Blockchain-Based Secure Cross-Domain Data Sharing for Edge-Assisted Industrial Internet of Things

Fengqun Wang, Jie Cui, *Senior Member, IEEE*, Qingyang Zhang, *Member, IEEE*, Debiao He, *Member, IEEE*, and Hong Zhong, *Member, IEEE*

***Abstract*— In the Industrial Internet of Things (IIoT), blockchain-based data-sharing frameworks can effectively build cross-domain trust and facilitate data sharing. However, secure data-sharing schemes are lacking for the IIoT scenario, in which smart devices cannot communicate across domains and can only access data through edge servers. In this study, we propose a lightweight and secure data-sharing scheme for the blockchain-enabled cross-domain IIoT, in which authorized smart devices can access cross-domain data anonymously. First, smart devices can dynamically generate pseudonyms by themselves and without the online participation of domain authorization centers, effectively reducing the storage overhead of smart devices and the workload of domain authorization centers. Second, the scheme combines broadcast encryption and proxy re-encryption techniques, which realize flexible data sharing across domains while protecting the privacy of smart devices. Detailed security proofs and analyses demonstrate that the proposed scheme is secure and resistant to various attacks. The performance analysis shows that our proposed scheme is efficient and performs better than related schemes.**

***Index Terms*— Industrial Internet of Things (IIoT), authentication, anonymous, blockchain, cross-domain, broadcast encryption, proxy re-encryption, data sharing.**

I. Introduction

WITH the application of the Internet of Things in industrial production, the Industrial Internet of Things (IIoT) has received widespread attention [1], [2], [3]. As industrial manufacturing progresses towards refinement, data sharing between multiple administrative domains (e.g., smart factories) is becoming an inevitable trend [4], [5], [6]. For example, in a smart factory, a smart device that manufactures tires must obtain the model, size, power, and other car parameters from other smart factories to accurately complete the customized production of tires. However, in cross-domain IIoT, entities in each administrative domain trust only their own domain authorization center. To overcome cross-domain trust barriers and build trust between multiple domains, many studies have introduced blockchains into cross-domain IIoT systems [7], [8].

Fig. 1 shows a typical blockchain-based data-sharing framework for the cross-domain IIoT. In this framework, edge servers work together to maintain the consistency of the consortium blockchain. Smart devices subscribe to the services and access cross-domain data based on production tasks [8]. For example, to accomplish customized production, the smart device producing car tires in smart factory B must access the car orders in smart factory A. However, to protect the security and privacy of the factory, the smart device cannot directly access the data in factory A. Therefore, the edge server in factory B typically acts as a proxy, forwarding car orders to the smart device. Specifically, the data owner (e.g., the edge server in domain A) encrypts the original data, stores the encrypted data in the cloud server, and finally uploads the metadata (including the ciphertext's storage address) to the blockchain [9], [10]. When a smart device in another domain (e.g., domain B) seeks to access the original data, the edge server in domain B checks whether the access request of the smart device is legitimate. If it is legitimate, the edge server obtains the corresponding metadata from the blockchain and forwards it to the smart device. Finally, the smart device retrieves the corresponding ciphertext from the cloud server through the metadata and decrypts it to obtain the original data.

At present, several data sharing schemes have been proposed to ensure that data is accessed securely. However, when applied to the scenario mentioned above, they still face the following challenges.

On the one hand, to prevent smart devices from targeted attacks and protect IIoT privacy, smart devices must remain anonymous. However, in the existing anonymous authenti-

Manuscript received 15 August 2023; revised 26 December 2023 and 16 February 2024; accepted 19 February 2024. Date of publication 4 March 2024; date of current version 2 May 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62272002, Grant U23A20308, and Grant 62202005; in part by the Excellent Youth Foundation of Anhui Scientific Committee under Grant 2108085J31; in part by the Natural Science Foundation of Anhui Province, China, under Grant 2208085QF198; in part by the Open Fund of Anhui Province Key Laboratory of Cyberspace Security Situation Awareness and Evaluation under Grant CSSAE-2021-008; and in part by the University Synergy Innovation Program of Anhui Province under Grant GXXT-2022-049. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ghassan Karame. *(Corresponding author: Jie Cui.)*

Fengqun Wang, Jie Cui, Qingyang Zhang, and Hong Zhong are with the Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, School of Computer Science and Technology, Anhui Engineering Laboratory of IoT Security Technologies, and the Institute of Physical Science and Information Technology, Anhui University, Hefei 230039, China (e-mail: cuijie@mail.ustc.edu.cn).

