algorithm

One of the biggest differences between the DAG blockchain and the bitcoin-like blockchain is their consensus algorithm. The consensus result of all nodes in the bitcoin-like blockchain is the longest chain and the transactions in this longest chain. The consensus result of all nodes in the DAG blockchain is not the longest chain, but a linear sequence of all blocks of the system and the transactions therein.

The DAG of this scheme is in the form of a parallel

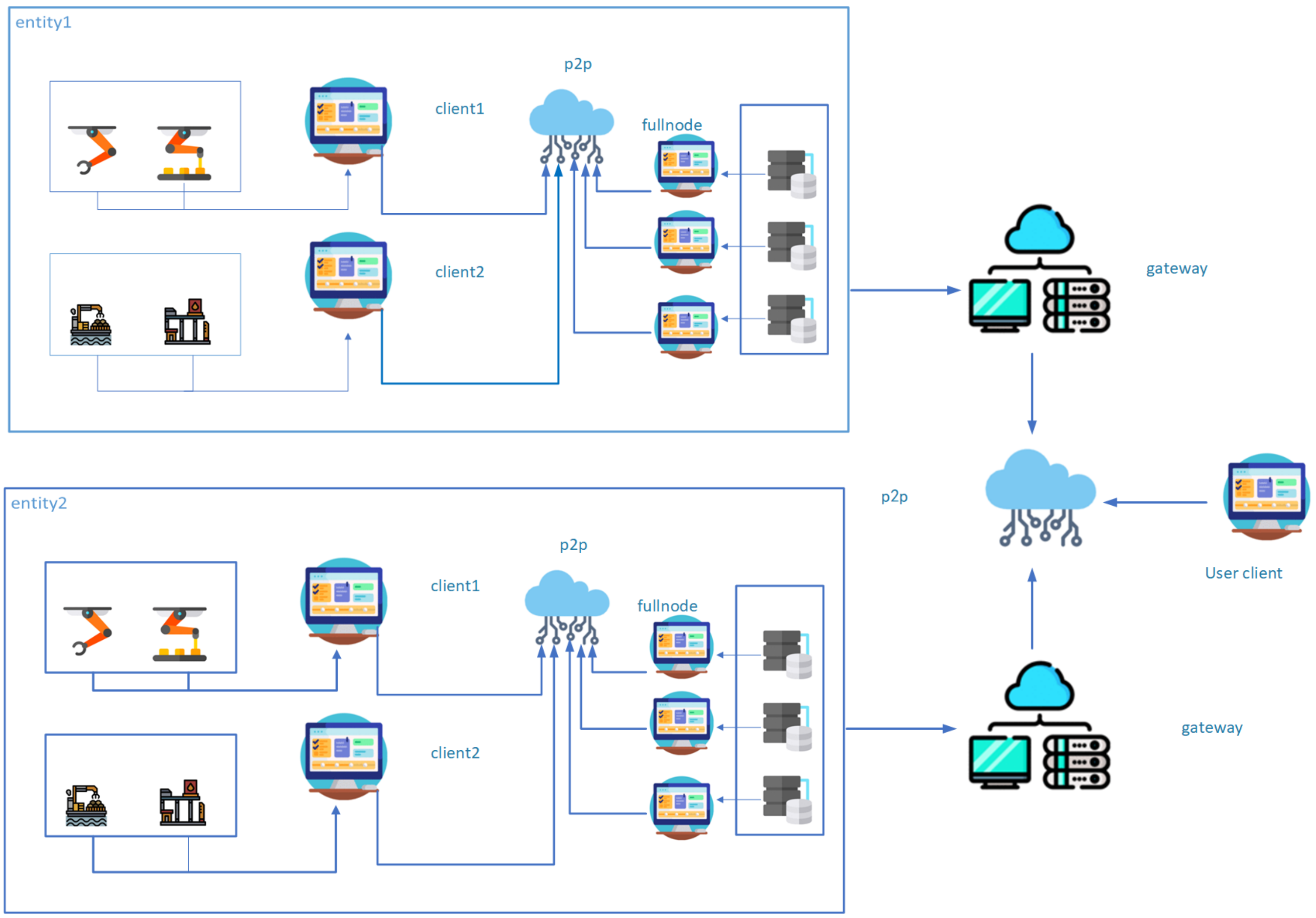

Figure 2: DAG network architectrue

chain. All full nodes in the DAG will have a chain of their own, and each node generating a new block will run a random selection algorithm to choose k nodes and connect to the last block of these k nodes.

All full nodes in the system need to run the consensus algorithm and eventually all nodes reach consensus on the topology graph of the entire DAG. The main procedure of the algorithm is as follows:

Algorithm 1 show the pseudo of the core procedure of the DAG consensus algorithm of the scheme. Algorithm 1 has two parts, in first part each block requests synchronization, creates new block and order the blocks. In Line 2, the node generates a genius block, which does not contain information and acts as an identifier for the node. In Line 4, a node run Algorithm 2 and get several nodes. Then in Lines 5 and 6, the node will synchronize with the node obtained from Algorithm 2. In Lines 7 and 8, the node create a new block and broadcast out the message. In Line 9, the node and nodes obtained from Algorithm 2 update the timestamp by Algorithm 3. In Line 10, the node connect the new block with the last blocks of nodes obtained from Algorithm 2. In Line 11, the node run Algorithm 4 to sort conflict blocks. In second part is the respond to synchronization requests. Both two parts runs in parallel.

Algorithm 1 Main procedure

```
 1: procedure MAIN PROCEDURE
 2:     each node create a genius block
 3: loop:
 4:     node A run Algorithm 2 get node A,B
 5:     node A sends sync requests to nodes B, C
 6:     nodes A, B, C perform sync operations
 7:     new block created
 8:     broadcast the message by p2p network
 9:     update the timestamp by Algorithm 3
10:     connect new block with the last block of node B,C
11:     run Algorithm 4 to sort conflict blocks
12: loop:
13:     request sync from a node
14:     perform sync operations
15: end procedure
```

In this scheme, nodes need to select some nodes for synchronization in each round. The synchronization process needs to choose nodes with short communication time as much as possible, which can reduce the communication load of the system. But if only nodes with short communication time are considered for selection, then the selected nodes for synchronization are fixed. The topol-

