Anonymous Message Authentication Scheme for Semitrusted Edge-Enabled IIoT

Jie Cui, Fengqun Wang, Qingyang Zhang, Yan Xu, and Hong Zhong

***Abstract*—As internet of things and other technologies are widely used in industrial manufacturing, automation and intelligence have witnessed rapid developments, resulting in the proposal of the industrial internet of things (IIoT). However, the IIoT still faces various network security threats; hence, data integrity, confidentiality, and anonymity need to be ensured. The use of cloud and edge servers as semitrusted third parties often results in the leaking of privacy sensitive user data. Meanwhile, existing security schemes treat the cloud and edge as fully trusted entities, which is not always valid. Considering edge servers as semitrusted entities, we propose a novel message authentication scheme that leverages group signature technology and proxy reencryption technology to ensure data integrity, confidentiality, and anonymity. Through theoretical analysis and performance comparison, we prove the security of our scheme. In addition, we implement our scheme on a real publish/subscribe system, and the experimental results show the feasibility of our scheme.**

***Index Terms*—Group signature, industrial internet of things (IIoT), proxy reencryption, publish/subscribe service, security.**

I. Introduction

IN RECENT years, the industrial internet of things (IIoT) [1]–[3] has attracted significant attention owing to its advantages of intelligence, automation, real-time, and interconnection. The IIoT is the combination of new technologies, such as edge computing [4] and machine learning [5], with intelligent devices, such as sensors and controllers, through various communication technologies to realize intelligent industrial manufacturing. In the IIoT environment, an intelligent device first perceives data and then transmits it to a cloud server. Subsequently, the cloud server stores, analyzes, and processes the received data [6], [7] and sends the processed data to the end-user. In the process of data generation and transmission, real-time requirements are essential [8]. For example, some industrial processes for intelligent manufacturing must be fed back within 60 μs [9]. However, uploading data to the cloud consumes a significant amount of bandwidth and causes response latency. Therefore, some researchers have proposed edge computing [10]–[12], which enables the storage and processing of data from the remote cloud to be moved closer to the end-user and device, thus effectively solving the problems of bandwidth and data latency in cloud services [13]. Fig. 1 shows an IIoT scenario based on edge computing. In the current IIoT scenario, publish/subscribe (pub/sub) [14], [15], a common middleware technology that creates an intermediate node that can store, filter, and transmit data is added between some data senders called publishers. Some data receivers called subscribers perform efficient collection and transmission of perception data and also manage commands. Although cloud and edge computing have significant advantages for the IIoT, they transmit sensitive data from industrial manufacturing to the complex Internet [16], which can lead to security threats.

Manuscript received July 1, 2020; revised October 2, 2020; accepted November 6, 2020. Date of publication November 24, 2020; date of current version August 25, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61872001, Grant U1936220, and Grant 61702005 and in part by the Open Fund for Discipline Construction, Institute of Physical Science and Information Technology, Anhui University. *(Corresponding author: Hong Zhong.)*

The authors are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China, with the Institute of Physical Science and Information Technology, Anhui University, Hefei 230039, China, and also with the Anhui Engineering Laboratory of IoT Security Technologies, Anhui University, Hefei 230039, China (e-mail: cuijie@mail.ustc.edu.cn; fqwang108@foxmail.com; qingyang.zhang.inchina@gmail.com; xuyan@ahu.edu.cn; zhongh@ahu.edu.cn).

Color versions of one or more of the figures in this article are available at https://doi.org/10.1109/TIE.2020.3039227.

Digital Object Identifier 10.1109/TIE.2020.3039227

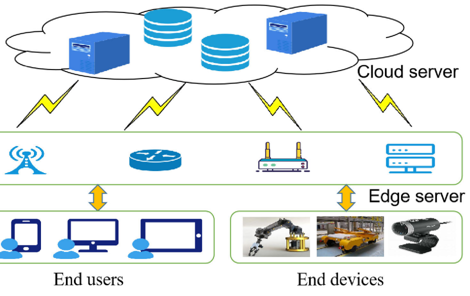


Fig. 1. IIoT scenario based on edge computing.

In the industrial manufacturing process, a large amount of extremely important privacy-sensitive data need to be transmitted to cloud or edge nodes [17], which may attract several attacks from the Internet. Therefore, to ensure the safe transmission and security of industrial manufacturing data, several researchers have outlined the following conditions. First, data integrity needs to be guaranteed [18]. If not, an attacker may manipulate the data without being detected, which could eventually lead to the attacker gaining control of the factory devices. Second, data confidentiality must be guaranteed [19]. If not, an attacker can intercept data from the industrial manufacturer, which can eventually lead to the attacker leaking the privacy-sensitive data of the factory. Particularly, in the pub/sub model, there is one-to-many communication, which is more challenging when data confidentiality and integrity are satisfied. Hence, several researchers [20] have proposed some security schemes for the IIoT.

In traditional schemes, transport layer security (TLS) [21] is often used as a classic protocol to ensure the integrity and confidentiality of IIoT data based on the pub/sub model, such as message queue telemetry transmission (MQTT) and extensible messaging and presence protocol (XMPP). Here, the intermediate node can obtain the original data using the secret key negotiated with the publisher. The intermediate node uses a secret key that is negotiated with the subscriber to encrypt the message, and the ciphertext is then sent to the corresponding subscriber. However, this only ensures data confidentiality and integrity between the cloud and the publisher or the cloud and the subscriber, without considering the identity of the publisher or subscriber. Meanwhile, data anonymity is very important. If there is no mechanism to guarantee data anonymity, the identity of the publisher can be easily known by an attacker through machine learning. As a result, the attacker can target factory devices or users based on the identity information obtained. Hence, some researchers have proposed several solutions for data anonymity [22], [23]. For example, Esposito *et al.* [15] proposed a message authentication scheme, that effectively guarantees data anonymity. However, abovementioned researchers regard the cloud or edge server as a completely trusted entity. Meanwhile, these entities are semitrusted because they are third parties; therefore, the proposed solutions are not suitable for scenarios in which the cloud or the edge server is semitrusted.

In this study, we comprehensively consider that the edge server is semitrusted. For the pub/sub-based IIoT, we propose a novel security scheme that uses group signature and proxy reencryption and to ensure the integrity, confidentiality, and anonymity of the IIoT data. The main contributions of our proposed scheme are summarized as follows.