Debiao He is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: hedebiao@163.com).

Digital Object Identifier 10.1109/TIFS.2024.3372806

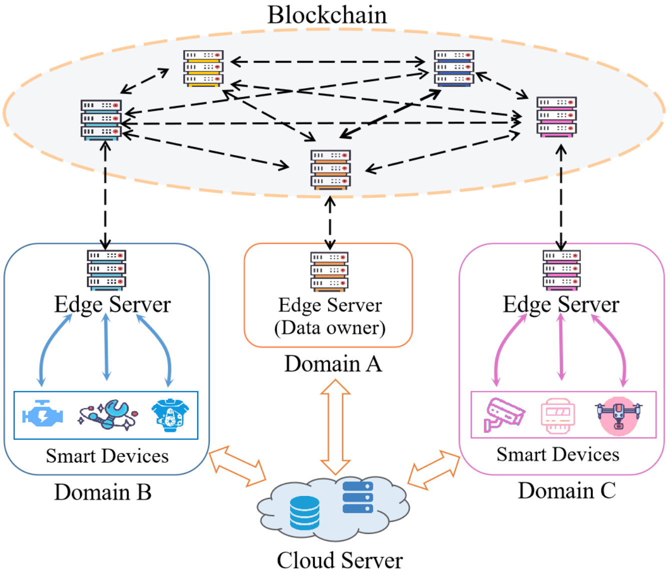


Fig. 1. Blockchain-based data-sharing framework in cross-domain IIoT.

cation algorithms, the computational cost or storage cost of smart devices is significant. For example, in scheme [11], smart devices utilize group signatures to achieve anonymity, but the signature algorithm involves many time-consuming pairing operations, which imposes a huge computational pressure on smart devices with limited computational power. In scheme [12], smart devices need to request many pseudonyms from the domain authorization center and store them in advance, which consumes considerable storage resources for smart devices with limited storage capacity. Therefore, it is necessary to design a lightweight and anonymous authentication algorithm.

On the other hand, existing data sharing schemes cannot meet the security and flexibility needs at the same time. For example, in scheme [13], the authors do not consider the anonymous authentication of smart devices, which may leak sensitive information about the IIoT. In scheme [14], the data owner directly encrypts the original data into a ciphertext that the data user can decrypt. However, in the cross-domain IIoT, it is difficult for data owners to communicate directly with smart devices and get their identities. In addition, smart devices accessing the same set of original data may come from different administrative domains, which increases the difficulty of achieving flexible access control for these smart devices. Therefore, for the scenario in which smart devices cannot cross-domain communication directly, it is urgent to design a data-sharing protocol that achieves flexible data sharing while ensuring the anonymity of smart devices.

A. Our Motivations and Research Focus

In this paper, we focus on IIoT scenarios where smart devices can only access data from other administrative domains with the assistance of edge servers. This scenario lacks the corresponding data sharing scheme. In addition, we find that the existing schemes cannot be directly applied to the scenarios we are interested in. The reasons are mainly focused on the following two points: 1) existing anonymous authentication algorithms have high computational overhead or storage overhead, which is intolerable for resource-constrained smart devices; 2) existing data sharing protocols do not simultaneously consider the security and flexibility of data sharing across domains. Therefore, we are motivated to propose a secure, flexible, and efficient cross-domain data sharing scheme for solving the above problems.

