applied sciences

MDPI

Article

Identity Management and Access Control Based on Blockchain under Edge Computing for the Industrial Internet of Things

Yongjun Ren [1,2], Fujian Zhu [1,2], Jian Qi [1,2], Jin Wang [3,4,*] and Arun Kumar Sangaiah [5]

[1] School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China; renyj100@126.com (Y.R.); zhufujian1995@gmail.com (F.Z.); qijian19940420@163.com (J.Q.)
[2] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science & Technology, Nanjing 210044, China
[3] School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha 410004, China
[4] School of Information Science and Engineering, Fujian University of Technology, Fujian 350118, China
[5] School of Computing Science and Engineering, Vellore Institute of Technology (VIT), Tamil Nadu, Vellore 632014, India; sarunkumar@vit.ac.in
* Correspondence: jinwang@csust.edu.cn

Received: 28 April 2019; Accepted: 14 May 2019; Published: 18 May 2019

Featured Application: There are more and more sensor nodes in the Internet of Things. Because of the lack of transmission bandwidth in the IoT environment, some sensors may not be able to upload valid data in time in the case of large-scale sensors. Our proposed solution can be applied to the edge network of large-scale nodes, transmitting more information under limited bandwidth, and maintaining the protocol of identity management and access control more precisely through blockchain technology.

Abstract: Edge computing provides a unified platform for computing, networking, and storage resources, enabling data to be processed in a timely and efficient manner near the source. Thus, it has become the basic platform for industrial Internet of things (IIoT). However, computing's unique features have also introduced new security problems. To solve the problem, in this paper, blockchain-based identity management combining access control mechanism is designed under edge computing. The self-certified cryptography is utilized to realize the registration and authentication of network entities. We bind the generated implicit certificate to its identity and construct the identity and certificate management mechanism based on blockchain. Secondly, an access control mechanism based on Bloom filter is designed and integrated with identity management. Moreover, for secure communication in resource-constrained edge devices, a lightweight secret key agreement protocol based on self-authenticated public key is constructed. These mechanisms work together to provide data security guarantees for IIoT such as authentication, auditability, and confidentiality.

Keywords: edge computing; industrial internet of things; identity management; access control

1. Introduction

In recent years, the rapid development of the Internet of Things has promoted social and economic development [1,2]. The United States Army published the Emerging Science and Technology Trends: 2016–2045—A Synthesis of Leading Forecasts Report in 2016. The report

concludes that more than 100 billion devices, including mobile phones and wearable devices, medical devices, electrical appliances, industrial sensors, surveillance cameras, cars, and clothing, will be connected to the Internet by 2045. These facilities fully automate the inspection, management, and maintenance of the original labor force [3,4].

At present, a great deal of research work has applied the Internet of things technology to various industrial control systems (ICS). Moreover, the industrial Internet of things (IIoT) has also been considered to be the pillar of industry 4.0 [5], qnd it is the key to the improvement of intelligent manufacturing. It integrates all kinds of sensors, controllers, special equipment, and advanced information technology with the ability of perception and monitoring into all links of industrial production process, collects data, and carries out the task of expanding the enterprise's capability [6–8]. Thus, it greatly improves the efficiency of production and the competitiveness of enterprises, helps to promote the coordinated and harmonious development of people, society, and nature, and ultimately upgrades the traditional industry to a new stage of intelligence.

Although cloud computing provides a computing platform for data processing of IIoT, the growth of network bandwidth is far from meeting the demand for data growth [9–11]. At the same time, the complicated network environment makes it difficult for the network delay to have a breakthrough improvement [12–14]. To solve the problems, edge computing arises at the historic moment, and has had extensive attention of researchers in the past two years [15,16]. The edge of the edge computing refers to the computing and storage resources on the edge of the network, which is closer to the user whether it is from the geographical distance or the network distance. Also, edge computing is a technology that uses these resources to provide services for users at the edge of the network, enabling applications to process data near the data source [17,18]. Thus, edge computing offers better support for mobile computing and IIoT applications than cloud computing.

One of the critical requirements of edge computing for IIoT applications is security. First of all, edge computing contains a large number of intelligent manufacturing terminals, which have potential security problems. For example, 82% of Android devices have at least 1 of 25 security vulnerabilities. Secondly, there are many kinds of networks connected by terminals under edge computing, and the security of the networks is difficult to guarantee, making them more vulnerable to attack [19]. According to statistics, 80% of routers use the default password. Moreover, some sensor devices have very limited resources, which make many existing security technologies unable to be used directly [20,21].

Blockchain technology successfully achieves consensus among distributed participants with malicious nodes without the intervention of any trust intermediary. Because of its similar topology to the Internet of things, blockchain technology has recently been applied in the internet of things to provide security and privacy protection [22,23]. This paper will make use of blockchain technology to build identity management and access control mechanism under edge computing to ensure data security of IIoT.

The rest of this paper is organized as follows. Section 2 introduces the related works of intelligent manufacturing and IIoT, and access control mechanism in IoT. In Section 3, we analyze the existing problem of data security in IIoT. Section 4 constructs identity management and access control mechanism based on blockchain under edge computing for IIoT. Finally, Section 5 draws conclusions of this paper.

2. Related Work

2.1. Intelligent Manufacturing and IIoT

Industrial internet of things is the main body of intelligent manufacturing system. The functions related to manufacturing equipment, production line control, and scheduling in manufacturing execution system are implemented through the Industrial Internet of Things [24]. The architecture of intelligent manufacturing system is shown in Figure 1. Industrial internet of things promotes the development of intelligent manufacturing, and the transformation and upgrading of industry [25,26]. The security of industrial internet of things is of great significance to the safety of intelligent

manufacturing system. IIoT is an important infrastructure of national lifeline industries, such as military defense technology, the grid, the petroleum and petrochemical industry, telecommunications, coal, civil aviation, shipping, and so on [27]. Today, with the increasingly fierce attack and defense war of network information security, it is facing a rising security risk.

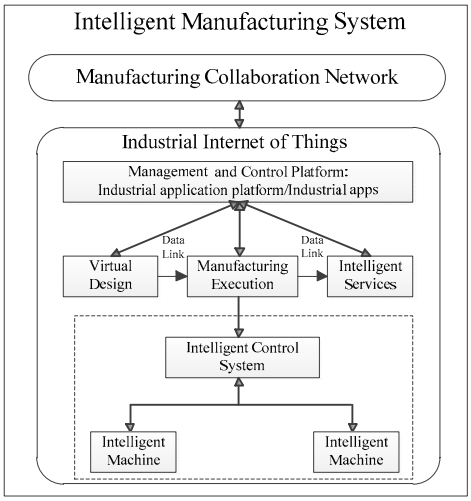

Figure 1. Architecture of intelligent manufacturing system.

2.2. Edge Computing

Edge computing was mainly initiated by Carnegie Mellon University (CMU) in 2015, which proposed a computing model with extensive descriptive significance from the perspective of academic research [28]. In June 2015, Carnegie Mellon University joined forces with Vodafone, Intel, and other companies and established the Open Edge Computing Initiative, which currently includes CMU, Intel, Nokia, and NTT. The organization defines edge computing as a new type of network model that provides computing and storage resources. This network function is located near the user's location. Edge computing also refers to a new computing model that performs calculations at the edge of the network [29,30].

These definitions all emphasize that edge computing is a new model. Its core concept is that computing should be closer to the source of data and can be closer to users. First of all, it shows that the distance of network devices is close. So, the unstable factors such as bandwidth, delay, and jitter can be easily controlled and improved due to the reduction of network size [31]. Second, the resources and users of edge network are in the same situation (e.g., location). Thus, the personalized services for users can be provided, according to the scene information, such as location-based services.