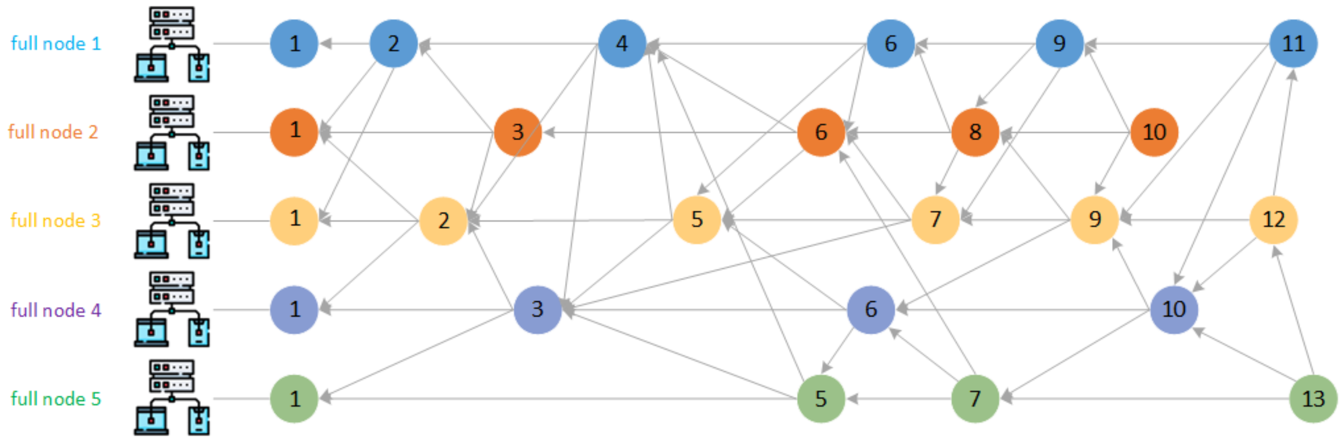

Figure 3: An example of lamport timestamp

Algorithm 2 Node selection algorithm

```
Require: k
Ensure: k nodes
for each node in node poll do
    compute $p = c + 0.2t$
end for
select the k nodes with the highest p
update t of these k nodes
return k nodes with highest p
```

Algorithm 3 Lamport timestamp procedure

```
procedure LAMPORT TIMESTAMP PROCEDURE
    initial timestamp of genius block with value one
loop:
    increments its timestamp before creating a block, and put timestamp in the new block
    send message with timestamp
    compare timestamp when receiving message
    update timestamp with bigger one when receiving message
end procedure
```

ogy diagram of the system DAG will then become several separated diagrams. Therefore, this scheme considers the communication time and the number of synchronization times at the same time, ensuring that nodes with long communication time will also be considered. The input of Algorithm 2 is the number k, the output of Algorithm 2 is k selected nodes for synchronization. The value p represents the priority of synchronization, which is calculated as p = c + 0.2t. The value c is the communication time of all nodes, the value t represents the number of synchronization times. Select the k nodes with the highest p after the calculation is completed, then update the t of these k nodes. At last, the algorithm ouput the selected k nodes.

The form of DAG in this scheme is parallel chain, each node has its own chain. The DAG blockchain needs to sequence all blocks in the system and determine the linear sequence of transactions in it. The Lamport timestamp algorithm can determine the hapen-before relationship of processes, and all nodes in the system are recognized for the timestamps of all processes during the communication. Therefore, this scheme uses the Lamport timestamp algorithm to determine the sequential order of nodes in the DAG. We can see the detail in Algorithm 3.

Using the lamport timestamp algorithm it is possible to identify blocks that have direct links in the DAG. This will sort most of the blocks in the DAG, see in Figure 3

In Figure 3, we can see that there are many blocks with the same timestamp. These blocks are called concurrent blocks, and in the literature [13] the solution to the order of these concurrent blocks relies on high precision synchronized clocks. However, in distributed systems, it is very costly to maintain high precision synchronized clocks. Therefore, we must find a way to order these concurrent blocks and make sure that they are recognized by all nodes. PBFT [5] consensus algorithm is now widely used in the blockchain. This algorithm reaches consensus through constant communication between nodes, which can guarantee consensus among nodes in the system in the presence of one-third of malicious nodes. In addition, the throughput of the algorithm is greater than Bitcoin's POW mechanism.

After running Algorithm 4, all blocks with the same lamport timestamp will have a consensus time t, and this consensus time is public recognition by all nodes. The system needs to sort all the blocks in the DAG, and the sorting method is that the node with a small lamport timestamp is in front, and the node with the same lamport timestamp has a small consensus time in front.

5 Fine-grained Access Control Scheme

Layered IIoT architecture based on DAG must satisfy following security requirements:

1) Unforgeability: In the proposed model, all data col-

Algorithm 4 Consensus time algorithm

```
for each node in node poll do
    compute master node $p = v \ mod \ |R|$
end for
while all block in current node do
    request:
        request to sort blocks with the same lamport timestamp
        request higher consensus time for block of current node, and send request to the master node
    pre-prepare:
        compare block generation times of blocks with the same Lamport timestamp
        block with the smaller block generation time get small consensus time, if they are the same then initiate a vote
        broadcast pre-prepare message
        wait for all nodes to receive the message
    prepare:
    for each node in node poll do
            execute the sort request
            give the corresponding block a consensus time $t$
            broadcast prepare message
    end for
    commit:
        node receives at least 2f sorted results same as its own
        broadcast commit message
    reply:
        requesting node receives 2f+1 commit messages
        update DAG and state database
end while
```

lected by the equipment should sign then send to clients, the signature must pass the verification process. In our scheme, all equipment is assigned a unique identity, the identity is public to the entity. The scheme generates a private key and sends it to the equipment with a secret channel. An attacker can not forge a valid signature without a private key.

2) Collusion attack resistant: In the proposed model, we implement a distributed ABE mechanism. There are plenty of authorities represented by entities, for each authority processes its attributes. If multi authorities decide to collude by their attributes, the scheme can resist the collusion.

In the IIoT system, the usage of equipment in the system is only to collect information, and its corresponding security guarantee is quite low. In addition, during the operation of the system, equipment should continue to be added to the system. If equipment is not authenticated, the security of the entire system is considerably damaged. In this paper, an authentication is performed at the gateway whenever equipment joins the system, a unique identifier is generated. When subsequent equipment needs to send messages to the network, they need to sign and send using this unique identity, and the client can initiate a transaction only after the client verifies the signature.

As mentioned above, the equipment in the IIoT system is divided into entities, and the interaction of equipment in different entities needs to satisfy the access control policies of the other party. The access control mechanism used in the proposed model is the CP-ABE algorithm, which can embed the access control strategy into the ciphertext and achieve fine-grained access control at the ciphertext level. Because of different entities in the system, the equipment in the entity and the users in the system need to register attributes in the gateway to obtain the master private key and decryption key. Correspondingly, ensure the registered attributes in different entities are trustworthy to the system. If the system adopts a centralized registration center, it is hard to prevent a single point of error, attributes management etc. Therefore, this solution intends to adopt a decentralized ABE solution. Each attribute registration center in the system corresponds to an entity, and the attribute registration center not only has the attribute registration function, but it also acts as the entity's gateway and IBS registration center. In this system we adopt the existing ABE [15] and IBS [17] scheme for security and convenience. The symbols used in this scheme can be seen in Table 1

Below the summary, we deliver data interaction process:

1) System setup(λ) $\rightarrow$ SP The system setup algorithm takes in a security parameters λ as input and outputs system parameters SP for the system. Choose a bilinear group G of order $N = p_1 p_2 p_3$ and a hash function $H : \{0,1\}^* \rightarrow G$ that maps the bits vector to a group element of G. Then choose g_1 as the generator of G_{p1}, additionally, let $e : G \times G \rightarrow G_T$.

$$SP = (N, g_1) \quad (4)$$

2) Gateway setupA(SP) $\rightarrow$ SK, PK Each gateway runs the gateway setupA algorithm with parameters SP as input to produce its secret key and public key pair, SK, PK. For each attribute random choose α_i, $\beta_i \in \mathbb{Z}_N$.