B. Our Contribution

In this paper, we propose a blockchain-based cross-domain data-sharing scheme for edge-assisted IIoT. The main contributions of this study are as follows:

- We design an anonymous authentication algorithm that achieves an efficient trade-off between the computational overhead and storage overhead of smart devices. In particular, the smart device can update pseudonyms by itself without sending pseudonym update requests to the domain authorization center and pre-storing massive pseudonyms.
- We design a blockchain-based cross-domain data-sharing protocol by combining broadcast encryption and proxy re-encryption techniques. The protocol realizes secure and flexible data sharing across domains without leaking the privacy of smart devices, making it suitable for the scenario that smart devices cannot access cross-domain data directly.
- The security proof and security analysis show that our proposed cross-domain data-sharing scheme is secure and meets the security objectives of the cross-domain IIoT. Moreover, computational and communication overhead analysis shows that the proposed scheme outperforms other related schemes.

C. Organization of the Rest Paper

Section II describes related work. Section III presents the relevant preliminaries. Section IV describes the system model and security requirements. Section V describes our proposed scheme. Section VI demonstrates the security of our proposed scheme through security proof and analysis. Section VII presents the performance. Section VIII summarizes the proposed scheme.

II. Related Work

In this section, we focus on existing data-sharing schemes and point out the limitations of their application to cross-domain IIoT environments.

A. Broadcast Encryption and Proxy Re-Encryption

Broadcast encryption [15] supports a broadcaster to send ciphertext to multiple receivers at the same time and is often used in one-to-many data-sharing schemes. In [16], Kim et al. proposed an adaptive broadcast encryption protocol that enables secure data sharing. However, the computational complexity of decryption increases with the number of data users. In [14], Xu et al. proposed an anonymous broadcast encryption protocol, the decryption complexity is constant. This protocol generates multiple label-ciphertext pairs, and only the authorized receiver can find the correct ciphertext and decrypt it via the label, thus achieving the anonymity of the receiver. Like the scheme [17], [18], [19], [20], in scheme [14], the broadcaster must have the receiver's identity set. However, the real identities of smart devices need to be anonymized, and these smart devices cannot communicate with the data owner across domains.

Proxy re-encryption is a cryptographic technique [21] that enables flexible data sharing while guaranteeing data confidentiality. It allows the data owner to delegate the encrypted data to a proxy server, which then re-encrypts the data and authorizes it to other data users. In [22], Chen et al. designed a data-sharing scheme based on a proxy re-encryption technique. However, the re-encryption algorithm in this scheme is run by data owners, which requires them to know the information of the data user and is not conducive to protecting the privacy of smart devices in a cross-domain IIoT. In [23], Agyekum et al. proposed a blockchain-based data-sharing scheme. The scheme uses identity-based encryption and proxy re-encryption techniques to achieve fine-grained access control to data. In [24], Zhang and Chen proposed a data-sharing scheme for 5G IIoT environments. The system utilizes proxy re-encryption and fog computing to achieve a secure and ready distribution of tasks. In [25], Manzoor et al. used blockchain and cloud computing technologies to design a data-sharing framework for smart device data transactions. In addition, the framework uses proxy re-encryption techniques to ensure data confidentiality and integrity. In [26], Lin et al. proposed a cloud-assisted data-sharing scheme. The scheme uses outsourcing decryption and proxy re-encryption technology to reduce the computational cost of decryption effectively. However, if these proxy re-encryption schemes are applied directly to cross-domain IIoT, the computational overhead of the data owner grows linearly with the number of administrative domains, generating significant redundancy overhead.

In [13], Sun et al. proposed a secure and efficient broadcast authorization scheme that uses broadcast proxy re-encryption. The scheme simultaneously considers the efficiency of the sharing policy, the security of the recipient's privacy, and the verifiability of the ciphertext to achieve flexible data sharing in the cloud. However, decryption requires many time-consuming cryptographic operations, which consume a large amount of computational resources for the receiver. Moreover, similar to the other broadcast encryption and proxy re-encryption schemes described above, this scheme does not determine the legitimacy of a smart device's access request.