The authors in the literature [32] proposed a three-tier network architecture composed of an edge network, edge network management center, and cloud server. The architecture of edge computing is shown in Figure 2. The edge network management center is located between the cloud and the terminal equipment. It can connect directly with the terminals through wireless connection and provide services to the terminals in the way of virtual machines. These intermediate-level computing centers or servers can implement different functions with different structures or scales as needed. In cloud computing, all calculations and the operations of storage are performed in the cloud computing data center. The terminal only sends requests to the cloud and receives and displays the processing results. For example, in a smart grid, the data collected by sensing devices are first transmitted to the micro-grid for processing. After that, grid data are transmitted to small-scale power stations or larger grids for further processing. Finally, the data are transferred to the cloud computing center for analysis and processing.

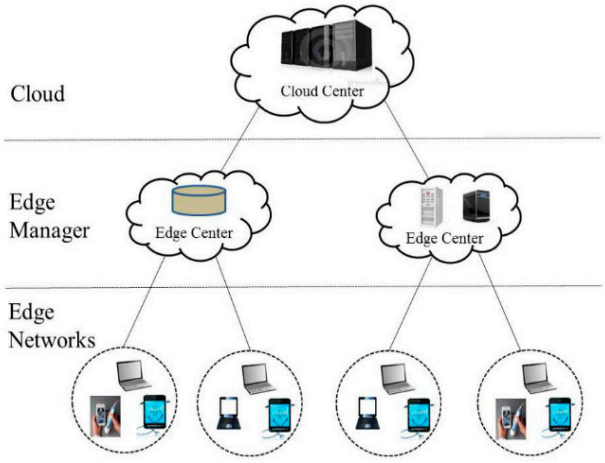

Figure 2. Architecture of edge computing.

2.3. Access Control

On the basis of the current research, the access control model in industrial internet of things can be divided into the following categories: Attribute-based access control (ABAC), role-based access control (RBAC), and access control technology based on usage control (UCON), based on capabilities and fuzzy logic [33].

Based on the RBAC and ABAC models, integrating context-related information and entity attributes, a variety of access control models based on RBAC or ABAC have emerged in IoT. The access control models combined some new features under IoT environment and improved the traditional access control models [34].

Based on UCON, many scholars have studied the access control model in the Internet of Things. UCON adopts a series of restrictions to ensure the security of access control. The access control architecture is based on a service-oriented architecture and includes the following components: Devices, services, trust management centers, and usage control access control models. The usage decision of the usage control model is determined by the attribute values of several factors such as device, service, condition, responsibility, and authorization. To ensure the security of the use control, the usage control model introduces seven sub-models related to authorization, seven responsibility-related models, and two condition-related models. However, these are still insecure in the IoT environment [35]. And they lack accurate representation of authorization process and precise definition under IoT environment.

There is also the problem that the current authorization framework such as RBAC and ABAC cannot effectively provide an easily scalable and manageable access control mechanism to provide dynamic scalability with many intelligent terminals and interactive services, which also cannot support access control requirements of distributed IoT environments. Based on the principle of layering, the resources and sensing layer can be hierarchically divided. Then, the corresponding access control scheme is designed according to the layering. Because traditional access control schemes cannot meet the security, privacy, or customized consumption and data diversity requirements of the perception layer in IoT, authors in the literature [27] propose hierarchical access control schemes. In the schemes, the user only needs to simply calculate a part of the user's single key to access the corresponding level.

Recently, Lin et al. proposed an authentication and access control scheme for industrial 4.0 based on blockchain [36,37]. The schemes utilize attribute-based digital signature and a certificateless multi-receivers encryption mechanism. However, the two cryptographic mechanisms require high computational power. Aafaf et al. achieved the anonymity and security of IoT data by deploying Fairaccess in UTXO to implement blockchain-based access control [38]. Maesa et al. released the transfer of rights in the blockchain, the access control rights are passed through the blockchain transaction, and the access resources can be easily passed [39]. However, their study only focuses on

the security of the framework and does not study the sensor capacity. So, the proposed schemes cannot be used on resource-constrained terminals.

3. Problems Statement

Due to the wide distribution and complex environment of edge computing, it was limited in computing and storage resources. A lot of applications have not realized the safety risk at the beginning of its design. A traditional measure is not fully adapted to the requirements of edge computing. Currently, edge computing meets the needs of industry digitization in agile connection, real-time service, data optimization, application intelligence, and it also faces many security challenges [40].

Typical identity management and access control mechanisms are based on a centralized trusted entity. Based on this concept of trust, each device stores all identity management and access control protocols, including protocols that are not needed by itself. The device cannot adaptively choose the protocol that it needs. In addition, the dynamic nature of the IIoT with a large number of devices will complicate the trust management of the central entity, thus affecting scalability. As the computing power of the terminal increases, there are more opportunities to bring intelligence to the terminal itself, especially in terms of security and access control logic. With the edge intelligence principle, the terminal can perform finer-grained control. However, the lack of security mechanisms can easily lead to serious consequences of misconfiguration. The edge computing system is a centralized and distributed hybrid network structure. Current access control mechanisms are either centralized or distributed. There is no access control mechanism for the edge computing architecture.

4. Identity Management and Access Control Based on Blockchain under Edge Computing for IIoT

The paper proposes a mechanism combining access control and identity management based on blockchain technology under edge computing to adapt to the IIoT.

4.1. System Architecture of IIoT

The architecture of our system is shown in Figure 3. The system consists of the following components: (1) Intelligent machines/edge network devices, (2) industrial control edge networks/intelligent control systems, (3) edge centers/management and control centers, (4) blockchain networks, and (5) cloud computing centers/manufacturing collaboration network.

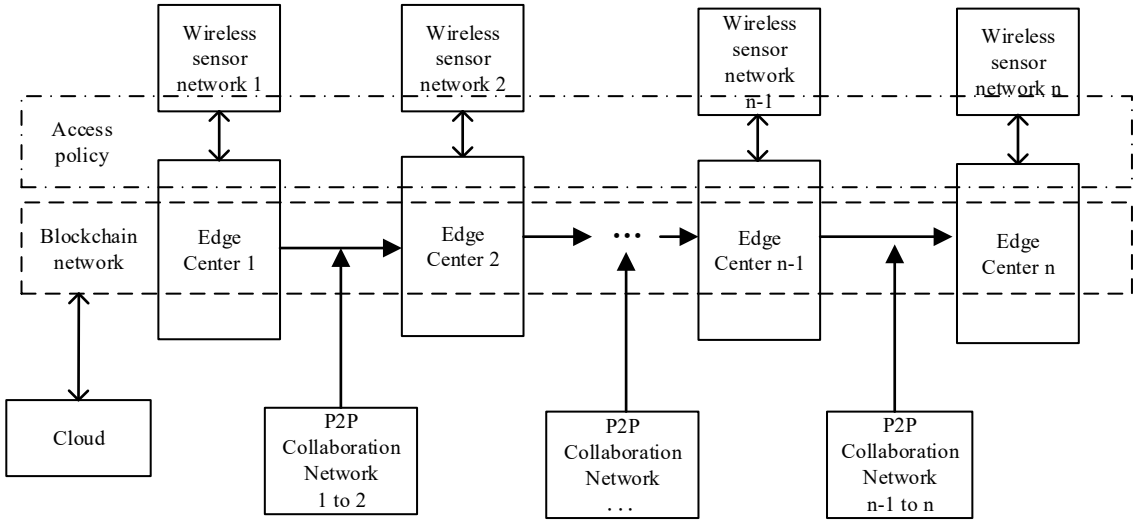

Figure 3. Identity management and access control model under edge computing.