1) Compared with the traditional IIoT scheme, which assumes that the edge server is fully trusted, we propose a more practical system model in which the edge server is semitrusted, which has a higher security request.
2) We propose a scheme for the pub/sub-based IIoT, using proxy reencryption and group signature technology, which guarantees data integrity, anonymity, and confidentiality.
3) Security analysis shows that our proposed scheme is applicable to the IIoT and can guarantee data security. In addition, we implement our scheme and compare its performance with other schemes, which further proves its feasibility.

II. Related Work

In this section, we first introduce the current research progress of Pub/Sub-based IIoT. Then, we introduce the research applied to the Pub/Sub system about security schemes, and point out some problems existing in the current security scheme.

A. Pub/Sub System in IIoT

In the IIoT, there are many end devices and end-users, and they have many data to transmit. The traditional scheme is to use the Internet to connect the sender and the receiver simply. However, due to the large amount of data interaction in the IIoT, its network topology is becoming increasingly more complex and less flexible. Therefore, many researchers are beginning to widely apply the Pub/Sub system with loose coupling, scalability in IIoT [24], [25].

IIoT is a specific application of internet of things (IoT) under the factory, and the Pub/Sub system is often used in it. To automate asset management in IIoT, Gandhewar *et al.* [26] designed an asset management architecture, and to meet the requirement of real-time data, and they used MQTT in this structure to make asset management more efficient. In order to effectively reduce the interdependence of entities in IIoT, Esposito *et al.* [15] mapped entities in IIoT to Pub/Sub system, achieving the flexibility of message transmission in IIoT. In other IoT application scenarios, Pub/Sub systems are also widely used. The system used in these scenarios can be used under IIoT with simple modifications or without modifications. In 2012, Kirsche and Klauck [27] focused on the dependence of sensors and actuators on middleware in the IoT, and they unified devices such as sensors and the Internet through XMPP, effectively omitting the middleware. In order to effectively support a large number of IoT devices and reduce the overhead between IoT devices and agents, in 2018, Sen and Balasubramanian [28] proposed a container architecture Nucleus for Pub/Sub systems with a limited number of agents in the IoT. Here, they use the lightweight communication protocol MQTT to realize the underlying network between IoT devices and agents.

B. Security in the Pub/Sub System

The abovementioned articles introduce suitable applications of the Pub/Sub model in the IIoT, but most researchers rarely consider the Pub/Sub model's security issues, which play a vital role in the IIoT. The leakage of privacy data can confuse the manufacturing of IIoT. Therefore, numerous scientific researchers have begun to consider the security of data transmission in IIoT.

To protect data integrity, Mektoubi *et al.* [29] proposed an efficient signature scheme that used Rivest–Shamir–Adleman encryption based on the elliptic curve encryption, and this signature scheme not only improves the computational efficiency but also ensures the integrity of the data. In order to ensure the confidentiality of the data in the Pub/Sub system, Bisne and Parmar [30] proposed a scheme that used attribute-based encryption and dynamic S-box advanced encryption standard (AES) in MQTT, and this scheme guarantees the confidentiality of data effectively. The publisher's identity in the Pub/Sub system needs to be anonymous; in 2018, Esposito *et al.* [15] mapped the data senders and recipients of the IIoT into a Pub/Sub model. The group signature technology is applied to the

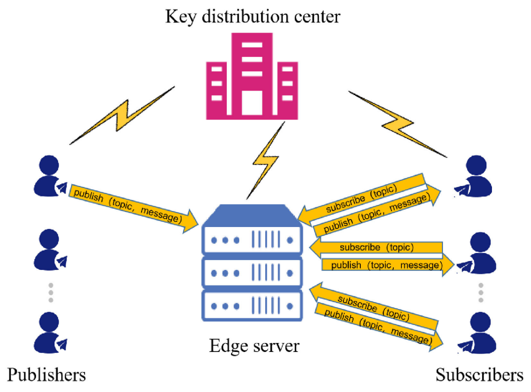


Fig. 2. Entities and network topology in our scheme.

Pub/Sub model, which effectively protects the data publisher's real identity. Although the abovementioned article have focused on the privacy security of data and have done some work to protect the data in the IIoT, they do not satisfy the confidentiality, anonymity, and integrity of the data at the same time. In our work, this is achieved by using group signature technology and proxy reencryption technology.

III. System Model

In this section, in order to present our proposed scheme more clearly, we present the details about our system model, including the primary function of each entity in the network model, with assumptions and security objectives.

A. Network Model and Assumptions

Fig. 2 illustrates our system model. And the connection among the key distribution center and the other entities in the network represents the secure keys distribution via a secure channel. It mainly contains four types of entities: key distribution center (KDC), edge server (ES), publishers, and subscribers. It is worth mentioning that in a real IIoT scenario, there are many publishers and subscribers. Publishers classify published messages into different topics by category; subscribers can subscribe to topics, and they can receive all messages on the topics to which they subscribe. For one publisher, the published topic can be subscribed by many subscribers, which is one-to-many communication. For example, in a smart factory, a factory manager's command can control the operation of multiple smart devices. The main functions of each entity are described belowmentioned.

1) KDC: The KDC is a cluster of servers in the IIoT environment. It is an independent, highly secure entity. Besides, KDC has powerful computing power and excellent storage capacity. The IIoT environment is responsible for generating system parameters and sending them to the corresponding entity.

2) ES: The ES is a cluster of servers that is closer to publishers and subscribers. It is a semitrusted entity. That is, it honestly follows the rules of encryption and decryption in the system, but it is curious about any data that passes through it. ES has good computing power and good storage capacity. The ES serves primarily as a connector between publishers and subscribers in the system. It can judge the legitimacy of a subscriber's identity when they subscribe to the topic; it can reencrypt the message published by the publisher, then store it, and finally filter the message to the legitimate subscriber.

3) Publisher: Publishers are end devices or end users in the IIoT environment, such as a factory manager. Each publisher is a fully trusted, independent entity. They have the poor computing power and limited storage capacity. In the Pub/Sub based IIoT scenario, the publisher acts as the sender of the message. They encrypt the messages and then transmit the encrypted message to an intermediate ES.