Through the above analysis, we find that the existing data sharing schemes cannot simultaneously take into account the computation overhead of IIoT entities, the anonymity of smart devices, and the flexibility of cross-domain data sharing. Therefore, existing data sharing protocols cannot be directly applied to the cross-domain IIoT scenario where smart devices cannot interact across domains.

B. Anonymous Authentication

Xue et al. [11] proposed a secure, efficient, and reliable access control framework. The framework utilizes group signature and broadcast encryption techniques to achieve user anonymity while maintaining data confidentiality. Similarly, Liu et al. [27] combined group signature and dynamic broadcast encryption techniques to design a multi-owner data-sharing scheme. Users can share data with other members without compromising privacy. However, in both schemes, data users require heavy computational overhead when accessing the data.

Pseudonym certificate-based authentication can guarantee device anonymity and is lightweight, so it is often used to authenticate devices with limited computing power. For example, Cui et al. [12] proposed a pseudonym certificate-based batch authentication scheme, which can meet the security and efficiency needs of IIoT systems. In [28], Jiang et al. proposed a data-sharing scheme based on proxy re-encryption techniques to achieve data confidentiality and secure access control. In addition, the scheme uses pseudonyms and identity-based signatures to guarantee data user anonymity and data integrity. However, in schemes [12] and [28], the smart device needs to pre-store a large number of pseudonym certificates and periodically request new certificates from the key distribution center. For resource-constrained smart devices, managing certificates is complex and requires many additional storage resources.

Although some schemes support smart devices to generate pseudonyms by themselves to improve the efficiency of managing pseudonym certificates and reduce the storage pressure on smart devices, there are still security or efficiency issues. For example, in the scheme [29], requesting pseudonyms requires frequent interaction between multiple entities, which can cause large time delays. In [30], although smart devices do not have to store many pseudonyms in advance, pseudonym generation requires the assistance of managers and blockchains. In [31], to generate pseudonyms by themselves, smart devices hold the system primary key. However, if a smart device is revoked or the primary key in one smart device is compromised, it threatens the entire system's security. In [32], Zhong et al. designed a broadcast encryption scheme for ad hoc networks. In this scheme, the authors utilized proxy re-encryption and broadcast encryption techniques to achieve secure data sharing. Moreover, in this scheme, the smart device can generate pseudonyms by itself. However, the resource-limited device in this scheme performs many time-consuming cryptographic operations and generates significant computational overhead. In [33], Xiong etal. designed a privacy-preserving authentication protocol for IIoT using proxy re-signatures. However, to ensure the anonymity of the smart device, the signature generated by the smart device needs to be sent over a secure channel. In [34], Li etal. designed a blockchain-based anonymous aggregate signature scheme for the IIoT. Although the scheme can effectively guarantee the anonymity of smart devices, its computational efficiency still needs to be further improved.

According to the above analysis, we find that existing anonymous authentication schemes require high computational overhead or storage overhead. Therefore, these authentication schemes are difficult to apply directly to resource-constrained smart devices.

III. Preliminaries

In this section, we review blockchain, bilinear map, and complexity assumptions related to the proposed scheme.

A. Blockchain

Blockchain is a distributed peer-to-peer network, which can also be considered as a shared distributed ledger [35]. Blockchain stores records in multiple distributed nodes to achieve data integrity and decentralization. In addition, each of its blocks is in the form of a Merkle tree to record multiple sets of transactions, which will be difficult to be tampered with once the transactions are recorded on the blockchain [36].

To achieve scalability and efficiency [37] of the data-sharing framework based on blockchain, the proposed scheme uses the consortium blockchain accessible only to authorized organizations.

B. Bilinear Map

In the proposed scheme, bilinear pairing [38] is involved in broadcast encryption and proxy re-encryption. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of order q. Denote g as a generator of $\mathbb{G}$. Let $\mathbb{Z}_q^*$ be a finite field. The map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ denotes a bilinear map with the following properties.