(1) Intelligent machines/edge network devices: Various types of industrial terminals are utilized to monitor and control the manufacturing processing.

(2) Industrial control edge networks/intelligent control systems: A large number of monitoring and controlling terminals construct an edge network through the form of ad hoc networks. The edge networks can collect all kinds of manufacturing information and can process the information locally. In many cases, real-time intervention can be carried out. In addition, the industrial control edge networks can be linked to nearby edge center and receive management by the centers.

(3) Edge centers/management and control centers: Edge centers/management and control centers are the entity responsible for managing access control rights for nearby manufacturing terminals/edge network devices. These manufacturing terminals form manufacturing edge networks. At the same time, manufacturing edge networks are connected to nearby management and control centers and managed by the centers.

Edge centers/management and control centers will be assumed as the miners in the blockchain network: Storing blockchain information and validating blockchain transactions. Moreover, the manufacturing terminals with limited resources and capabilities are subject to the management of edge centers, and they are prevented from participating in the blockchain network due to limitations of their hardware. Every manufacturing terminal must be registered under the control of the edge center node.

(4) Blockchain network: A blockchain network can be constructed by multiple edge centers. The manufacturing terminals are limited to their computing power. Thus, they are not parts of a blockchain network. The blockchain can be used by the authorized entities to read but can only write by the edge centers. The edge center can use the blockchain to get a specific access control policy that suits itself. The access control information is completely decentralized and can prevent from tampering.

(5) Cloud computing center: Used to support the cyber foraging service under edge computing. The cyber foraging service is an important means to solve the shortage of computing resources of the manufacturing terminals and edge manufacturing networks. Through the cyber foraging service, the heavy computing task can be transferred to the cloud computing center.

4.2. System Setup and Operation

This section describes the establishment and operation of the hierarchical network architecture. The process can be divided into the following distinct phases: Network establishing, registering edge centers and manufacturing terminals into the system, defining access control policies for edge centers and manufacturing terminals, and changes in access control policies of nodes. The operation process is shown in Figure 4.

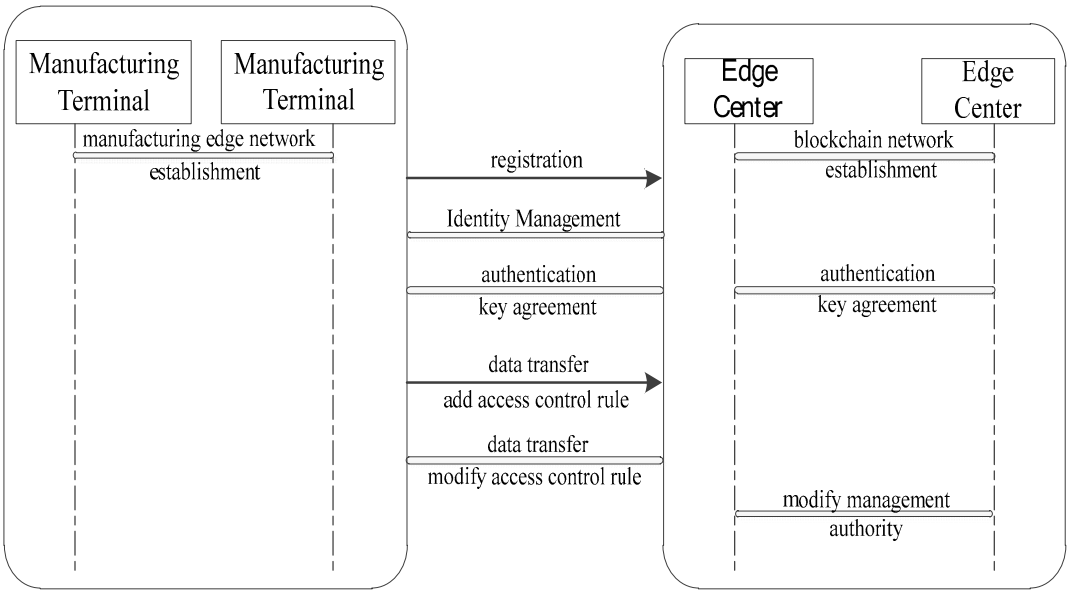

Figure 4. Setup and operation of access control with identity management.

(1) Network establishment: Each industrial control edge networks and blockchain network is constructed. The blockchain network is created by edge centers, which manages industrial control edge networks. Each manufacturing terminal and edge center has a unique identity. Once the blockchain network is established, edge centers will build access management and identity management mechanisms for manufacturing terminals in the blockchain network.

Each industrial control edge network and manufacturing terminal can be connected to an adjacent edge center, which is also a node in the blockchain network. The edge center is responsible for building and managing data access policies and identity management information, which are stored in the blockchain.

(2) Registration and authentication: The blockchain nodes need to know the address of the smart contract before registering as a miner. After the information is obtained, it can send the transaction to the feature registration manager. Thereafter, once the transaction is successfully accepted into the blockchain, the cloud computing center will receive its registered address [41].

(3) Identity management: Edge centers can register manufacturing terminals under the control of the cloud computing center. Manufacturing terminals can be registered on any nearby edge center based on their own conditions. The identity management policy of each terminal is stored in the blockchain. The terminals should be able to verify the identity before the access operation.

(4) Access policy definition: Manufacturing terminals define access control rules for their stored resources. There are several ways to define their permissions. The article uses an attribute-based access control mechanism to define a list of attributes for access rights and lists network devices that have access to specific resources.

(5) Access policy modification: The access control rights of the resources can be changed. The manufacturing terminal negotiates with the edge center to modify its access control policy. If the edge center modifies the access policy of a terminal, it will be verified and modified in the blockchain network.

(6) Modification of management rights: As mentioned earlier, manufacturing terminals must be managed by edge centers. When an edge center needs to transfer management control to another node, add, or remove other edge centers from the system, it needs to be implemented through blockchain transactions.

One advantage of our approach is that transferring administrative control of IIoT devices is a simple process, because all operations in the system are defined and implemented using blockchain transactions, and the edge centers do not need to interact with everyone.

4.3. Registration and Authentication Based on Self-Certified Public Key

Under edge computing, a large number of manufacturing sensors and controlling devices have low computing and communication capabilities and cannot perform large amounts of computation and communication. Therefore, lightweight authentication mechanisms need to be used. Thus, we use the self-certified public key-based system (SCKBS) to implement registration and authentication of network nodes [42].

The SCKBS system is similar to the identity-based public key system (IBS). The public key itself has an authentication function. Although it also depends on the key distribution center (KDC), the KDC does not directly generate user private keys, but only generates partial private keys corresponding to user identities. The user himself combines some private keys to obtain the private key.

The specific registration and certification process are as follows. Let E be the elliptic curve over the finite field Fp of order q, P be a base point of the order of the prime number n in $E(Fp)$, U and V be the edge centers in the blockchain network, H be a Hash function that maps a finite binary string to an integer set $[2,n-2]$, KDF be a key derivation function, MAC be a message authentication code, and $||$ denote a bit string connection.

Edge center node authentication: Let (q_{CA},Q_{CA}), respectively, be the private key and public key of the cloud computing center, q_{CA} be a random integer, and there is an equation of $Q_{CA} = q_{CA}P$.

The process of describing edge center U in a blockchain network to obtain its implicit certificate is as follows:

1. The node U randomly selects the integer $g_U \in [2, n-2]$, calculates $G_U = g_U P$, and sends G_U and its identity ID_U to the cloud computing center.