4) Subscriber: Subscribers are end devices or end users in the IIoT environment, such as a smart device. Each subscriber is a trusted entity. They have the poor computing power and limited storage capacity. In the Pub/Sub-based IIoT scenario, the subscriber acts as the receiver of the message. First, they subscribe to the topic of interest to the edge server and then wait for the message to arrive. Finally, they can decrypt the received message and verify its validity.

Here, it is worth noting that an end device or end user may be both a publisher and a subscriber in the system model. For example, if an end device such as a smart device publishes a message about the production schedule status, it acts as a publisher, and if the smart device subscribes to a topic regarding production order, it acts as a subscriber.

B. Security Objectives

There are two main types of attackers in our proposed scheme: a semitrusted edge server and a malicious Internet attacker. The semitrusted edge server is always trying to understand the plaintext meaning of the messages passing through it. The malicious Internet attacker can intercept and tamper with messages. In this section, we describe the security objectives to be achieved in our proposed scheme.

1) Integrity: Data integrity refers to the accuracy, legitimacy, and consistency of the messages in the system, which is the key to ensure that the data have not been tampered with. In our scheme, once an Internet attacker has tampered with the data in IIoT, entities will know that the data have been tampered with.

2) Confidentiality: Privacy-sensitive data in IIoT needs to be protected. ES is the intermediate forwarding node of the Pub/Sub system, where the data are still encrypted. This ensures that even if ES is malicious, there is no way to learn anything about the original data. Similarly, for Internet attackers, there is no way to know.

3) Anonymity: The real identity of the publisher needs to be protected. In the IIoT system, only KDC can trace the publisher's real identity. The adversarial edge nodes, malicious attackers, and other entities cannot infer the publisher's identity from the data. Furthermore, the publisher's identity also requires anonymity for subscribers because in some IIoT scenarios, for example, clients of the factory can check their order information, but they do not have the right to know, which specific device had published this information.

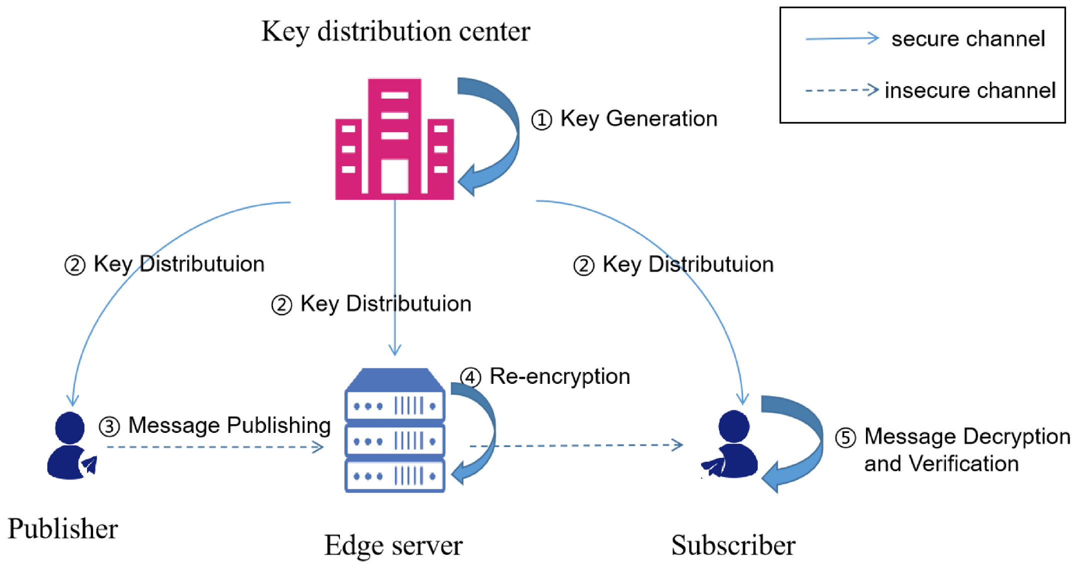


Fig. 3. Message authentication steps in our proposed scheme.

IV. Proposed Scheme

In this article, we design our own scheme inspired by group signature [31] and discrete logarithm [32]. And in this section, we describe our specific scheme in detail. It can be divided into five main stages: Key generation and distribution, message publishing, reencryption, message decryption and verification, and message tracing. The whole process of message authentication is shown in Fig. 3.

In the key generation and distribution phase, when a publisher defines a new topic, or a new entity joins the group, the KDC generates parameters and then distributes them to publishers, subscribers, and ES. In the message publishing phase, the publisher publishes a message, and the issued message is encrypted and sent with a signature to the ES. In the reencryption phase, upon receiving the publisher's data, the ES will reencrypt and store the message and then send it to the corresponding subscribers. In the message decryption and verification phase, subscribers receive their subscriptions; they can decrypt the message and recover the original data. In the message tracing phase, once a publisher is found to have published an incorrect message, the KDC can expose the real identity of the message publisher.

A. Key Generation and Distribution

In our scheme, when a publisher defines a new topic or a new entity joins the group, the KDC needs to generate many parameters that are used to encrypt, decrypt, sign, and validate messages in the Pub/Sub system.

1) Step 1: First, let G_1 and G_2 be two multiplicative cyclic groups of prime order q. In addition, let $e : G_1 \times G_2 \rightarrow G_T$ denote a computable map, let g_1 be a generator of G_1, g_2 be a generator of G_2. And the KDC chooses three secure one-way hash functions: $H_1 : \{0, 1\}^* \rightarrow G_1$, $H_2 : G_1 \rightarrow \{0, 1\}^*$, $H_3 : \{0, 1\}^* \rightarrow Z_q^*$. At the same time, the KDC generates random numbers $r \in Z_q^*$, $w = g_2^r$, $h \in G_1$. When a publisher publishes an invalid message, the KDC should be able to trace the publisher's real identity. Therefore, the KDC randomly generates keys $\varepsilon_1, \varepsilon_2 \in Z_q^*$ that can trace the specific identity of the publisher. In addition, there is a set of numbers u, v that satisfy the following properties: $u = h^{\varepsilon_1^{-1}}$, $v = h^{\varepsilon_2^{-1}}$, and the public key of the group signature is $gpk = (g_1, g_2, h, u, v, w)$. Here, the gpk is sent to publishers and subscribers on the same topic. The private key is $gmsk = (\varepsilon_1, \varepsilon_2)$, and stored in the KDC. The private key can be used by KDC to trace the identity of the publisher.