- **Bilinearity:** $e(g^a, h^b) = e(g^b, h^a) = e(g, h)^{ab}$, for any $a, b \in \mathbb{Z}_q^*$ and $g, h \in \mathbb{G}$.
- **Non-Degeneracy:** $e(g, g) \neq 1$.
- **Computablity:** $e(g, h)$ can be efficiently calculated, for any $g, h \in \mathbb{G}$.

C. Complexity Assumptions

The security of the proposed scheme relies mainly on two hardness assumptions. One is the discrete logarithm (DL) assumption, and the other is the computational Diffie-Hellman (CDH) assumption.

- **DL Problem:** Let g be the generator of $\mathbb{G}$. For $A \in \mathbb{G}$ and $A = g^a$, where $a \in \mathbb{Z}_q^*$. Given g and A, the DL problem is to compute a. If solving the DL problem is computationally infeasible, the DL assumption holds.
- **CDH Problem:** Let g be the generator of $\mathbb{G}$. For $g^a, g^b \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q^*$. Given g^a and g^b, the CDH problem is to compute $g^{ab} \in \mathbb{G}$. If solving the CDH problem is computationally infeasible, the CDH assumption holds.

IV. Background

In this section, we introduce the system mode, the high-level workflows, and the security objectives.

A. System Model

Take the example of a smart factory. To accomplish customized production, smart devices in administrative domain B need to access data in administrative domain A with the assistance of edge servers. Fig. 2 illustrates the blockchain-based data-sharing model. The system model consists of the following five entities: domain authorization center (DAC), edge server (ES), smart device (SD), cloud server (CS), and blockchain (BC). In this system model, blockchain acts as a trusted platform that is composed of edge servers in multiple administrative domains. The platform supports smart contracts and consensus mechanisms, promoting trust-building between various domains. Moreover, the platform is distributed and protected by cryptographic algorithms, thus providing tamper-proof and verifiable data storage, further facilitating secure cross-domain data sharing.

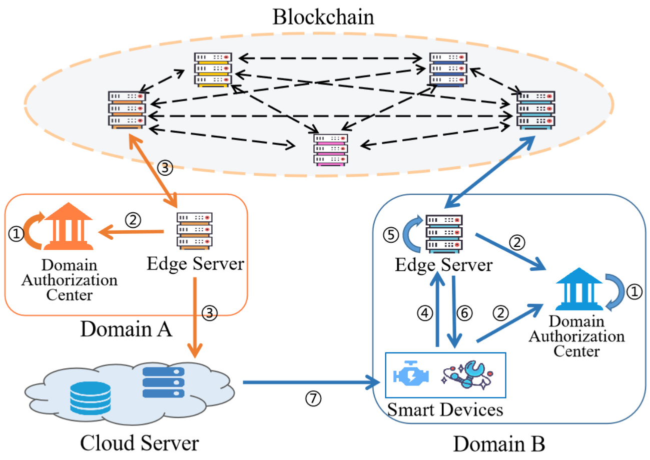


Fig. 2. System model.

*1) **DAC**:* Each administrative domain has a DAC, which is responsible for the registration and authorization of all entities in the domain.

*2) **ES**:* ESs participate in smart device authentication and data sharing. Each ES can act as a data owner. This paper assumes that the ES in domain A is the data owner.

*3) **SD**:* There are many smart devices in each administrative domain. They have limited computing power and storage capacity. Smart devices request and use original data based on the IIoT services they subscribe to.

*4) **CS**:* CS is a third-party server with sufficient storage space and is mainly responsible for storing shared data.

*5) **BC**:* The blockchain is a distributed ledger maintained by ESs in different administrative domains. It is mainly responsible for storing metadata generated by the data owner, and the metadata can be shared with each valid ES.

B. High-Level Workflows

The high-level workflows consist of the following seven phases.

① System setup. For each administrative domain, DAC generates the system parameters for the domain to which it belongs.

② IIoT entities registration. ES and SD are registered at the DAC, respectively.