$$PK_i = \{e(g_1, g_1)^{\alpha_i},\ g^{\beta_i} \forall i\}, SK = \{\alpha_i, \beta_i \forall i\} \quad (5)$$

3) Gateway setupB(k) $\rightarrow$ GBP Each gateway run the gateway setupB algorithm with a security parameters k as input and outputs gateway setupB parameters GBP for the gateway. Choose a random value $\alpha \in \mathbb{Z}_p$ and choose a random generator g and $g_3 \in G$.Compute $g2 = g^{\alpha}$. Then the gateway choose two random values $u^{'}, m^{'} \in G$ and two vectors $U = (u_i), M = (m_i)$ of length n_u, n_m(n_u represents the length of identity, n_m represents the length of message), besides u_i and m_i are both chosen from

Table 1: Symbols used in this scheme

Symbols	Description
G, G_T	group used in ABE and IBS
G_{p_1}	subgroup used in ABE
g, g_2, g_3	generator if group G and G_T
g_1	generator of group G_{p_1}
H	hash function
e	composite order bilinear pairing
λ_i, β_i	random value choose in $\mathbb{Z}_N$
SK, PK	private and public key of ABE
GBP	global parameter of IBS
MSK	master secret key of IBS
$u^{'}, m^{'}$	two random values choosed in G
U	random vector with length n_u
M	random vector with length n_m
EID	identity of equipment
K_{cid}	private key in IBS mechanism
r_u	random value choosed in Z_p
δ	signature
T	transcation
(A, ρ)	access control matrix
CT	ciphertext
s	random value
v	random vector
λ_x, ω_x	intermediate used in ABE
GID	identity used in ABE
i	attruibute
$K_{i,GID}$	attribute-identity pair

G randomly.

$$GBP = (params, MSK)$$
$$params = (G, G_T, e, g, g_2, g_3, u^{'}, m^{'}, U, M) \quad (6)$$
$$MSK = g_3^{\alpha}$$

4) Equipment KeyGen(GBP, EID) $\rightarrow K_{cid}$ The algorithm takes GBP generated from gateway setupB algorithm and a unique identity EID of the equipment as input. The algorithm then outputs an equipment secret key K_{cid} and sends it to the equipment in a safe channel. Let u be the string of n_u bits of identity EID, $u[i]$ represent the ith bit of the u, and $U \subset \{1, \ldots, n_u\}$ be the subString of u which all bits are 1. Choose a random value $r_u \in \mathbb{Z}_p$.

$$K_{cid} = (g_3^{\alpha}(u^{'} \prod_{i \in U} u_i)^{r_u}, g^{r_u}) \quad (7)$$

5) Equipment Sign(GBP, K_{cid}, m) $\rightarrow \delta$ Given the common parameter GBP, the private key K_{cid} and the message m , the equipment sign algorithm generate a signature δ of EID on $\mathfrak{m}$. Let $\mathfrak{m}$ be the bit string of length n_u representing the EID and let $\mathfrak{m}$ be the message. Let $\mathcal{U}$ be the set of index i such that $\mathfrak{u}[i] = 1$, and $\mathcal{M}$ be the set of index j that $\mathfrak{m}[j] = 1$. Choose a random value $r_m \in \mathbb{Z}_p$

$$\delta = (\delta_1, g^{r_u}, g^{r_m}) \in G^3$$
$$\delta_1 = g_3^{\alpha}(u^{'} \prod_{i \in U} u_i)^{r_u}(m^{'} \prod_{i \in \mathcal{M}} m_i)^{r}_{m} \quad (8)$$

6) Client Verify(GBP, EID, m, δ). The equipment send message m , its id EID and a signature δ to the client. And the client accept the message if the verify algorithm outputs true. Given a signature $\delta = (\delta[1], \delta[2], \delta[3]) \in \mathbb{G}_3$ of an identity EID on m, the verifier accepts δ if the following equality holds.

$$e(\delta[1], g) = e_1 e_2 e_3$$
$$e_1 = e(g_3, g_2)$$
$$e_2 = e(u^{'} \prod_{i \in \mathcal{U}} u_i, \delta[2]) \quad (9)$$
$$e_3 = e(m^{'} \prod_{i \in \mathcal{M}} m_i, \delta[3])$$

7) Client Transaction encryption (T, (A, ρ), SP, PK) $\rightarrow$ CT. After the client receives the message and passes verification, the client gets the correct message M. Then the client needs to package the message into a transaction. Then the client encrypts the transaction and sends it to the p2p network. The encryption algorithm takes a transaction T, an access matrix (A, ρ), the set of public keys for relevant authorities, and the system parameters SP. Then outputs a ciphertext CT. The algorithm chooses a random $s \in \mathbb{Z}_N$ and a random vector $v \in \mathbb{Z}_N^l$ with s as its first entry. Let λ_x denote $A_x \cdot v$, where A_x is row x of A. The algorithm also chooses a random vector $\omega \in Z_N^l$ with 0 as its first entry. Let ω_x denote $A_x \cdot \omega$. For each row A_x of A, chooses a random $r_x \in \mathbb{Z}_N$.

$$C_0 = Te(g_1, g_1)^s,$$
$$C_{1,x} = e(g_1, g_1)^{\lambda_x} e(g_1, g_1)^{\alpha_{\rho(x)} r_x}, \quad (10)$$
$$C_{2,x} = g_1^{\rho_x}, C_{3,x} = g_1^{y_{\rho(x)} r_x} \ g_1^{\omega_x} \forall x.$$

8) Client KeyGen(GID, SP, i, SK) $\rightarrow K_{i,GID}$ The client key generation algorithm takes in an identity GID, the system parameters SP, an attribute i belonging to some authority, and the secret key SK for the authority. It produces a key $K_{i,GID}$ for this attribute, identity pair.

$$K_{i,GID} = g_1^{\alpha_i} H(GID)^{y_i}. \quad (11)$$

9) Full client Decryption(CT, SP, $K_{i,GID}$) $\rightarrow T$ The decryption algorithm takes in the system parameters SP, the ciphertext, and a collection of keys corresponding to attribute, identity pair all with the same fixed identity GID. It outputs either the message M when the collection of attributes i satisfies the access matrix corresponding to the ciphertext. Otherwise, decryption fails. We assume the ciphertext

is encrypted under an access matrix (A, ρ). To decrypt, the decryptor first obtains $H(GID)$ from the random oracle. If the decryptor has the secret keys $\{K_{\rho(x),GID}\}$ for a subset of rows A_x of A such that $(1, 0, \ldots, 0)$ is in the span of these rows, then the decryptor proceeds as follows. For each such x, the decryptor computes.

$$\frac{C_{1,x} \cdot e(H(GID), C_{3,x})}{e(K_{\rho(x),GID}, C_{2,x})} = e(g_1, g_1)^{\lambda_x} e(H(GID), g_1)^{\omega_x} \tag{12}$$

The decryptor then chooses constants $c_x \in \mathbb{Z}_N$ such that $\sum_x c_x A_x = (1, 0, \ldots, 0)$ and computes:

$$\prod_x (e(g_1, g_1)^{\lambda_x} e(H(GID), g_1)^{\omega_x})^{c_x} = e(g_1, g_1)^s. \tag{13}$$

Recall that $\lambda_x = A_x \cdot v$ and $\omega_x = A_x \cdot \omega$, where $v \cdot (1, 0, \ldots, 0) = s$ and $\omega \cdot (1, 0, \ldots, 0) = 0$. The message can then be obtained as:

$$T = C_0 / e(g_1, g_1)^s. \tag{14}$$

6 Security and Privacy Analysis

6.1 Correctness

6.1.1 Correctness of IBS

Given a private key $K_{cid} = (g_3^{\alpha}(u^{'} \prod_{i\in U} u_i)^{r_u}, g^{r_u})$ the number of server n and a threshold parameter t, this algorithm distributes K_{cid} to n servers as follows.