2. If both G_U and ID_U are valid, the cloud computing center randomly selects the integer $g_{\mathrm{CA}} \in [2, n-2]$, calculating $G_{\mathrm{CA}} = g_{\mathrm{CA}} P$ and $B_U = G_U + G_{\mathrm{CA}}$. The implicit certificate of node U is $IC_U = (Q_{\mathrm{CA}}, ID_U, B_U, t_U)$, where t_U represents the expiration time of the certificate. CA calculates $e_U = H(ID_U)$ and $s_U = g_{\mathrm{CA}} e_U + q_{\mathrm{CA}}$ mod n, then sends (s_U, IC_U) to node U.

3. After node U receives (s_U, IC_U), it computes $e_U = H(ID_U)$, $q_U = s_U + g_U e_U$ mod n, and $Q_U = q_U P$. (q_U, Q_U) is the long-term private key and public key of node U, then U verifies whether equation $Q_U = e_U B_U + Q_{\mathrm{CA}}$ holds. If the above equation holds, then node U considers the implicit certificate IC_U to be correct.

As a result, each edge center can perform authentication registration and obtain its public and private key pairs and corresponding implicit certificates. The implicit certificates of every node are stored in blockchain, and they are managed by cloud computing centers or edge centers.

4.4. Certificate and Identity Management Based on Blockchain

In our system, each entity has an identity. Each identity corresponds to a unique implicit certificate. The identity and certificate of entity in our system are one-to-one correspondence. Moreover, the identity of entity is bound to its implicit certificate. In our system, the identity and certificate are stored in blockchain network, and they are defined as merkle hash trees of blockchain, and their integrity is regularly verified. They can only be stored on the appended form. Thus, it ensures that identity and certificates will not be deleted or modified after being appended. The identity, certificate, and the history of the published key of entity in the network will be archived publicly in the blockchain. If a fraud certificate or fake key is issued, they will soon be detected. The edge centers collect certificate information and verify certificate transactions of other entities in the blockchain. Each edge center collects blocks from other edge centers, validate blocks, and attaches valid blocks to their local copies to make the blockchain longer.

The timestamp service is an essential component in identity and certificate management. When issuing and storing certificates and public keys, the corresponding timestamp is provided. The nodes in the blockchain network retain local copies of identities and certificates. Before a secure session among the manufacturing terminal, the edge center, and cloud computing center, the certificate of each party will be verified according to the local copy of the blockchain. The certificates can be obtained directly from the local blockchain.

Each network node must ensure that its certificate is valid. That is, its certificates need to be updated regularly. When a network entity performs a dishonest act and is discovered, its certificate will be revoked. The edge center initiates a transaction that revokes the certificate of a malicious network entity and broadcasts the transaction. After validation by most nodes of blockchain, the transaction is recorded in the blockchain. It also means that the malicious node is removed from the edge network and its identity is invalid. When the unexpired certificate is revoked, the corresponding certificate revocation list (CRL) file will be included in the transaction. The certificate revocation operation is also recorded. Because the blockchain has the characteristics of anti-counterfeiting, non-tampering, and is easy to use, intelligent contracts to realize, certify, and identity revocation have good transparency and credibility. The storage of identities and certificates in the blockchain is shown in Figure 5.

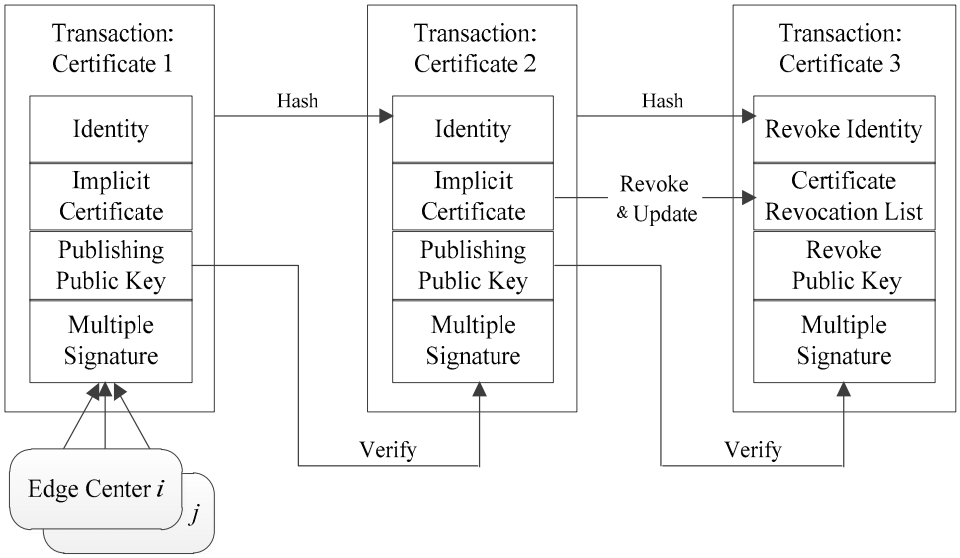

Figure 5. Identity and certificate management based on blockchain.

Due to blockchain, the trust relationship of edge network is established in this system. As new terminals join the network, they can become trust anchors. That is, more terminals are allowed to connect the manufacture network. The trust will evolve to support growth of the decentralized network. The network terminal must first verify the identity of the communicator node, ensure that it is a valid node, and then carry out the corresponding access services.

4.5. Lightweight Key Exchange Based on Self-Certified Public Key

Similar to identity-based key system, the scheme in SCKBS does not need a certificate to guarantee the reliability of public key. Moreover, the public key itself has a self-authentication function. Although the scheme of SCKBS also relies on key distribution center (KDC), KDC does not directly generate the private key of network terminals. KDC only produces a part of the private key corresponding to the terminals' identity. The terminals get the actual private key, which effectively solves the key escrow problem. In addition, SCKBS is based on elliptic curve cryptography, which can reduce computation and communication and improve execution efficiency. Therefore, it is particularly suitable for resource-constrained edge networks. The proposed scheme is as follows: $H_1: \{0,1\}^* \to Z_q$ and $H_2: \{0,1\}^* \to \{0,1\}^\lambda$ are two Hash functions. For each party $\hat{A}$, its two long-term private keys are $a_1, a_2 \in Z_q$, and the corresponding long-term public key is $A_1 = a_1 P, A_2 = a_2 P$. Let $\hat{A}$and $\hat{B}$ be two participants in the agreement and each have two long-term public and private key pairs.

(1) $\hat{A}\hat{A}$ selects the short-term private key $\tilde{x} \in \{0,1\}^\lambda$, calculates $x = H_1(\tilde{x}, a_1, a_2)$ and the transient public key $X = xP$, destroys x, and sends $(\hat{B}, \hat{A}, X)$ to $\hat{B}\hat{B}$.

(2) After receiving $(\hat{B}, \hat{A}, X)$, $\hat{B}\hat{B}$ verifies X. $\hat{B}$ selects the short-term private key $\tilde{y} \in \{0,1\}^\lambda$, calculates $y = H_1(\tilde{y}, b_1, b_2)$ and the short-term public key $Y = yP$, destroys y, and sends $(\hat{A}, \hat{B}, X, Y)$ to $\hat{A}$. $\hat{B}$ calculates $Z_1 = (y + b_1)(X \cdot A_1)$, $Z_2 = (y + b_2)(X \cdot A_2)$, $Z_3 = yX$, and $SK = H_2(Z_1, Z_2, Z_3, X, Y, \hat{A}, \hat{B})$. $\hat{B}$ uses SK as its session key.

(3) After receiving $(\hat{A}, \hat{B}, X, Y)$, $\hat{A}$ verifies Y . $\hat{B}$ calculates $Z_1 = (x + a_1)(Y \cdot B_1)$, $Z_2 = (x + a_2)(Y \cdot B_2)$, $Z_3 = xY$, and $SK = H_2(Z_1, Z_2, Z_3, X, Y, \hat{A}, \hat{B})$. $\hat{A}$ uses SK as its session key.