2) Step 2: In our proposed scheme, each publisher has a unique identified private key. When a new publisher joins the groups, the KDC sets i to the number of the ith publisher. As for ith publisher, the private key is used to generate a publisher's message encryption key that is set to $a_i \in Z_q^*$. The subscriber does not have the right to know the specific identity of the publisher, but it can know whether the publisher who published the message is legal. Therefore, it is necessary to hide the identity of the publisher, so a key $gsk[i] = (RID_i, x_i)$ is set to hide the publisher's identity, here $RID_i = g_1^{\frac{1}{(r+x_i)}}$, $x_i \in Z_q^*$. In our proposed scheme, the KDC generates a pseudonym group $PID_i = \{PID_{i,1}, PID_{i,2}, \ldots, PID_{i,p}\}$ for ith publisher, which used for hiding the real identity of the ith publisher. Subsequently, the KDC will send $gpk, gsk[i], a_i, PID_i, H_1, H_2, H_3$ to the ith publisher via a secure channel.

3) Step 3: When a new subscriber joins the group, the KDC needs to distribute a private key to the new subscriber. To reduce the computational overhead of ES in reencrypting, in our scheme, the subscriber's private key is the same, set to

$$SK_B = b \in Z_q^*. \tag{1}$$

And SK_B is sent to the corresponding subscriber via a secure channel.

4) Step 4: When the ES receives the message, the message can be reencrypted so that it can be decrypted by the subscriber, so when a publisher defines a new topic, the private key of reencryption of ES is set as $SK_{A\rightarrow B} = b/a_i$. Then, the $SK_{A\rightarrow B}$ is sent to the ES.

B. Message Publishing

When the publisher needs to publish a message, in order to protect the security of the data, it is necessary to sign and encrypt the data, and finally send the encrypted data to ES.

1) Step 1: To generate group signatures, the publisher needs to randomly generate $\alpha, \beta, r_\alpha, r_\beta, r_x, r_{\delta_1}, r_{\delta_2} \in Z_q^*$ and compute $T_1 = u^\alpha$, $T_2 = v^\beta$, $T_3 = RID_i h^{\alpha+\beta}$, $R_1 = u^{r_\alpha}$, $R_2 = v^{r_\beta}$, $R_3 = e(T_3, g_2)^{r_x} \cdot e(h, w)^{-r_\alpha \cdot - r_\beta} \cdot e(h, g_2)^{-r_{\delta_1} \cdot - r_{\delta_2}}$, $R_4 = {T_1}^{r_x} \cdot u^{-r_{\delta_1}}$, $R_5 = {T_2}^{r_x} \cdot u^{-r_{\delta_2}}$. Where $T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5$ are parameters for the group signature; T_1, T_2, T_3 can also be used to trace the real identity of the publisher by KDC. In addition, for later authentication, the publisher computes a challenge value

$$c = H_3(m, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5). \tag{2}$$

where the m is a message the publisher needs to post.

Then, the publisher computes $s_\alpha = r_\alpha + c\alpha$, $s_\beta = r_\beta + c\beta$, $s_x = r_x + cx$, $s_{\delta_1} = r_{\delta_1} + c\delta_1$, $s_{\delta_2} = r_{\delta_2} + c\delta_2$. Here, $s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2}$ are used to verify the validity of identity by subscriber, and the ith publisher computes its signature

$$\sigma = (\mathrm{T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2}}). \tag{3}$$

2) Step 2: To generate the encryption key, the publisher first generates a random number $k \in \{0,1\}^*$, computes $k' = H_1(k)$. Subsequently, the publisher generates the encryption key $SK_A = (k')^{a_i}$.

Next the publisher sets $M = (m||\sigma)$, computes $c_1 = PID_{i,j}||(M \oplus H_2(k'))$, $c_2 = SK_A$, and lets the ciphertext $c_a = (c_1, c_2)$. Finally, the publisher sends c_a to the ES. Where the c_a will be reencrypted by ES and converted into ciphertext that subscribers can decrypt.

C. Reencryption

Once the ES receives the message from the publisher, it will reencrypt and store the data, and then push the reencrypted data to the corresponding subscriber.

ES according to the pseudonym information in the message, selects the corresponding secret key to reencrypt the data. The ES first lets $c_1' = M \oplus H_2(k')$ and then computes

$$c_2' = (c_2)^{\frac{b}{a_i}}. \tag{4}$$

The reencrypted ciphertext is ${c_a}' = ({c_1}', c_2')$. Finally, the ${c_a}'$ is pushed to the corresponding subscriber. Later, the corresponding subscriber can use the c_2 to calculate the secret key that can decrypt the ciphertext c_a'.

It is worth noting that in order to minimize the length of the message, ES removes the pseudonyms $PID_{i,j}$ from the message during reencryption.

D. Message Decryption and Verification

Upon the subscriber receives the subscribed message, first decrypts the message and recovers the original data. And then the subscriber verifies the message.

TABLE I
SECURITY COMPARISON FOR ADVERSARIAL EDGE SERVER

	[15]	mTLS	Ours
Anonymity	×	×	√
Integrity	√	√	√
Confidentiality	×	×	√

√: The requirement is satisfied.
×: The requirement is not satisfied.

TABLE II
SECURITY COMPARISON FOR MALICIOUS INTERNET ATTACKERS

	[15]	mTLS	Ours
Anonymity	√	√	√
Integrity	√	√	√
Confidentiality	×	√	√

√: The requirement is satisfied.
×: The requirement is not satisfied.

1) Step 1: Upon the subscriber receives c_a', it first decrypts c_a' with its own private key: $(c_2')^{\frac{1}{b}} = k'$, $M = H_2(k') \oplus c_1'$. After getting the plaintext, the subscriber can get m and σ.