1) First, the gateway picks $a_0, a_1, \ldots, a_{t-1} \in \mathbb{Z}_p$, constructs the polynomial $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$ over $\mathbb{Z}_p$ which set $r_u^{'} = a_0$.

2) Second, it computes the public parameter $Y = (g_3^{\alpha}(u^{'} \prod_{i\in U} u_i)^{r_u - r_{u'}}, g^{r_u})$ for all n parties:

3) Third, for each equipment in entity E_h, it computes the shared key $K_{cid,h} = f(h)$, and the verification key $k_h = e(u^{'} \prod_{i\in\mathcal{U}} u^{'}, g)^{f(h)}$:

4) Last, the gateway secretly sends the distrubuted privated key $K_{cid,h}$ to each equipment E_h, $1 \leq h \leq n$, and publishs $Y, y_1, y_2, \ldots, y_n$.

Give the message m, the n shares $\{E_h\}_{k=1}^{n}$ generate the signature of the identity EID by following computation.

1) With its shared key $K_{cid,h} = f(k)$, each sharer E_h randomly selects $r_h \in \mathbb{Z}_p$, computes and broadcasts the signature share $\delta_h = ((u^{'} \prod_{i\in U} u_i)^{f(k)})(m^{'} \prod_{i\in\mathcal{M}} m_i)^{r}_{m}, g^{r_k})$

   With the verification key $E_k^{'}$ $k_h = k_h = e(u^{'} \prod_{i\in U} u_i)^{f(k)})$, the validity of the signature share $\delta_h = (\delta_h[1], \delta_h[2])$ due to player E_h can be publicly verified by checking $e(\delta_h[1], g) = k_h \cdot e(m^{'} \prod_{i\in\mathcal{M}} m_i, \delta_h[2])$

2) Each equipment locally reconstructs the full signature as follows. It first collect t valid signature shares using the above verifycation equation. Suppose that Φ is the set of indices of t honest players who generated valid signature shares. Given the signatrue shares $\{\delta_k\}_{k\in\Phi}$ and the public parameter Y:

$$\begin{aligned} \{\delta_h\}_{h\in\Phi} &= \{(\delta_h[1], \delta_h[2])\}_{h\in\Phi}, \\ Y &= (Y[1], Y[2]) \\ &= (g_3^{\alpha}(u^{'} \prod_{i\in U} u_i)^{r_u - r_{u'}}, g^{r_u}) \end{aligned} \tag{15}$$

the signature $\delta = (\delta[1], \delta[2], \delta[3])$ is computed as follows.

$$\begin{aligned} \delta[1] &= Y[1] \prod_{h\in\Phi} \delta_h[1]^{l_{\Phi,h}}, \\ \delta[2] &= Y[2], \\ \delta[3] &= \prod_{h\in\Phi} \delta_h[2]^{l_{\Phi,h}}, \end{aligned} \tag{16}$$

where the Lagrange coefficient $l_{\Phi,h} = \prod_{j\in\Phi, j\neq h} \frac{-j}{h-j}$.Since

$$\begin{aligned} \delta[1] = Y[1] \prod_{h\in\Phi} \delta_h[1]^{l_{\Phi,h}} & \\ = Y[1]((u^{'} \prod_{i\in U} u_i)^{\sum_{h\in\Phi} f(k) l_{\Phi,h}})(m^{'} \prod_{i\in\mathcal{M}} m_i)^{\sum_{h\in\Phi} r_h l_{\Phi,h}}) & \\ = g_3^{\alpha}(u^{'} \prod_{i\in U} u_i)^{r_u}(m^{'} \prod_{i\in\mathcal{M}} m_i)^{\sum_{h\in\Phi} r_h l_{\Phi,h}}), & \\ \delta[2] = Y[2] = g^{r_u}, & \\ \delta[3] = \prod_{h\in\Phi} \delta_h[2]^{l_{\Phi,h}} = (m^{'} \prod_{i\in\mathcal{M}} m_i)^{\sum_{h\in\Phi} r_h l_{\Phi,h}}), & \end{aligned} \tag{17}$$

it is obvious that $\delta = (\delta[1], \delta[2], \delta[3])$ is a valid signature. In other words, the correctness property of our IBS mechanism is satisfied.

6.1.2 Correctness of ABE

In ABE mechanism, the ciphertext is divided into four parts. The correctness of decryption is verified as follows. The ABE mechanism we adopted is distributed. First, we verify Equation 18.

$$\begin{aligned} & \frac{C_{1,x} \cdot e(H(GID)C_{3,x}}{e(K_{\rho(x),GID}, C_{2,x})} \\ & = \frac{e(g_1, g_1)^{\lambda_x} e(g_1, g_1)^{\alpha\rho(x) r_x} \cdot e(H(GID), g_1^{y\rho(x) r_x} g_1^{w_x})}{e(g_1^{\alpha_{\rho(x)}} H(GID)^{y_{\rho(x)}}, g_1^{r(x)})} \\ & = \frac{e(g_1, g_1)^{\lambda_x} e(g_1, g_1)^{\alpha\rho(x) r_x} \cdot e(H(GID), g_1^{y\rho(x) r_x} g_1^{w_x})}{e(g_1, g_1)^{\alpha_\rho(x) r_x} e(H(GID), g_1)^{y_\rho(x) r_x}} \\ & = \frac{e(g_1, g_1)^{\lambda_x} \cdot e(H(GID), g_1^{w_x}) \cdot e(H(GID), g_1^{y_{\rho(x)} r_x})}{e(H(GID), g_1)^{y_\rho(x) r_x}} \\ & = e(g_1, g_1)^{\lambda_x} e(H(GID), g_1)^{w_x} \end{aligned} \tag{18}$$

then we verify Equation 19. We should recall that $\lambda_x = A_x \cdot v$ and $w_x = A_x \cdot w$. Then

$$\begin{aligned}&\prod_x (e(g_1, g_1)^{\lambda_x} e(H(GID), g_1)^{\omega_x})^{c_x} \\ &= \prod_x (e(g_1, g_1)^{A_x C_x \cdot v} e(H(GID), g_1)^{\omega A_x C_x}\end{aligned} \tag{19}$$

From the above, we know that $A_x C_x \cdot v = s$ and $\omega A_x C_x = 0$. So

$$\begin{aligned}&\prod_x (e(g_1, g_1)^{A_x C_x \cdot v} e(H(GID), g_1)^{\omega A_x C_x} \\ &= e(g_1, g_1)^s e(H(GID), g_1)^0 \\ &= e(g_1, g_1)^s\end{aligned} \tag{20}$$

From the above, we know that $C_0 = Me(g_1, g_1)^s$. Therefore, we can obtained the Message M as:

$$T = \frac{C_0}{e(g_1, g_1)^s} = \frac{Me(g_1, g_1)^s}{e(g_1, g_1)^s} = T \tag{21}$$

So the correctness property of our ABE mechanism is satisfied.