Theorem 1. *If H_1 and H_2 are random oracles, and ECDH is assumed to be true for group E(Fp), the proposed protocol is eCK model-safe.*

eCK Model Security.

The eCK model is an important tool to analyze protocol. It consists of a set of protocol participants. For the security parameter k, the advantage $Adv_{M,\Sigma}^{AKE}(k)$ of the PPT attacker m attack protocol Σ is

$Adv_{M,\Sigma}^{AKE}(k) = |Succ_{M,\Sigma}^{AKE}(k) - \frac{1}{2}|$, $Succ_{M,\Sigma}^{AKE}(k)$ is the probability of the bit $\hat{b} = b$ output by the attacker when making a test query on the fresh oracle. If the advantage $Adv_{M,\Sigma}^{AKE}(k)$ is negligible for any PPT attacker M, then the protocol Σ is secure.

Proof. Let λ be the security parameter and M be the attacker of the protocol. If there is such an M to win the indistinguishable game with a non-negligible probability, that is, after the Test query, the value of b is correctly guessed. Let us illustrate how to construct a simulator S that uses M to solve the ECDH problem with a non-negligible advantage. (U, V) is the ECDH challenge of S. Among that, $U, V \in E(F_p)$, whose task is to compute ECDH == D.

Ⅰ. Forgery attack: At a certain moment, M calculates Z_1,Z_2,Z_3, and does a query to H_2 with pair $(Z_1, Z_2, Z_3, X, Y, \hat{A}, \hat{B})$.

Ⅱ. Key duplication attack: Makes two mismatched sessions calculate the same session key value and obtains this session key value by querying another session key value.

The input of the function is the script record of the session. Because it is a random language machine, and each session has a different temporary private key, the probability that two non-matching sessions have the same participants and the same transient public key can be ignored. Therefore, the probability of successfully performing a key replication attack is negligible.

Below, we mainly analyze counterfeit attacks. We define three conditions that exist when forging attacks:

Case 1: There is an honest entity$\hat{B}$, that does the query $H_1(\tilde{y}, b_1, b_2)$ before doing a Static-Key Reveal ($\hat{B}$), query (Static-Key Reveal ( ) query is not required at this time).

Case 2: Case 1does not occur. The test session has a matching session.

Case 3: Case 1 does not occur. The test session does not have a match in session.

If Case 1, Case 2, or Case 3 occurs with a non-negligible probability, a counterfeit attack will be successfully performed with non-negligible advantages. We will analyze these three situations separately. □

Case 1. *In the $n(\lambda)$ party, randomly select one party $\hat{B}$ and set its long-term public key to $B_1 = V$, $B_2 = \frac{sP}{rV}$, where $s, r \in Z_q$. At this point, S does not know the long-term private key of $\hat{B}$, and the remaining long-term private key of $n(\lambda) - 1$ is randomly assigned. When M queries all parties except $\hat{B}$, because S knows its long-term private key, it can answer correctly. Below we will explain how to simulate $\hat{B}$, as well as other oracles such as H_1 and H_2 queries.*

(a) Simulation of ($\hat{B}$). When the incoming message is $(\hat{B}, \hat{A})$ or $(\hat{B}, \hat{A}, X)$, select $\tilde{y} \in \{0,1\}^{\lambda} y \in Z_q$, calculate $Y = yP$, generate a new session identifier and return a message: For 1 $(\hat{B}, \hat{A})$, the identifier is 1 $(\hat{B}, \hat{A}, Y, X)$, return $(\hat{A}, \hat{B}, Y)$, for $(\hat{B}, \hat{A}, X)$, the identifier is$(\hat{B}, \hat{A}, Y, X)$, return $(\hat{A}, \hat{B}, X, Y)$. If the incoming message is $(\hat{B}, \hat{A}, Y, X)$, check if S has a session $(\hat{B}, \hat{A}, Y, X)$. Terminate if there is no, otherwise the updated ID is $(\hat{B}, \hat{A}, Y, X)$.

(b) $H_1(\tilde{y}, b_1, b_2)$. S checks if it is $b_1 P = V$ or $b_2 P = s\frac{P}{rV}$. If yes, S solves the ECDH problem for $b_1 P = V$, $ECDH(U, V) = b_1 U$ and $b_2 P = s\frac{P}{rV}$, $ECDH(U, V) = (b_2 - s) rU$, respectively. All other cases return normally (for new queries, a random value of Z_q is returned, and a consistent value is returned for the query that was done).

(c) Ephemeral-Key Reveal (sid): For the session sid, S return the short-term private key$\tilde{y}$.

(d) Static-Key Reveal ($\hat{A}$): If $\hat{A} = \hat{B}$, S terminates. Otherwise, the two long-term private keys of $\hat{A}$ returned normally.

(e) Session-Key Reveal(sid): S return the session key SK_{sid} as follows:

Ⅰ. If sid is not owned by $\hat{B}$, $(Z_1, Z_2, Z_3, X, Y, \hat{A}, \hat{B})$ and $SK_{sid} = H_2(Z_1, Z_2, Z_3, X, Y, \hat{A}, \hat{B})$ can be calculated by S.

Ⅱ. If the sid is owned by $\hat{B}$, Scheck whether the previous session has been queried. If yes, the previous session key SK_{sid} is returned.

Ⅲ. Otherwise, set $sid = (\hat{B}, \hat{A}, Y, X)$, then SK_{sid}=$H_2(ECDH(Y \cdot B_1, X \cdot A_1), ECDH(Y \cdot B_2, X \cdot A_2), Z_3, X, Y, \hat{A}, \hat{B})$ Due to the ECDH problem in the parameters, S cannot directly answer to avoid the attacker to

distinguish between the simulated environment and the real environment. S *calculate* $\overline{Z_1} = \frac{Z_1}{(y(X\cdot A_1)\cdot a_1 B_1)}$, $\overline{Z_2} = \frac{Z_2}{(y(X\cdot A_2)\cdot a_2 B_2)}$, *by verifying if* $r\overline{Z_1 Z_2} = sX$, *determine whether* Z_1, Z_2 *is generated correctly. If correctly generated, if and only if* Z_1, Z_2 , $Z_1 = (x + a_1)(Y \cdot B_1), Z_2 = (x + a_2)(Y \cdot B_2)$ *is equivalent to* $\overline{Z_1} = xb_1 P, \overline{Z_2} = xb_2 P$. *In addition,* S *check whether* $Z_3 = yX$. *If both pass, the value of the query* ($Z_1, Z_2, Z_3, X, Y, \hat{A}, \hat{B}$) *for* H_2 *will be the return as* SK_{sid}. *In other cases, a random value* SK_{sid} *is returned to* M.