2) Step 2: In order to verify the signature, the subscriber needs to perform: $\tilde{R_1} = u^{s_\alpha} \cdot {T_1}^{-c}$, $\tilde{R_2} = {T_2}^{-c}$, $\tilde{R_3} = e(T_3, g_2,)^{s_x} \cdot e(h, w)^{-s_\alpha - s_\beta} \cdot e(h, g_2)^{-s_{\delta_1} - s_{\delta_2}} \cdot (\frac{e(T_3,w)}{e(g_1,g_2)})^c$, $\tilde{R_4} = T_1^{s_x} \cdot u^{-s_{\delta 1}}$, $\tilde{R_5} = {T_2}^{s_x} \cdot v^{-s_{\delta_2}}$, then calculates

$$c' = H_3(m, T_1, T_2, T_3, \tilde{R_1}, \tilde{R_2}, \tilde{R_3}, \tilde{R_4}, \tilde{R_5}). \tag{5}$$

Finally, the subscriber determines if (5) is equal to (2). If they are equal, it proves that the received message is valid, and the subscriber processes the message. If they are not equal, the subscriber discards the data.

E. Message Tracing

When, for example, a production command in a smart factory is found to have been issued incorrectly, it needs to find out which specific publisher issued the command. In our proposed scheme, the KDC can trace the source.

When tracing the publisher's real identity, the KDC can use $SK_{A\to B}$ and SK_B to reencrypt and decrypt the message. Then, the KDC can get T_1, T_2, T_3. Because the KDC has a set of keys $gmsk = (\varepsilon_1, \varepsilon_2)$, it can calculate $u^{\alpha^{\varepsilon_1}} \cdot v^{\beta^{\varepsilon_2}}$, then it can get $h^{\alpha+\beta}$. Finally, the KDC can calculate $T_3/(h^{\alpha+\beta})$ to get RID_i, which means the KDC get the real identity of the publisher of the message.

V. SECURITY ANALYSIS

In this section, we compare the security of [15], anonymous TLS, and proposed scheme against semitrusted ES and malicious Internet attackers, respectively. The results are illustrated in Tables I and II, which show that our scheme is securer than others.

MQTT and XMPP mainly use traditional TLS to ensure data security. And the original TLS only guarantees the data's confidentiality and integrity, not the anonymity of the data. Therefore, in our experiments, we have modified the original TLS scheme, mainly by adding the group signature to it and utilizing AES to encrypt it, forming an anonymous TLS scheme (labeled as mTLS) for discussion. In [15], they adopted an identity-based encryption cryptography technique to ensure the security of message transmission when adding and deleting group members. Only the group signature was employed during the message publishing subscription, and the message was not encrypted.

A. Anonymity

Before sending its message c_a to the ES, the publisher hides its real identity RID_i in T_3, where $T_3 = RID_i h^{\alpha+\beta}$. When the intermediate ES and malicious Internet attackers get the message sent by the publisher, only T_3 can be obtained. From the construction of T_3, these attackers cannot calculate RID_i unless they can get the $gmsk$ or compute $h^{\alpha+\beta}$. It is worth noting that $gmsk$ is forever stored in a fully trusted KDC, which is not available to an attacker.

When the subscriber decrypts the requested message, only the content of the message can be obtained, the RID_i of the publisher cannot be known. In our scheme, only KDC with $gmsk$ can calculate the publisher's RID_i from the publisher's message.

After analysis, we find that for ES, neither the mTLS scheme nor the scheme of [15] cannot guarantee the anonymity of the messages. Because in the mTLS scheme, the ES encrypts and decrypts messages using AES, selecting the corresponding secret key based on the real identity of the publisher or subscriber. And in [15], the ES does not hide the real identity of the publisher, so it can obtain the real identity of the publisher. For other attackers on the Internet, both [15] and mTLS use group signatures, so the anonymity of the data can be guaranteed.

B. Integrity

Malicious Internet attackers temper a message and sent it to the corresponding ES or subscribers. Subscribers use $R_1^{\sim} = u^{s_\alpha} \cdot T_1^{-c}$, $R_2^{\sim} = T_2^{-c}, R_3^{\sim} = e(T_3, g_2,)^{s_x} \cdot e(h, w)^{-s_\alpha - s_\beta} \cdot e(h.g_2)^{-s_{\delta 1} - s_{\delta 2}} \cdot (\frac{e(T_3,w)}{e(g_1,g_2)})^c$, $R_4^{\sim} = T_1^{s_x} \cdot u^{-s_{\delta 1}}$, $R_5^{\sim} = T_2^{s_x} \cdot v^{-s_{\delta 2}}$ to determine if (5) is equal to (2) to judge the validity of the received messages. If the identity of the publisher is found to be invalid after verification, the message will be discarded. Therefore, our scheme can ensure the integrity of the message.

ES is curious but honest, and it does not maliciously tamper with the data. Therefore, for ES, [15] and the mTLS scheme can guarantee data integrity. However, Internet attackers will tamper with the data, but both [15] and mTLS use group signatures, so the integrity of the data can be guaranteed.

C. Confidentiality

In our scheme, we utilize proxy reencryption. After the ES receives the data encrypted by the publisher, it cannot know the real identity RID_i, and the message exists as ciphertext. If the ES wants to decrypt the message to get plaintext, it must have $SK_{A\to B}$ and SK_B to reencrypt the message via $c_2' = (c_2)^{\frac{b}{a_i}}$ and then decrypte the message via $(c_2')^{\frac{1}{b}} = k'$, $M = H_2(k') \oplus c_1$, but the ES only has $SK_{A\to B}$. Hence, the ES cannot get useful knowledge about the message. For Internet attackers, they can only get the ciphertext because the data exists as ciphertext. Thus, our proposed scheme can guarantee the confidentiality of messages.

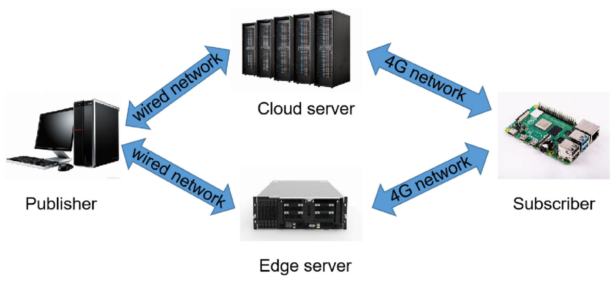


Fig. 4. Experimental network topology.