6.2 Security

1) Unforgeability: The IBS mechanism is adopted from existing scheme, so the unforgeability of IBS mechanism is rely on three theorems in the existing scheme [17]. The following is three theorems:

Theorem 1. *The (t, n) IBS mechanism is robust in the presence of up to t - 1 malicious servers if $n \geq 2t - 1$*

Theorem 2. *The (t, n) IBS mechanism is $(t_2, q_e, q_s, \epsilon)$ UF-IDS secure, assuming the scheme [23] is $(t_1, q_e, q_s, \epsilon)$ UF-IDS secure where*

$$t_1 = t_2 + q_s(t + 1)(n - t + 1)T_e \tag{22}$$

and T_e is the time for exponentiation in G.

Theorem 3. *Theorem 1 The Paterson-Schuldt identity-based signature scheme is (t, q_e, q_s, t) unforgeable against adpative chosen identity and message attack in the standard model, assuming that the CDH problem in G is (t', ϵ') is intractable, where*

$$\begin{aligned}\epsilon' &= \frac{\epsilon}{16(q_e + q_s)q_s(n_u + 1)(n_m + 1)} \\ t' &= t + O\{[q_e n_u + q_s(n_u + n_m)]\rho + (q_e + q_s)\tau\},\end{aligned} \tag{23}$$

where ρ is the time for a multiplication in G_1 and τ for an exponentiation.

Proof. Our IBS mechanism is based on existing paper, so the basic security assurance is same to the paper. If you want see proof of three theorems, please refer to [17] in detail. □

Combining the above theorems 5.1, 5.2, 5.3, we can obtain the unforgeablity of the IBS mechanism. Which is the IBS mechanism is $UF-IBTHS$ secure against an adversary who corrupts up to $t \leq (n+1)/2$ players, if the CDH problem is intractable in group G.

2) Collusion attack resistance: To prevent collusion attacks, in our scheme the ABE mechanism uses the global identity to tie together the various attributes to one specific user so that the global identities cannot be successfully combined with the attributes of another user in decryption. The message M is blinded by encryption algorithm with $e(g_1, g_1)^s$, which the g_1 is a generator of subgroup G_{P_1}, and the s is a random value of $\mathbb{Z}_N$. The value s can be split into shared λ_x according to the matrix, the value 0 can be split into shared ω_x. The decryptor must recover the blinding factor $e(g_1, g_1)^s$ by pairing their keys for attribute, identity pairs (i, EID) with ciphertext elements to obtain the shared s. If the decryptor has a satisfying set of keys with the same identity EID, these additional results will cancel from the final result, because the vector ω_x are shares of 0. So two users with different identities EID_1 and EID_2 attempt to collude and combine their keys, there must be some result that can not be canceled with each other, thereby preventing the recovery of $e(g_1, g_1)^s$ [15].

6.3 Privacy Analysis

In current IIoT systems, there is a lack of equipment-to-client authentication measures. The equipment collects information and sends it to the client, the attacker can easily forge the collected information and send it to the IIoT system. This lead to great damage to the security of the system and the integrity of data. So in our work, the scheme integrate IBS into the IIoT system. The scheme divides the IIoT system into different entities. And in one entity, there are a certain equipment, several clients, and a unique gateway. They both belong to the entity, the gateway act as an authority in the entity. The gateway can deliver unique identity to equipment and generate public and secret keys. Also, the gateway can send the secret key to the corresponding equipment. Then the equipment can sign the message with the secret key, the client who receives the signature can use the public identity to verify the validity of the message.

The related works either keep the transaction data in a plain domain or use symmetric encryption like AES for encryption. If data is saved in a plain domain, this means everyone can see the sensitive data. The purpose of this scheme is to preserve data confidentiality, so we must avoid this phenomenon. Also, if the data is encrypted by symmetric encryption. It will lead to plenty of generation of secret keys and their transmission, which result in a bunch of waste a large of Storage and communication resources. Besides, the owner of data can not control which one is qualified to access the data. Not to mention fine-grained access control.

In this paper, the DAG network is composed of light clients and full nodes. And the full nodes are responsible to verify the transaction. Since the transaction is encrypted with ABE algorithm, the node that can decrypt the transaction is the node that is qualified to verify the transaction. This means the transaction does not need all full nodes to verify its confidentiality, which can improve the speed of verification of transcation.

The goal of this paper is to ensure the data integrity of the data owners. In this paper, the transactions are not stored in plaintext. Transactions must be encrypted before uploading to the full nodes. The full nodes that are not eligible for decryption can only package it into blocks and can not view private data, which ensure data privacy for the data owners.

The IIoT system is known for the attack of Man-in-the-middle and equipment hijacking. With the IBS mechanism, the attack can not forge the secret key therefore the system can resist the attack of Man-in-the-middle and equipment hijacking. Even though the attacker breaks the IBS mechanism, the attack can not obtain sensitive data either due to the ABE mechanism. Since the proposed model is built on top of the well researched IBS, decentralized ABE, and blockchain technology, we can assume that there is no security vulnerability in the rest of the model [25].

7 Experimental Evaluation

7.1 Numerical Analyses

Compare with the proposed model against AES-based blockchain models [4, 9, 10], the proposed model has improved in scalability and key management. First, if a blockchain system uses AES for the encryption algorithm, when a new node joins the blockchain network, the new node must apply keys with all nodes. This causes damage to the scalability of the whole system, which leads to a key management problem, each node should generate and keep secret keys with all nodes. That means lots of system resources are wasted on key generate and management. The proposed model uses ABE as an encryption algorithm, the keys are generated by the gateway according to attributes it keeps and then delivery keys to clients. No need to match with each node is greatly reduced the key required by the system. Therefore, the scalability and key management problems can be alleviated.

Nevertheless, the adoption of IBS and ABE increases the computational cost for encryption and decryption in contrast to the original scheme. We can numerically check the computation costs of the IBS and ABE mechanism in the proposed model. As mentioned above, IBS and ABE mechanisms both have five distinct algorithms. And not all of them need real-time computing. In IBS mechanism, system setup, gateway setupB, and equipment keygen can compute off-line. In the ABE mechanism, system setup, gateway setupA, and client keygen can compute off-line. The computational cost for hash functions is negligible compared to pairing and exponentiation.

This scheme is running on the industrial IoT architecture, and there are many studies exploring access control issues in industrial IoT today. In this paper, the literature [25], literature [33], literature [18], and literature [34] are compared with our scheme using four metrics: availability of device authentication measures, availability of access control mechanisms, access control structure, and ability to outsource encryption. The comparisons are shown in Table 2 and Table 3.