③ Data encryption and storage. Assume that ES^A is the data owner. After the original data is generated, ES^A encrypts the data into the cloud server and uploads the corresponding metadata onto the blockchain.

④ Sign the request message. When a smart device in domain B wants to obtain shared data, it generates a data request message. Then, the smart device signs the message and sends it to the ES^B.

⑤ **Verify the request message.** Once ES^B receives a message from a smart device, ES^B verifies the message.

⑥ **Transformation key generation and data sharing.** When the access request from the smart device is legitimate, ES^B generates the transformation key and transformed ciphertext. Then, ES^B shares the ciphertext with the corresponding smart device.

⑦ **Data decryption.** The smart device decrypts the transformed ciphertext to obtain the key and storage index of the encrypted original data. Then, the smart device retrieves the encrypted original data from the cloud server and decrypts it to obtain the original data.

C. Security Objectives

The proposed scheme aims to achieve the following security objectives.

*1) **Correctness**:* The correctness consists of three aspects. The first aspect indicates that ES can correctly verify the legitimacy of smart devices' access requests. The second aspect indicates that ES can correctly batch verify multiple legitimate access requests. The last aspect indicates that ES and SD subscribed to the corresponding IIoT service can obtain the plaintext by decrypting.

2) Confidentiality: The confidentiality of original data should be protected from attackers.

3) Access Control: The scheme should ensure that only SD successfully subscribed to the IIoT service can request the corresponding original data.

4) Anonymity: A smart device's real identity cannot be accessed by entities other than the DAC and ES of the domain in which the smart device is located.

5) Un-Linkability: An attacker cannot tell if intercepted data requests are generated by the same smart device.

6) Resistance to Common Attacks: Our scheme can resist several common types of attacks, such as replay, modification, and impersonation attacks, to secure smart devices when requesting data.

V. Proposed Scheme

Inspired by previous encryption techniques, including broadcast encryption [14] and proxy re-encryption [26], we propose a cross-domain data sharing scheme. In this section, we give the specific details of the scheme. Some notations are listed in Table I.

Our proposed scheme has the following three advantages:

- The proposed scheme uses broadcast encryption and proxy re-encryption algorithms to achieve expandable and flexible cross-domain data sharing. That is, data can be shared securely and efficiently to any smart device that has access credentials.
- Smart devices can generate pseudonyms by themselves and don't need to request pseudonyms from the DAC. On the one hand, it effectively reduces the workload of the DAC. On the other hand, smart devices do not have to store a large number of pseudonyms in advance, thus saving storage resources and reducing storage pressure.
- ES^B can perform batch verification, thus increasing the verification efficiency of smart device' request messages.

TABLE I

Notations and Definitions Used

Notations	Definitions
DAC^X	Domain authorization center in domain X
ES^X	Edge server in domain X
SD_i	$i-th$ smart device
CS	Cloud server
s/P_{pub}	System secret/public key in domain B
ω	Access credential
EID_a/EID_b	Real identity of ES^A/ES^B
lsk_{esb}/LPK_{esb}	Long secret/public key of ES^B
lsk_{esa}/LPK_{esa}	Long secret/public key of ES^A
RID_i	Real identity of SD_i
lsk_i/LPK_i	Long secret/public key of SD_i
ask_i/APK_i	Anonymous secret/public key of SD_i
PID_i	Pseudonym of SD_i
σ_i	Signature of SD_i
M_{Serv}	Request message corresponding to $Serv$
Enc/Dec	Symmetric encryption/decryption

A. System Setup

In each administrative domain, the DAC is responsible for completing the system setup [7]. For example, in administrative domain B, DAC^B needs to perform the following steps.