In ES, for mTLS, ES can decrypt the message with a negotiated secret key, in [15], the encryption of messages is not considered. Therefore, neither scheme guarantees the confidentiality of the message in ES. For malicious attackers in the Internet, in [15], since the data are not encrypted, the confidentiality of the data cannot be guaranteed. On the contrary, in mTLS, it encrypts data, so it guarantees the anonymity of the data over the Internet.

VI. Performance Analysis

In the IIoT environment, end users' computing power, terminal devices, and edge nodes are limited. Meanwhile, real-time data are also of great significance for industrial manufacturing. Therefore, to prove that our proposed scheme's communication and computing costs are compatible with IIoT scenarios, we conducted relevant experiments on the Pub/Sub system.

A. Experiment Setup

We implemented our proposed scheme with approximately 1000 lines of C code on the NATS server version 2.1.0 system. We used the GMP library version 6.1.2 to implement large integer operations and the PBC library version 0.5.14 to implement pairing calculations. We chose the Type-A curve, which provides an 80-b security level.

As shown in Fig. 4, we used a PC as a publisher. This PC running Ubuntu 18.04.3 has Intel Core i5-7500 CPU 3.40 GHz and 16 GB of memory. We used a Raspberry Pi 4 as a subscriber, with 1.5 GB Hz CPU and 4 GB memory. Here, PC carries out data transmission through a wired network, and Raspberry Pi carries out data transmission through the 4 G network. In addition, the server of our Pub/Sub system is deployed on a cloud server. In order to realize the comparison scheme of ES and the cloud server, we respectively deployed experiments on two cloud servers. A cloud server, closer to the client (about 400 km), is regarded as the ES, and another cloud server that is far away from the client (about 1500 km) is regarded as the cloud server. The two cloud servers' configuration is the same, running Ubuntu 18.04.3, ecs.g5.large instance with two vCPUs @2.5 GHz, and the network bandwidth is 100 Mbps.

TABLE III
COST COMPARISON (MS)

	Group Signature	Encryption	Transmission	Re-Encryption	Decryption	Verification	Sum
[15] (Edge)	13.092	-	50.581	-	-	72.995	136.668
mTLS (Edge)	13.126	0.004	52.558	0.021	0.015	73.076	138.800
Ours (Edge)	13.102	1.124	52.550	1.925	5.033	73.387	147.121
et al.[15] (Cloud)	13.101	-	73.472	-	-	73.028	159.601
mTLS (Cloud)	13.122	0.004	72.414	0.020	0.018	73.166	158.744
Ours (Cloud)	13.092	1.122	71.258	1.954	4.986	73.076	165.488

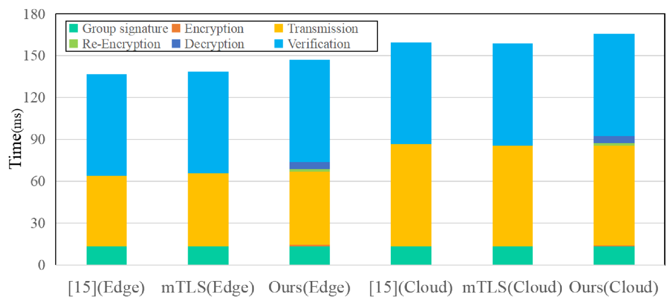


Fig. 5. Time consumption of different schemes.

TABLE IV
COST OF THREE SCHEMES

	publisher	edge server	subscriber
Esposito *et al.*[15]	$12T_{exp}$+$3T_{pair}$	0	$12T_{exp}$+$5T_{pair}$
Our scheme	$13T_{exp}$+$3T_{pair}$	$1T_{exp}$	$13T_{exp}$+$5T_{pair}$
mTLS	$12T_{exp}$+$3T_{pair}$	0	$12T_{exp}$+$5T_{pair}$

T_{exp}: exponential operation.
T_{pair}: pairing operation.

To conduct a comprehensive evaluation of the protocol, we compared our scheme with another two schemes, the one without any encryption operation and the mTLS scheme. Each scheme was run 1000 times, and the average was reported for comparison.

B. Experimental Comparison

In the experiment, group signature generation time, encryption time, decryption time and transmission time are shown in Fig. 5; the specific cost time is shown in Table III, and the computational complexity is shown in Table IV. The following is a detailed analysis of the time consumed by these processes.

1) Group Signature Generation Time: In our proposed scheme, the same group signature scheme is used as the other two encryption schemes, so in our experiments, we can see the group signature generation time is 13.106 ms.

2) Encryption Time: As we can see, the time of mTLS encryption is not obvious because mTLS encryption only takes about 4 μs, but in our scheme, the time required for encryption is 1.123 ms because in our proposed scheme, the first encryption operation in reencryption requires an exponential operation. Although our encryption operation time is relatively long, it does not affect the user's experience, and our proposed scheme makes the data more secure and confidential than the unencrypted scheme. Compared to mTLS encryption, because each publisher only encrypts their own signatures and data, so it does not affect encryption performance.

3) Decryption Time: The decryption time means that the subscriber receives the ES's message to perform the decryption operation. The mTLS decryption operation only took about 16.5 μs in our experiments, so the mTLS decryption time is not very obvious. In our proposed scheme, the decryption operation takes about 5.010 ms. As with the publisher, since each subscribed message decryption operation is decrypted by its corresponding subscriber, an operation of about 1 ms does not affect the subscriber experience during the entire encryption and decryption transmission process.

4) Transmission Time: As shown from the figure, if the intermediate server is configured at the edge, the transmission time is approximately 51.896 ms. If the server is configured on the cloud, the transfer time is about 72.381 ms. It can also be easily seen that the server's transmission time configured on the cloud is significantly greater than the transmission time configured to the edge end. Thus, in our experiments, we propose arranging the ES to the edge server.

5) Reencryption Time: Our experiments found that the mTLS decrypted for about 14 μs, and the encryption operation took about 7 μs. If the reencryption scheme is used, then the decryption operation is not required, and only the ES needs to perform the reencryption operation on the received message. In our experiments, the time required is about 1.940 ms.

In our experiments, although we can see that the encryption and decryption time of mTLS is relatively short compared to the reencryption operation, after mTLS decrypts the ES, the message to be released by the publisher is completely exposed to the ES, which is dangerous because the ES is is a semitrusted. If mTLS is used for encryption and decryption, then the plaintext is easily intercepted by the intermediate ES, and the data cannot be guaranteed safety. However, the reencryption scheme we applied guarantees the confidentiality of the data.