(f) H_2*Query: S returns the value of* $H_2(\cdot)$ *as follows:*

Ⅰ. S *check if the same query was made for* H_2 *before. If yes, the previous H2(.) value is used as the value of the query.*

Ⅱ. If the input is $(Z_1, Z_2, Z_3, X, Y, \hat{A}, \hat{B})$, S *check if Session Key Reveal(.) has done a query with sid b. If so, s calculates*$\overline{Z_1} = \frac{Z_1}{(y(X\cdot A_1)\cdot a_1 B_1)} = xb_1 P$, $\overline{Z_2}$ =*and judges whether or not* $r\overline{Z_1 Z_2} = sX$. *If the above formula is true, the return value of Session-Key Reveal(.) is used as the value of this* $H_2(\cdot)$*query.*

Ⅲ. In other cases, S *select a random value to return.*

The probability of a trapdoor test error is at most $\frac{2s(\lambda)h_2(\lambda)}{q}$. *If every trapdoor test is correct, then* S *perfectly simulates the* M *environment until* M *has done a Static-Key Reveal* ($\hat{B}$) *query.* S *sets* $(V, \frac{sP}{rV})$ *to the public key of*$\hat{B}$ *with a probability of at least*$\frac{1}{n}(\lambda)$, *where* $\hat{B}$ *is an honest entity and* M *queries* $H_1(*, b_1, b_2)$ *for it before doing Static-Key Reveal*($\hat{B}$). *The probability of* S*success is the following:*

$$Pr(S) \geq \frac{P_1(\lambda)}{n(\lambda)} - \frac{2s(\lambda)h_2(\lambda)}{q}$$

where $P_1(\lambda)$ *is the probability of occurrence of Case1. (1).*

Case 2 M *Randomly select two sessions* sid *and* sid^*, *assumed to be owned by* $\hat{A}$ *and* $\hat{B}$, *respectively. Suppose* M *chooses one for the test session and the other for the matching session. For sessions other than* sid *and* sid^*, S *is simulated normally. For these two sessions,* S *selects the short-term private key* $\tilde{x}, \tilde{y} \in \{0,1\}^\lambda$, *and sets the short-lived public key to* U *(instead of* $H_1(\tilde{x}, a_1, a_2)P$*) and* V *(instead of* $H_1(\tilde{y}, b_1, b_2)P$*). Set the same random key value* SK *for both sessions.*

The attacker chooses one of sid *and* sid^* *as the test session with a probability of at least*$\frac{2}{s(\lambda)^{2'}}$, *and the other is the matching session. If* M *win a forgery attack, it must query* H_2 *for*$(*,*$,$ECDH(U, V), X, Y, \hat{A}, \hat{B})$, *and* S *can solve the ECDH problem. What we need to consider is whether* M *can distinguish between real and simulated environments. If it can distinguish, it must do a* $(\tilde{x}, a_1, a_2)$ *or* $(\tilde{y}, b_1, b_2)$ *query to* H_1. *Since sid is a fresh session,* M *cannot simultaneously query the long-term and short-term private keys of* $\hat{A}$ *(or* $\hat{B}$*). If* M *queries the short-term private key (or) via Ephemeral-Key Reveal(.), will not be able to query (or) for, because in Case 1 we exclude this. Without the Ephemeral-Key Reveal(.) query, cannot get information about (or) because (or) is only used in one session. The probability of success for S is the following:*

$$Pr(S) \geq \frac{2P_2(\lambda)}{(s(\lambda)^2 h_2(\lambda))}$$

where$P_2(\lambda)$ *is the probability of occurrence of Case 2. (2).*

Case 3 S *randomly selects two participants* $\hat{A}$, $\hat{B}$ *and one session sid.* S *sets* $B_1 = V$ *and* $B_2 = \frac{sP}{rV}$ *as the long-term public key of* $\hat{B}$, *and the corresponding long-term private key that* S *does not know.* S *randomly assigned for the long-term private key of the remaining*$n(\lambda) - 1$ *party. With at least* $\frac{1}{n(\lambda)}^2$ *probability,* M *selects* $\hat{A}$ *and* $\hat{B}$, *respectively, as the owner and counterpart of the session sid, at the same time, selecting sid as the test session. When* S *has the long-term private key of the participant, which is activated by* M, S *can simulate normally, but the test session is an exception. For the test session,* S *selects* $\tilde{x} \in_R \{0,1\}^\lambda$ *and sets the short-term public key* X *to* U, *selecting the random value SK as the key value for the test session. The simulation for* $\hat{B}$ *is similar to Case 1.*

The test session M must query H_2 for a 7-tuple of $(Z_1^*, Z_2^*, Z_3^*, X, Y, \hat{A}, \hat{B})$. While $Z_1^* = (x + a_1)(Y^* \cdot B_1) = (u + a_1)(Y^* \cdot V)$, $Z_2^* = (x + a_2)(Y^* \cdot B_2)$ and $X = U$ are the message sent by the test session and B is the message from the attacker.

The test session M must query H_2 for a 7-tuple of $(Z_1^*, Z_2^*, Z_3^*, X, Y, \hat{A}, \hat{B})$. While $Z_1^* = (x + a_1)(Y^* \cdot B_1) = (u + a_1)(Y^* \cdot V)$, $Z_2^* = (x + a_2)(Y^* \cdot B_2) = (u + a_2)(Y^* \cdot \frac{sP}{rV})$ and $X = U$ are the message sent by the test session, Y^* is the message from the attacker. Knowing s, r, M calculates $\overline{Z_1} = Z_1^*/a_1(Y^* \cdot V) = u(Y^* \cdot V)$, $Z = (-1/(1 + r))(\frac{\overline{Z_2}}{(\overline{Z_1} \cdot sU)}) = vU = uV$ and $\overline{Z_2} = Z_2^*/a_2(Y^* \cdot \frac{sP}{rV}) = u(Y^* \cdot \frac{sP}{rV})$. Due to freshness requirements, M cannot do Static-Key Reveal ($\hat{B}$) query and cannot query the long-term private key and the short-term private key of $\hat{A}$ at the same time. At this point, there is the only one way for M to distinguish between the real environment and the simulated environment, that is doing $(\tilde{x}, a_1, a_2)$ query to H_1 and then verifying whether $U = H_1(\tilde{x}, a_1, a_2)P$.

The probability of this event is $(k + h_1(\lambda))/2^\lambda - 1$. In this case, the following is the probability of the success of S

$$P_{r(s)} \geq \frac{1}{s(\lambda)n(\lambda)^2 h_2(\lambda)} P_3(\lambda)\left(1 - \frac{k + h_1(\lambda)}{2^\lambda - 1}\right) - \frac{2s(\lambda)h_2(\lambda)}{q}$$

where$P_3(\lambda)$ is the probability of the occurrence of Case 3. (3).

Comprehending the formula of (1)–(3), and let

$$\Xi_1 = \frac{P_1(\lambda)}{n(\lambda)} - \frac{2s(\lambda)h_2(\lambda)}{q}$$

$$\Xi_2 = \frac{2P_2(\lambda)}{s(\lambda)^2 h_2(\lambda)}$$

$$\Xi_3 = \frac{P_3(\lambda)}{s(\lambda)n(\lambda)^2 h_2(\lambda)}\left(1 - \frac{K + h_1(\lambda)}{2^\lambda - 1} - \frac{2s(\lambda)h_2(\lambda)}{q}\right).$$

Then, $\Pr(S) \geq max\{\Xi_1, \Xi_2, \Xi_3\}$ is the success of S.

If PPT attacker M implements any of the above attacks with a non-negligible advantage, then we can successfully solve the ECDH problem with a non-negligible advantage, which contradicts the ECDH assumption, so the protocol is eCK-safe.

4.6. Combining Identity Management and Access Policy

The edge network terminals of IIoT will define its own access control policy and pass the access control policy to the edge centers through the session key. The edge center stores the access policy (AP) of the network nodes' resource in the blockchain. After the AP is stored in the blockchain, the edge centers can control access to the terminal resources according to the policy.

Generally, an access policy of a network terminal resource includes: A condition of an entity ID that is authorized to access; a resource that is allowed to access. That is, in our scenario, the resource owner decides which object is authorized to access, and the conditions that the object must have to grant access. A blockchain network can be thought of as a distributed database managed by all edge center nodes. This means that each access policy added to the blockchain cannot be subsequently removed. So, the blockchain-based access control strategy has the characteristics of irreversible modification and traceability. Stored on the blockchain is the access strategy of the manufacturing data, not the data itself, thus greatly reducing the need to store data in the blockchain.