Table 2: Scheme comparison I

Scheme	Authentication	Access control
scheme [21]	N	Y
scheme [34]	N	Y
scheme [35]	N	Y
scheme [36]	N	Y
our scheme	Y	Y

Table 3: Scheme comparison II

Scheme	Access Structure	Multi-authority
scheme [21]	And-gate	Y
scheme [34]	Access tree	N
scheme [35]	And-gate	N
scheme [36]	Access tree	N
our scheme	LSSS	Y

7.2 Experimental Analyses

7.2.1 Experiment of Consensus Algorithm

The throughput of bitcoin is about 3-7 tps, and such a low throughput is not sufficient in IIoT applications that require high performance and high concurrency. In this scheme, block generation does not require all nodes to compete with each other, and new blocks can be generated as long as the nodes' transaction pools meet the requirements. The new block needs to link other nodes and obtain its own Lamport timestamp and consensus timestamp in the subsequent consensus process.

In order to be applicable to real-life scenarios industrial IoT applications with high performance and high concurrency, the throughput of this scheme cannot be too low, so the core of the algorithm needs to be implemented and the performance of the scheme needs to be tested. So we focus on testing the performance of the consensus algorithm, and we simulate the DAG consensus algorithm and tests the performance of the algorithm under different scenarios by changing the parameters of the algorithm.

In the efficiency test of this solution, we test the impact of each parameter on the performance of the algorithm. The first test is the throughput of the system in relation to the total number of requested messages. The full nodes of the system were set to a total of seven, the malicious nodes to two, the number of clients to three, the node timeout time to 500 ms, the client timeout time to 800 ms, and the basic network experiment between nodes to 2 seconds. The test results are shown in the Figure 4 and Figrue 5.

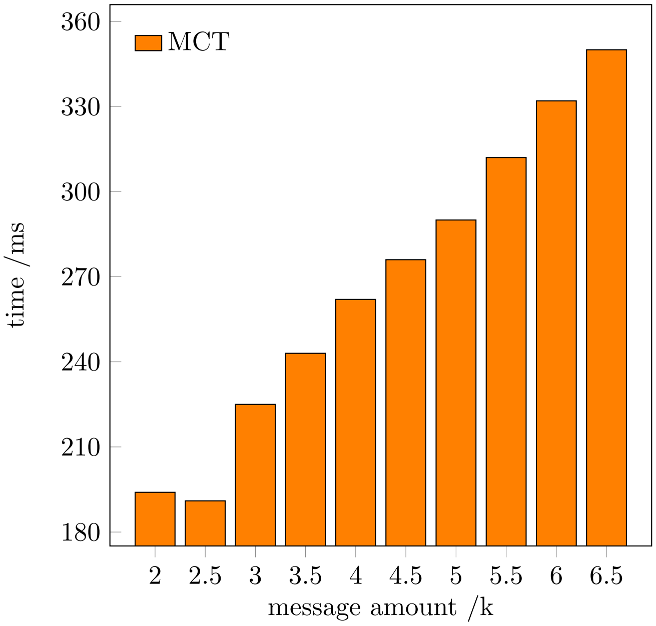

Figure 4: Average message confirmation time

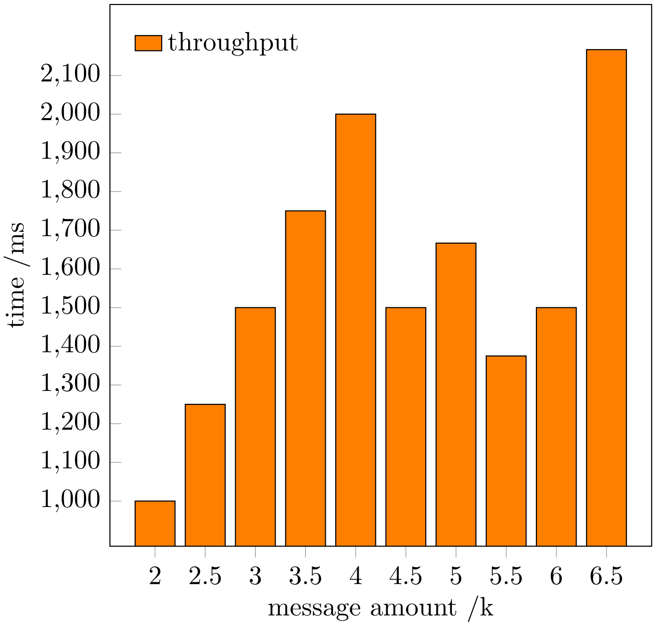

Figure 5: System throughput

Observing Figure 4, it can be seen that the message acknowledgement time is increasing as the total number of messages increases with the same number of nodes and clients. Observing Figure 5, it can be seen that the throughput of the system does not increase with the total number of messages; the throughput fluctuates and decreases after increasing to a maximum point. The reason for this is that as the total number of messages increases without any change in the number of nodes and clients processing the messages, the average message acknowledgement time is bound to keep increasing. As for the throughput, the throughput of the system increases as the total number of messages increases when the total number of messages reaches the message processing limit of the nodes. And when the node reaches the upper limit of message processing, the increase in the total number of messages will bring blockage to the system. The throughput of the system also tends to decrease. In the implementation of the system, there is a certain randomness in message sending, so the downward trend is not always down, and there is a certain volatility. Compared to the 3-7tps performance of the Bitcoin system, the DAG consensus algorithm of this scheme is fully usable in industrial IoT systems.

7.2.2 Experiment of Access Control Scheme

According to the below Table 4 and Table 5, we can find that the most time-consuming algorithm is the setup algorithm. Fortunately, we can compute it off-line. Imaging that an entity is established initially, the entity should assign an identity to equipment and execute the setup algorithm, and the system is not prepared to function. We can execute the setup algorithm for from the beginning as the preparatory work. Therefore, the time-consuming algorithm has little impact on the performance of the system. And the keygen algorithm has the same reason for computing off-line. Also if we observe Table 4 and Table 5, we can find that the time complexity of keyGen is not rising as the length of the message increases. But the time complexity of keygen is increasing as the length of the identity increases. When the length of identity remains unchanged, the time complexity of the keygen algorithm can be affected by the random oracle, so the time complexity seems to be random.

The equipment is responsible for collecting information and signing the information. In the actual application environment, the information can just be timestamp and temperature. So the information can be represented by only 300 bits, but considering the scalability of the system in the future, we also measure the longer bits. Also in an entity 16 bits is enough for an appliance, but considering the scalability of the system, we also measure the 32 bits identity. And if we observe the Figure 6, we can find that the time complexity of the sign and verify is increasing as the length of the message increases.