On the other hand, in real life, one topic can be subscribed by many subscribers. If there are a total of n subscribers, then in the mTLS scheme, n decryption operations and n encryption operations are required, and in the reencryption scheme, only one reencryption operation is required. As shown in Fig. 6, when the number of subscribers subscribing to the same message is less than 275, our scheme consumes more time than the mTLS scheme. However, the time taken relative to the whole process is still less, so it is fully applicable to IIoT environments. Even though in this case the encryption and decryption speed of our proposed scheme is not as fast as that of mTLS, our solution is based on the environment still applicable to IIoT. Regarding the

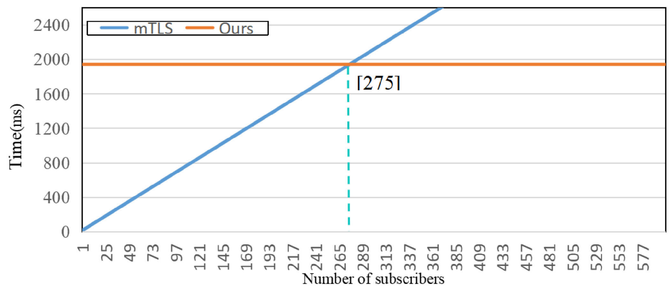


Fig. 6. Performance comparisons between mTLS and our scheme on edge server side.

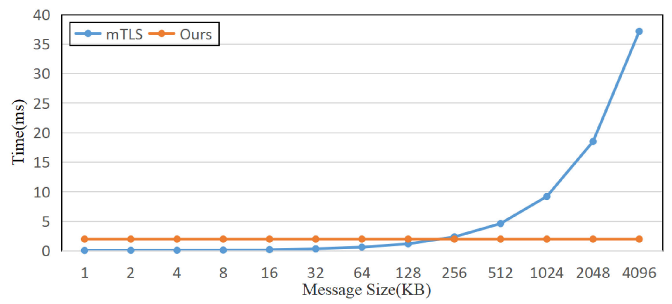


Fig. 7. Time consumption for reencryption with different message lengths in mTLS and our scheme.

problem that ES is a semitrusted entity, it effectively ensures the confidentiality and anonymity of the data when the number of subscribers subscribing to the same message is greater than 275. For example, when subscribers are devices of the smart factory and the manager of the factory as the publisher needs to issue control commands to make the smart device work, the number of smart devices is likely to exceed 275. In this case, our proposed scheme not only shows high security, but it also has some advantages in time consumption. In addition, Fig. 7 shows that as the length of published messages increases, the time required for mTLS to reencrypt is increasing during the reencryption phase. However, in our proposed scheme, the time required to reencrypt does not change. And when the message length is long, such as 256 KB, compared to mTLS, our proposed scheme can save more time. Because the mTLS needs to reencrypt the whole message and in our proposed scheme, we only reencrypt the SK_A, the length of SK_A is constant. Therefore, it can be proved that our proposed scheme has more advantages over mTLS when the message length is long. For example, when a factory manager requests a video or picture that a smart device is monitoring remotely and when factory managers need to install software into smart devices using over the air technology, the message length is easily reach 256 KB.

In our scheme, if some work can be done in advance on end devices, the computation overhead of the end devices will be reduced. For example, before the publisher generates the group signature, it can compute $e(h, w)$ and $e(h, g_2)$, resulting in saving two pairing operations and reducing up to 8% of the time on generating the group signature (i.e., approximately 13 ms in Table III). Furthermore, before the subscriber decrypts a message, it can compute $e(h, w)$, $e(g_1, g_2)$, and $e(h, g_2)$, resulting in saving three pairing operations and reducing up to 25% of the time on signature verifying (i.e., approximately 73 ms in Table III).

VII. Conclusion

The development of the IIoT had significantly altered production processes in traditional factories, thereby ensured the efficient production and personalization of goods. Meanwhile, security was an important issue that couldnot be ignored in the IIoT environment. In this study, we mapped the entities of the IIoT to the pub/sub model and proposed a message authentication scheme that could be practically applied to the IIoT. The scheme used group signature and proxy reencryption technology to achieved message anonymity, confidentiality, and integrity. Security analysis and experiments showed that the scheme was effective and performs well. However, the scheme was mainly applicable to the authentication of a single message. In the future, we would use batch certification to design a secure and efficient scheme.

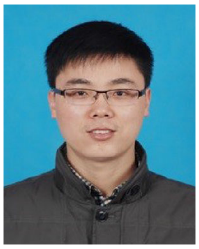
Acknowledgment

The authors would like to thank anonymous referees for their detailed comments and suggestions regarding this article.

References

[1] L. Da Xu, W. He, and S. Li, “Internet of things in industries: A survey,” *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.
[2] M. Nkomo *et al.*, “Overlay virtualized wireless sensor networks for application in industrial internet of things: A review,” *Sensors*, vol. 18, no. 10, 2018, Art. no. 3215.
[3] C. Arnold *et al.*, “The industrial internet of things from a management perspective: A systematic review of current literature,” *J. Emerg. Trends Marketing Manage.*, vol. 1, no. 1, pp. 8–21, 2017.
[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
[5] G. Shah and A. Tiwari, “Anomaly detection in IIoT: A case study using machine learning,” in *Proc. ACM India Joint Int. Conf. Data Sci. Manage. Data*, 2018, pp. 295–300.
[6] J.-S. Fu, Y. Liu, H.-C. Chao, B. K. Bhargava, and Z.-J. Zhang, “Secure data storage and searching for industrial IoT by integrating fog computing and cloud computing,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4519–4528, Oct. 2018.
[7] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, “Secure integration of IoT and cloud computing,” *Future Gener. Comput. Syst.*, vol. 78, pp. 964–975, 2018.
[8] T. Rieger, S. Regier, I. Stengel, and N. L. Clarke, “Fast predictive maintenance in industrial internet of things (IIoT) with deep learning (dl): A review,” in *Proc. CERC*, 2019, pp. 69–80.
[9] Y. Chen, Q. Feng, and W. Shi, “An industrial robot system based on edge computing: An early experience,” in *Proc. Workshop Hot Top. Edge Comput. (HotEdge 18)*, 2018, pp. 1–6.
[10] Q. Zhang, Q. Zhang, W. Shi, and H. Zhong, “Distributed collaborative execution on the edges and its application to amber alerts,” *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3580–3593, Oct. 2018.
[11] N.-N. Dao, Y. Lee, S. Cho, E. Kim, K.-S. Chung, and C. Keum, “Multi-tier multi-access edge computing: The role for the fourth industrial revolution,” in *Proc. IEEE Int. Conf. Inf. Commun. Technol. Convergence.*, 2017, pp. 1280–1282.
[12] Q. Zhang, Q. Zhang, W. Shi, and H. Zhong, “Firework: Data processing and sharing for hybrid cloud-edge analytics,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 9, pp. 2004–2017, Sep. 2018.
[13] Q. Zhang, H. Zhong, J. Wu, and W. Shi, “How edge computing and initial congestion window affect latency of web-based services: Early experiences with baidu?,” in *Proc. IEEE/ACM Symp. Edge Comput.*, 2018, pp. 393–398.
[14] R. De Ligt, “Industrial process control using IP communications with publisher subscriber pattern,” U.S. Patent 15 218 844, Jan. 2018.