The edge center completely controls the identity management of network terminals. Each network terminal can select the credentials they wish to share with their own attribute credentials. The terminal may choose to store these attributes on the blockchain, and use the identity information stored on the blockchain to identify the correct network terminal. Using attribute-based credentials allows users to display only the credentials they choose.

Data hash table is often used in existing access control schemes. However, Bloom filters have the advantage of space and time, and their space advantage is that their volume does not increase with

increasing elements. The time advantage is that it does not rely on the loop structure to determine if an element is in the collection [43]. Thus, the bloom filter is utilized in our scheme. The details are shown in the Figure 6.

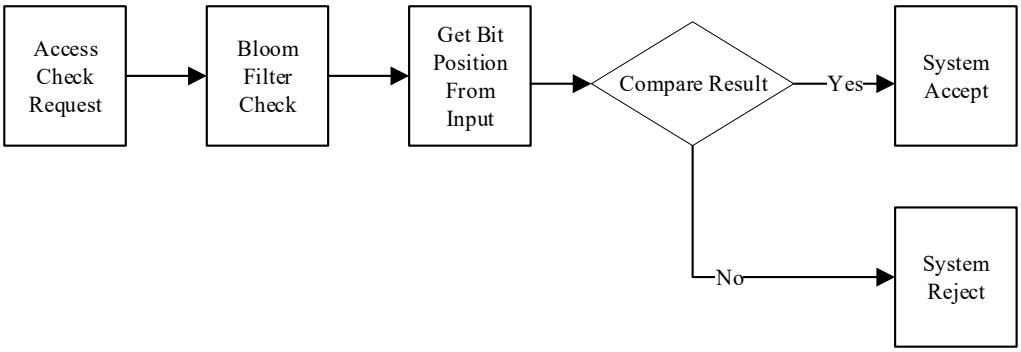

Figure 6. Flow diagram of the access control based on bloom filter.

In our system, the blockchain network is utilized to represent the transfer permissions of access resources from one edge center to another. Specifically, we recommend storing the representation of the right to access resources in the blockchain, allowing this right to be managed through the transaction among the edge center in the blockchain network. The main advantages of the method are as follows. The access rights of resources are initially defined by the owners of resources (manufacturing terminals) and are stored in the blockchain through policy creation transactions (PCT) of blockchain. Access rights to resources can easily be transferred from one edge center to another without the intervention of the edge network terminals. Therefore, any edge center can check the operator's authority.

The access right can be transferred from the current holder (such as Oi) to another Oj through right transfer transaction (RTT) in the blockchain. RTT is created by Oi, so no intervention of network terminals is required in any transfer of rights. When passing rights through RTT, Oi can only modify the variable conditions that regulate their rights by restricting them. In addition, policy updates can be done through policy update transaction (PUT). Because all transactions have timestamps, these changes are obvious and traceable.

4.7. Performance Evaluation

In this section, we evaluate the performance of the proposed schemes. We use Ubuntu 16.04 (Canonical Ltd., London, UK) and C language to construct the schemes. The device's identity of IIoT in our scheme is underpinned by the interactions between Ethereum smart contracts. Symmetric encryption and message authentication coding are 128 bits. The energy consumption of nodes sending information is 1.62 mJ/bit, and the energy consumption of receiving information is 2.025 mJ/bit. So, the sending message and received message of a terminal are one pass and 367 bits. The energy of communication is 1337.715 mJ.

The smart contracts of identity management and access control are published in the edge center. Because of the strong capabilities of the edge center, the experiment used a 2.3 GHz Intel Core i7 (8 core) (i7 9900t; Intel, Santa Clara,CA, USA) and 16 GB of RAM memory capacity (DDR4; SAMSUNG, Gyeonggi Province, Korea). Raspberry Pi 3 (raspberry pi foundation, Cambridge, UK) is utilized as terminals of IIoT. We use the edge terminal to issue 50 requests per second to test whether the system can achieve stable throughput. All concurrent clients request access to data resources from the edge center node. The latency between the edge center and a terminal limits the overall performance of the network.

The results show that the edge center with identity management and access control performs best under up to 100 concurrent terminals, with 90 requests per second without timeout. The Fairaccess solution pays more attention to privacy and integrity, while our solution focuses more on sensor capacity and unit time to get more effective data. So, we use a lightweight encryption

mechanism. Compared with the ordinary edge center, the edge center with identity management and access control can more efficiently acquire the effective data needed by the system and filter some unnecessary data, which can improve the node capacity of the sensor network and process more valid data in unit time. We believe that the performance of the edge center is acceptable because the node has various resources and powerful computing, storage, and communication capabilities. Despite the limited capabilities of the terminal, bottlenecks still occur in the case of an excessive number of nodes, but experiments have shown that terminals with identity management and access control are more scalable than terminals without these features.

5. Conclusions

The industrial Internet of things needs stronger security solutions. In this paper, a system model of identity management and access control for IIoT under edge computing is firstly constructed, and the roles of the entities and the corresponding scope of authority are clarified. Secondly, it describes the establishment and operation of the identity management and access control in the edge network. Also, a lightweight registration and certification agreement of intelligent terminals, lightweight key exchange protocol, and access control mechanism are integrated to provide security guarantees such as authentication, auditability, and confidentiality for IIoT.

Future research will evaluate various attribute-based access control mechanisms and optimize their performance to our system. Based on blockchain, a key agreement protocol will be designed and integrated into the proposed system.

Author Contributions: Conceptualization, Y.R.; F.Z. and J.Q.; Funding acquisition, Y.R.; J. W.; Investigation, Y.R.; F.Z. and J.Q.; Software, J.Q.; Writing—original draft, Y.R.; F.Z. and J.Q.; Writing—review and editing, Y.R.; J.W.; A.K.S.

Funding: This work was supported by the NSFC [61772280, 61772454, 61811530332, 61811540410], and the PAPD fund from NUIST.

Conflicts of Interest: Page: 14

The authors declare no conflict of interest.

References

1. Xu, L.; He, W.; Li, S. Internet of Things in Industries: A Survey. *IEEE Trans. Ind. Inform.* **2014**, *10*, 2233–2242.
2. Chen, C.; Lin, M.; Liu, C. Edge computing gateway of the industrial Internet of Things using multiple collaborative microcontrollers. *IEEE Netw.* **2018**, *38*, 24–32.
3. Steiner, W.; Poledna, S. Fog computing as enabler for the Industrial Internet of Things. *Elektrotech. Inf.* **2016**, *133*, 310–314.
4. Yin, C.Y.; Xi, J.W.; Sun, R.X.; Wang, J. Location Privacy Protection based on Differential Privacy Strategy for Big Data in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3628–3636.
5. Kim-Kwang, R.C.; Stefanos, G.; Jong, H.P. Cryptographic solutions for industrial Internet-of-Things: Research challenges and opportunities. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3567–3569.
6. Lian, J.; Zhang, H.; Zhang, Y.; Zhang, Y. Innovative Conception of Industrial Internet of Things. *Process Autom. Instrum.* **2018**, *39*, 39–42.
7. Wang, J.; Cao, J.Y.; Simon, S.R.; Park, J.H.; An improved ant colony optimization-based approach with mobile sink for wireless sensor networks. *J. Supercomput.* **2018**, *74*, 6633–6645.
8. Wang, W.; Deng, Z.; Wang, J. Enhancing Sensor Network Security with Improved Internal Hardware Design. *Sensors*, **2019**, *19*, 1752.
9. Zhao, Z.; Liu, F.; Cai, Z.; Xiao, N. Edge computing: Platforms, applications and challenge. *J. Comput. Res. Dev.* **2018**, *55*, 327–337.
10. Wang, J.; Ju, C.W.; Gao, Y.; Sangaiah, A.K.; Kim, G.J. A PSO based Energy Efficient Coverage Control Algorithm for Wireless Sensor Networks. *Comput. Mater. Contin.* **2018**, *56*, 433–446.
11. Hesham, E.; Sharmi, S.; Mukesh, P.; Deepak, P.; Akshansh, G.; Manoranjan, M.; Lin, C. Edge of Things: The big picture on the Integration of edge, IoT and the cloud in a distributed computing environment. *IEEE Access* **2017**, *6*, 1706–1717.