When we observe the above Table 6, Table 7 and Figure 7, Figure 8, we can find that the time consumption of all five algorithms in the ABE mechanism is increasing with the increase of attributes. If there is only one entity in the whole system, the time consumption is less than

Table 4: Time complexity when identity length is 16 (ms)

Length	Setup	KeyGen	Sign	Verify
300	481.40	17.54	23.58	29.69
350	538.84	17.16	24.39	30.49
400	587.28	16.89	25.31	31.68
450	654.87	17.16	26.22	32.05
500	702.59	16.76	26.73	33.78
550	764.47	17.54	28.11	34.33
600	830.79	17.40	29.43	36.05
650	909.54	17.17	30.40	37.63
700	958.29	17.34	31.18	37.01
750	1017.32	16.99	32.30	40.40

Table 5: Time complexity when identity length is 32 (ms)

Length	Setup	KeyGen	Sign	Verify
300	506.27	18.28	23.29	30.71
350	581.60	17.66	24.05	31.90
400	659.31	18.23	26.27	34.33
450	701.00	17.88	26.87	34.61
500	773.20	18.27	28.34	35.19
550	800.15	18.47	28.49	36.05
600	889.85	17.82	29.50	36.34
650	918.83	18.14	30.45	38.97
700	990.48	18.39	31.29	39.96
750	1027.43	18.00	32.17	40.77

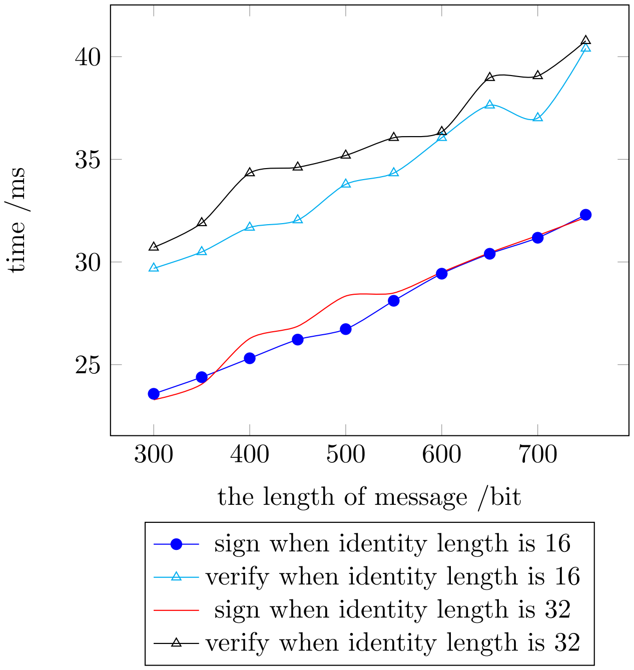

Figure 6: Time complexity of sign and verify algorithm

the situation when there are ten entities in the system. But the disparity is insignificant. Only the encrypt and decrypt must compute online. The other three algorithms can compute in the initial stage or participate in the system dynamically. That means the whole mechanism is scalable. According to the above experimental data, we can say that although the introduction of the IBS and Abe algorithm affects the efficiency of the system, it does not significantly reduce the performance of the DAG network.

Table 6: Time complexity when gateway number is 1 (ms)

	Gsetup	Asetup	KeyGen	Enc	Dec
2	1267.38	65.77	99.71	172.88	65.92
3	1336.67	99.28	154.49	249.28	95.56
4	1285.76	125.24	200.61	327.07	136.67
5	1235.49	154.10	252.45	405.66	162.58
6	1253.36	181.87	297.76	481.54	192.70
7	1260.48	209.55	355.33	568.16	228.15
8	1282.50	236.40	400.44	636.75	256.08
9	1267.22	273.43	438.03	707.10	288.04
10	1296.35	320.47	486.12	797.00	320.67
11	1327.57	336.50	543.12	850.51	345.08

Table 7: Time complexity when gateway number is 10 (ms)

	Gsetup	Asetup	KeyGen	Enc	Dec
2	1312.35	72,94	53.10	171.78	67.03
3	1341.85	133.28	164.00	255.45	98.74
4	1371.55	154,16	205.54	322.93	131.49
5	1383.08	185.49	257.15	469.36	194.40
6	1373.28	215.62	305.36	540.78	219.45
7	1389.09	247.57	359.43	631.72	247.30
8	1293.83	266.61	405.04	706.67	285.79
9	1391.22	298.56	453.58	774.12	347.51
10	1413.11	325.59	502.68	830.04	355.21
11	1391.89	353.87	551.70	935.77	375.37

8 Conclusion

IIoT systems are vulnerable to single point of failure when storing data, and data owners may tamper with or deny historical data for their own benefit. To address these issues, we have added blockchain technology to the IIoT architecture. Since the performance and concurrency of blockchain technology cannot meet the needs of IIoT, we replaced the blockchain technology with DAG blockchain. In some IIoT applications that require strict data privacy, the lack of secure authentication and access control mechanisms for devices can be detrimental to these applications. To solve the above problems, we propose an IIoT architecture based on the DAG blockchain, and we also do a layered operation of the DAG network in order to facilitate the understanding and management of devices. To make this architecture practical, we design an

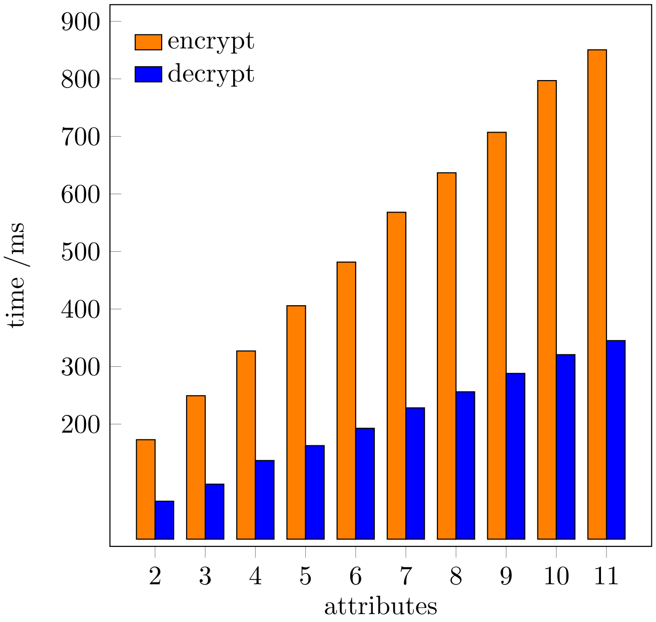

Figure 7: Time complexity of encrypt and decrypt algorithm when entity number is 1

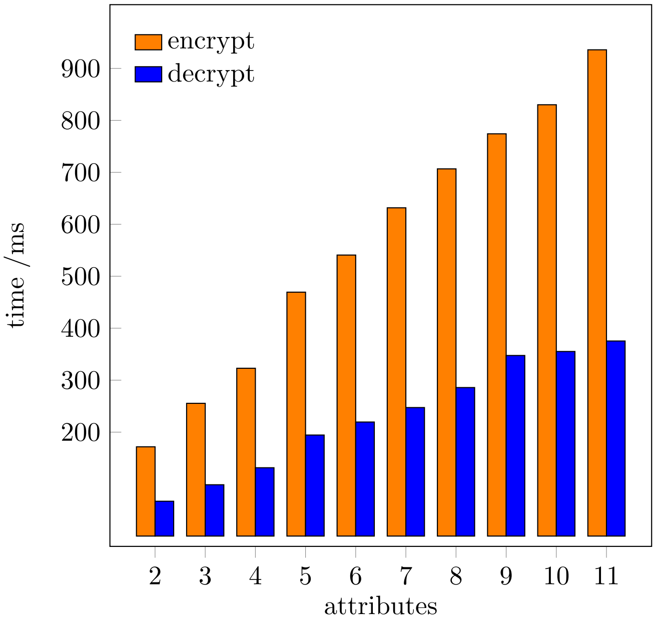

Figure 8: Time complexity of encrypt and decrypt algorithm when entity number is 10

efficient DAG consensus algorithm, and the experimental results show that our architecture is usable for IIoT. On top of this architecture, we also propose a fine-grained access control scheme. It makes the IIoT devices must be authenticated before joining the system, which ensures the reliability of data. And the data must meet the access control conditions in the process of sharing, and the data cannot be accessed without meeting the access control conditions. The experimental results of the scheme show that the performance of the IIoT system will not be affected too much after the security mechanism is added to the system.