[15] C. Esposito, A. Castiglione, F. Palmieri, and A. De Santis, "Integrity for an event notification within the industrial internet of things by using group signatures," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3669–3678, Aug. 2018.
[16] N. Subramanian and A. Jeyaraj, "Recent security challenges in cloud computing," *Comput. Elect. Eng.*, vol. 71, pp. 28–42, 2018.
[17] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," in *Proc. IEEE 52nd ACM/EDAC/IEEE Des. Autom. Conf.*, 2015, pp. 1–6.
[18] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu, "Blockchain based data integrity service framework for IoT data," in *Proc. IEEE Int. Conf. Web. Serv.*, 2017, pp. 468–475.
[19] R. H. Khalaf and A. H. Mohammed, "Confidentiality and integrity of sensing data transmission in IoT application," *Int. J. Eng. Technol.*, vol. 7, no. 4.25, pp. 240–245, 2018.
[20] Y. Shen, "The blockchain based system to guarantee the data integrity of IIoT," p. 62, 2018.
[21] T. Dierks and E. Rescorla, "The transport layer security (TLS) protocol version 1.2," Internet Requests for Comments, RFC 5246, Aug. 2008. [Online]. Available: https://www.rfc-editor.org/info/rfc5246
[22] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, "PA-CRT: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *IEEE Trans. Dependable Secure Comput.*, to be published, doi: 10.1109/TDSC.2019.2904274.
[23] J. Wang, L. Wu, K.-K. R. Choo, and D. He, "Blockchain based anonymous authentication with key management for smart grid edge computing infrastructure," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1984–1992, Mar. 2020.
[24] A. V. Uzunov, "A survey of security solutions for distributed publish/subscribe systems," *Comput. Secur.*, vol. 61, pp. 94–129, 2016.
[25] A. Stanford-Clark and H. L. Truong, "MQTT for sensor networks (MQTT-SN) protocol specification," International Business Machines Corporation, Armonk, NY, USA, 2013.
[26] R. Gandhewar, A. Gaurav, K. Kokate, H. Khetan, and H. Kamat, "Cloud based framework for IIoT application with asset management," in *Proc. IEEE 3rd Int. Conf. Electron., Commun. Aerosp. Technol.*, 2019, pp. 920–925.
[27] M. Kirsche and R. Klauck, "Unify to bridge gaps: Bringing XMPP into the internet of things," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops.*, 2012, pp. 455–458.
[28] S. Sen and A. Balasubramanian, "A highly resilient and scalable broker architecture for iot applications," in *Proc. IEEE 10th Int. Conf. Commun. Syst. Netw.*, 2018, pp. 336–341.
[29] A. Mektoubi, H. L. Hassani, H. Belhadaoui, M. Rifi, and A. Zakari, "New approach for securing communication over MQTT protocol a comparison between RSA and elliptic curve," in *Proc. IEEE 3rd Int. Conf. Syst. Collaboration.*, 2016, pp. 1–6.
[30] L. Bisne and M. Parmar, "Composite secure mqtt for internet of things using abe and dynamic S-box AES," in *Proc. IEEE Innovations Power Adv. Comput. Technol.*, 2017, pp. 1–5.
[31] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proc. Annu. Int. Cryptol. Conf.*, 2004, pp. 41–55.
[32] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, 1998, pp. 127–144.



Jie Cui was born in Henan Province, China, in 1980. He received the Ph.D. degree in computer science and technology from the University of Science and Technology of China, Hefei, in 2012.

He is currently a Professor and the Ph.D. Supervisor of the School of Computer Science and Technology with Anhui University. His current research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security, and software-defined networking (SDN).

He has more 100 scientific publications in reputable journals (e.g., IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS and IEEE INTERNET OF THINGS JOURNAl), academic books and international conferences.

Fengqun Wang is currently working toward the graduate degree in computer technology with the School of Computer Science and Technology, Anhui University, Hefei, China. His research focuses on the security of IIoT.

Qingyang Zhang received the B.Eng. degree in 2014 in computer science and technology from Anhui University, Hefei, China, where he is currently working toward the Ph.D. degree in computer science and technology.

His research interest includes edge computing, computer systems, and security.

Yan Xu received the B.S. degree in information security and M.S. degree in systems analysis and integration from Shandong University, Jinan, China, in 2004 and 2007, respectively, and the Ph.D. degree in computer science and technology from the University of Science and Technology of China, Hefei, in 2015. She is currently an Associate Professor of School of Computer Science and Technology, Anhui University. Her research interests include information security and applied cryptography.

Hong Zhong was born in Anhui Province, China, in 1965. She received the Ph.D. degree in computer science from the University of Science and Technology of China, Hefei, in 2005.

She is currently a Professor and the Ph.D. Supervisor of the School of Computer Science and Technology, Anhui University. Her research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security and SDN.

She has more than 120 scientific publications in reputable journals (e.g., IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON BIG DATA, and IEEE IOT JOURNAL), academic books and international conferences.