12. Shirazi, S.N.; Gouglidis, A.; Farshad, A.; Hutchison, D. The extended cloud: Review and analysis of mobile edge computing and fog from a security and resilience perspective. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 2586–2595.
13. Wang, J.; Gao, Y.; Yin, X.; Li, F.; Kim, H. An Enhanced PEGASIS Algorithm with Mobile Sink Support for Wireless Sensor Networks. *Wirel. Commun. Mobile Comput.* **2018**, *2018*,9472075.
14. Zhou, Y.; Zhang, D. Near-end cloud computing: Opportunities and challenges in the post-cloud computing era. *Chin. J. Comput.* **2019**, *42*, 677–700
15. Abbas, N.; Zhang, Y.; Taherkordi, A.; Skeie, T. Mobile edge computing: A survey. *IEEE Internet Things J.* **2018**, *5*, 450–465.
16. Gusev, M.; Dustdar, S. Going back to the roots—The evolution of edge computing, an IoT perspective. *IEEE Internet Comput.* **2018**, *22*, 5–15.
17. Wang, J.; Cao, Y.Q.; Li, B.; Kim, H.J.; Lee, S. Particle Swarm Optimization based Clustering Algorithm with Mobile Sink for WSNs. *Future Gener. Comput. Syst.* **2017**, *76*, 452–457.
18. Tirkolaee, E.B.; Hosseinabadi, A.A.R.; Soltani, M.; Sangaiah, A.K.; Wang, J. A Hybrid Genetic Algorithm for Multi-trip Green Capacitated Arc Routing Problem in the Scope of Urban Services. *Sustainability* **2018**, *10*, 1366.
19. Wang, J.; Cao, J.Y.; Ji, S.; Park, H.J. Energy Efficient Cluster-based Dynamic Routes Adjustment Approach for Wireless Sensor Networks with Mobile Sinks. *J. Supercomput.* **2017**, *73*, 3277–329.
20. Gao, Y.; Wang, J.; Wu, W.; Sangaiah, A.K.; Lim, S. A Hybrid Method for Mobile Agent Moving Trajectory Scheduling using ACO and PSO in WSNs. *Sensors* **2019**, *19*, doi:10.3390/s19030575.
21. Pan, J.; McElhannon, J. Future Edge Cloud and Edge Computing for Internet of Things Applications. *IEEE Internet Things J.* **2018**, *5*, 439–449.
22. He, D.; Chan, S.; Guizani, M. Security in the IoT supported by mobile edge computing. *IEEE Commun. Mag.* **2018**, *56*, 56–61.
23. Humayed, A.; Lin, J.; Li, F.; Luo, B. Cyber-Physical Systems Security—A Survey. *IEEE Internet Things J.* **2017**, *4*, 1802–1831.
24. Tschorsch, F.; Scheuermann, B. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Commun. Surv. Tutor.* **2018**, *18*, 2084–2123.
25. Wang, J.; Gao, Y.; Liu, W.; Sangaiah, A.K.; Kim, H.J. An Improved Routing Schema with Special Clustering using PSO Algorithm for Heterogeneous Wireless Sensor Network. *Sensors* **2019**, *19*, doi:10.3390/s19030671.
26. Ren, Y.; Liu, Y.; Ji, S.; Sangaiah, A.K.; Wang, J. Incentive Mechanism of Data Storage Based on Blockchain for Wireless Sensor Networks. *Mob. Inf. Syst.* **2018**, *2018,* doi:10.1155/2018/6874158.
27. Kyusakov, R.; Eliasson, J.; Delsing, J. Integration of wireless sensor and actuator nodes with it infrastructure using service-oriented architecture. *IEEE Trans. Ind. Inform.* **2013**, *9*, 43–51.
28. Zeng, D.; Dai, Y.; Li, F.; Sherratt, B.; Wang, J. Adversarial learning for distant supervised relation extraction. *CMC Comput. Mater. Contin.* **2018**, *55*, 243–254.
29. Alrawais, A.; Alhothaily, A.; Hu, C.; Cheng, X. Fog computing for the Internet of Things: Security and privacy issues. *IEEE Internet Comput.* **2017**, *21*, 34–42.
30. Satyanarayanan, M. The emergence of edge computing. *IEEE Comput.* **2017**, *50*, 30–39.
31. Wang, J.; Gao, Y.; Liu, W.; Wu, W.; Lim, S.J. An Asynchronous Clustering and Mobile Data Gathering Schema based on Timer Mechanism in Wireless Sensor Networks. *Comput. Mater. Contin.* **2019**, *58*, 711–725.
32. Ouaddah, A.; Mousannif, H.; Elkalam, A.; Ouahman, A. Access control in the Internet of Things: Big challenges and new opportunities. *Comput. Netw.* **2017**, *112*, 237–262.
33. Vahid, R.; Karimi, P.; Alencar, C.; Donald, D. A formal modeling and analysis approach for access control rules, policies, and their combinations. *Int. J. Inf. Secur.* **2017**, *16*, 43–74.
34. Kumar, D.; Ashish, M.; Sagar, P. A novel proxy signature scheme based on user hierarchical access control policy. *J. King Saud Univ.* **2013**, *25*, 219–228.
35. Cirani, S.; Picone, M.; Gonizzi, P. Iot-OAS: An OAuth-based authorization service architecture for secure services in IoT scenarios. *J. Sens.* **2015**, *15*, 1224–1234.
36. Lin, C.; He, D.; Huang, X.; Choo, K.; Athanasios, K. BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0. *J. Netw. Comput. Appl.* **2018**, *116*, 42–52.
37. Konstantions, C.; Michael, D. Blockchains and smart contracts for the Internet of Things. *IEEE Access* **2016**, *4*, 2292–2303.

38. Aafaf, O.; Anas, A.E.; Abdellah, A.O. FairAccess: A new Blockchain-based access control framework for the Internet of Things. *Secur. Commun. Netw.* **2016**, *9*, 5943–5964.
39. Maesa, D.D.F.; Mori, P.; Ricci, L. Blockchain Based Access Control. *Distrib. Appl. Interoper. Syst.* **2017**, *10320*, 206–220.
40. Novo, O. Blockchain meets IoT: An architecture for scalable access management in IoT. *IEEE Internet Things J.* **2018**, *5*, 1184–1195.
41. Ren, Y.J.; Leng, Y.; Cheng, Y.P.; Wang, J. Secure data storage based on blockchain and coding in edge computing. *Math. Biosci. Eng.* **2019**, *16*, 1874–1892. doi:10.3934/ mbe.2019091.
42. Tsaur, W. Several security schemes constructed using **ECC**-based self-certified public key cryptosystems. *Appl. Math. Comput.* **2005**, *168*, 447–464.
43. Hieb, J.; Schreiver, J.; Graham, J. Using bloom filters to ensure access control and authentication requirements for SCADA field devices. *Crit. Infrastruct. Prot.* **2012**, *390*, 85–97.