Acknowledgments

This work was supported by the Natural Science Foundation of Chongqing (no.cstc2020jcyj-msxmX0343).

References

[1] T. Alam, "A survey on the use of blockchain for the internet of things," *International Journal of Electronics and Information Engineering*, vol. 13, no. 3, pp. 119–130, 2021.

[2] A. Beimel *et al.*, "Secure schemes for secret sharing and key distribution," Technical Report, Technion-Israel Institute of Technology, 1996.

[3] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Theory of cryptography conference*. Springer, 2005, pp. 325–341.

[4] A. Boudguiga, N. Bouzerna, L. Granboulan, A. Olivereau, F. Quesnel, A. Roger, and R. Sirdey, "Towards better availability and accountability for iot updates by means of a blockchain," in *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2017, pp. 50–58.

[5] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OsDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[6] S. M. Choi, J. Park, Q. Nguyen, and A. Cronje, "Fantom: A scalable framework for asynchronous distributed systems," *arXiv preprint arXiv:1810.10360*, 2018.

[7] A. Churyumov, "Byteball: A decentralized system for storage and transfer of value," *URL https://byteball. org/Byteball. pdf*, 2016.

[8] L. Cui, S. Yang, Z. Chen, Y. Pan, M. Xu, and K. Xu, "An efficient and compacted dag-based blockchain protocol for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4134–4145, 2019.

[9] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for iot," in *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 2017, pp. 173–178.

[10] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*. IEEE, 2017, pp. 618–623.

[11] Hedera, *Hashgraph*, 2015. (https://www.hederahashgraph.com/)

[12] U. Kannengiesser and H. Müller, "Towards viewpoint-oriented engineering for industry 4.0:

A standards-based approach," in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*. IEEE, 2018, pp. 51–56.

[13] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," in *Concurrency: the Works of Leslie Lamport*, 2019, pp. 179–196.

[14] S. D. Lerner, *Dagcoin Draft*, 2015. (https://bitslog.files.wordpress.com/2015/09/dagcoinv41.pdf)

[15] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2011, pp. 568–588.

[16] C. Li, P. Li, D. Zhou, W. Xu, F. Long, and A. Yao, "Scaling nakamoto consensus to thousands of transactions per second," *arXiv preprint arXiv:1805.03870*, 2018.

[17] F. Li, W. Gao, G. Wang, K. Chen, and X. Wang, "Efficient identity-based threshold signature scheme from bilinear pairings in standard model," *International Journal of Internet Protocol Technology 7*, vol. 8, no. 2-3, pp. 107–115, 2014.

[18] J. Li, Y. Shi, and Y. Zhang, "Searchable ciphertext-policy attribute-based encryption with revocation in cloud storage," *International Journal of Communication Systems*, vol. 30, no. 1, p. e2942, 2017.

[19] Y. Liu, K. Wang, Y. Lin, and W. Xu, "Lightchain: a lightweight blockchain system for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3571–3581, 2019.

[20] S. Mumtaz, A. Alsohaily, Z. Pang, A. Rayes, K. F. Tsang, and J. Rodriguez, "Massive internet of things for industrial applications: Addressing wireless iiot connectivity challenges and ecosystem fragmentation," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 28–33, 2017.

[21] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[22] A. C. Panchal, V. M. Khadse, and P. N. Mahalle, "Security issues in iiot: A comprehensive survey of attacks on iiot and its countermeasures," in *2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN)*. IEEE, 2018, pp. 124–130.

[23] K. G. Paterson and J. C. Schuldt, "Efficient identity-based signatures secure in the standard model," in *Australasian conference on information security and privacy*. Springer, 2006, pp. 207–222.

[24] S. Popov, "The tangle," *White paper*, vol. 1, no. 3, 2018.

[25] Y. Rahulamathavan, R. C.-W. Phan, M. Rajarajan, S. Misra, and A. Kondoz, "Privacy-preserving blockchain based iot ecosystem using attribute-based encryption," in *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 2017, pp. 1–6.

[26] A. Rojko, "Industry 4.0 concept: Background and overview." *International Journal of Interactive Mobile Technologies*, vol. 11, no. 5, 2017.

[27] Y. Sompolinsky and A. Zohar, "Phantom," *IACR Cryptology ePrint Archive, Report 2018/104*, 2018.

[28] F. Tang, S. Ma, Y. Xiang, and C. Lin, "An efficient authentication scheme for blockchain-based electronic health records," *IEEE access*, vol. 7, pp. 41 678–41 689, 2019.

[29] K. Tsiknas, D. Taketzis, K. Demertzis, and C. Skianis, "Cyber threats to industrial iot: A survey on attacks and countermeasures," *IoT*, vol. 2, no. 1, pp. 163–188, 2021.

[30] J. Wan, J. Li, M. Imran, D. Li *et al.*, "A blockchain-based solution for enhancing security and privacy in smart factory," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3652–3660, 2019.

[31] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[32] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in internet-of-things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, 2017.

[33] K. Yu, L. Tan, M. Aloqaily, H. Yang, and Y. Jararweh, "Blockchain-enhanced data sharing with traceable and direct revocation in iiot," *IEEE transactions on industrial informatics*, vol. 17, no. 11, pp. 7669–7678, 2021.

[34] Y. Zhang, D. He, and K.-K. R. Choo, "Bads: Blockchain-based architecture for data sharing with abs and cp-abe in iot," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[35] K. Zhou, T. Liu, and L. Zhou, "Industry 4.0: Towards future industrial opportunities and challenges," in *2015 12th International conference on fuzzy systems and knowledge discovery (FSKD)*. IEEE, 2015, pp. 2147–2152.

Biography

Fei Tang received his Ph.D from the Institute of Information Enginneering of Chinese Academy of Sciences in 2015. He is currently an associate professor of the School of Cyberspace Security and Law, Chongqing University of Posts and Telecommunications. His research interests are public key cryptography, blockchain and Privacy-Preserving Computation.

Zhangtao Ye is a is a graduate student of Chongqing University of Posts and telecommunications. His research interests are public key cryptography and blockchain.

Kun Dong is a is a graduate student of Chongqing University of Posts and telecommunications. His research interests are public key cryptography and privacy computing.

Dong Huang received his Ph.D from the Chongqing University in 2012. He is currently an professor of the Key Laboratory of Advanced Manufacturing Technology of Ministry of Education, Guizhou University. His research interests are cryptography and intelligent security.