1) Input a security parameter ψ, DAC^B chooses a bilinear map: $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where $\mathbb{G}$ and $\mathbb{G}_T$ are bilinear groups with the order q. Assume that g is a generator of $\mathbb{G}$.
2) DAC^B selects four secure hash functions: $H_1 : \{0, 1\}^* \to \mathbb{Z}_q^*$, $H_2 : \{0, 1\}^* \to \mathbb{G}$, $H_3 : \mathbb{G}_T \to \{0, 1\}^*$, $H_4 : \{0, 1\}^* \to \{0, 1\}^*$.
3) DAC^B chooses a random number $s \in \mathbb{Z}_q^*$ as system secret key in domain B, then DAC^B calculates the corresponding system public key $P_{pub} = g^s$.
4) For the IIoT service $Serv$, DAC^B generates the access credential $\omega = H_1(Serv, s, T_c)$, where T_c indicates current timestamps. Then DAC^B computes $W = g^{\omega}$.
5) DAC^B keeps s secretly and broadcasts system parameters $Para = \{q, \mathbb{G}, \mathbb{G}_T, e, g, P_{pub}, H_1, H_2, H_3, H_4, W\}$.

Remark 1: There are multiple IIoT services in the IIoT system, and the corresponding access credentials are different. For the convenience of description, we discuss only one type of IIoT service in this article.

B. IIoT Entities Registration

This phase contains mainly the registration of ES and SD.

- **The ES registration**

In the administrative domain B, the registration process for ES^B is as follows.

1) ES^B sends its real identity EID_b to DAC^B.
2) Once a registration request is received from ES^B, DAC^B chooses a random number $lsk_{esb} \in \mathbb{Z}_q^*$ as the long secret key of ES^B.

3) DAC^B computes $LPK_{esb} = g^{lsk_{esb}}$ as the long public key for ES^B.
4) DAC^B returns $\{lsk_{esb}, LPK_{esb}, W, Serv\}$ to ES^B and ES^B keeps lsk_{esb} secretly.

It is worth noting that the ES is registered in the same way in other administrative domains. For example, in administrative domain A, ES^A obtains the secret key $lsk_{esa} \in \mathbb{Z}_q^*$ and the public key $LPK_{esa} = g^{lsk_{esa}}$ by registration.

- **The SD registration**

The registration process for the smart device SD_i in administrative domain B is as follows.

1) SD_i sends its identity RID_i to DAC^B.
2) Once a registration request is received from SD_i, DAC^B chooses a random number $lsk_i \in \mathbb{Z}_q^*$ as the long secret key of SD_i.
3) DAC^B computes $LPK_i = g^{lsk_i}$ as the long public key for SD_i.
4) DAC^B returns $\{LPK_i, lsk_i, \omega, Serv\}$ to SD_i and SD_i keeps $\{lsk_i, \omega\}$ secretly. Then DAC^B sends LPK_i to ES^B.

C. Data Encryption and Storage

Assume that both ES^A and ES^B subscribe to the service $Serv$. When ES^A in administrative domain A wants to share data, ES^A first encrypts it and sends the ciphertext to the cloud server. Next, ES^A runs the broadcast encryption algorithm to encrypt the relevant parameters (including the storage address and encryption key) associated with the encrypted data and then uploads them to the blockchain.

1) ES^A calculates $h_{eid,b} = H_2(EID_b)$ and then computes $RSK_{a,b} = (LPK_{esb})^{lsk_{esa}}$. Finally, ES^A calculates $r_{a,b} = H_1(RSK_{a,b}, Serv)$, $SK_{a,b} = (h_{eid,b})^{r_{a,b}}$ and $y_b = g^{r_{a,b}}$.
2) ES^A randomly selects a key $k \in \mathbb{Z}_q^*$ and performs a symmetric encryption algorithm $ED = Enc_k(m)$ on the original data m.
3) ES^A stores the ED in the cloud server and gets the corresponding storage index $index$. Then ES^A computes $h_{ed} = H_4(ED)$.
4) ES^A selects the current timestamp t_{esa} and computes $r = H_1(index, k, t_{esa}, h_{ed})$. Then ES^A computes $C_0 = g^r$.
5) ES^A computes $C_b = (C_{b,1}, C_{b,2})$, where $C_{b,1} = (k||index||h_{ed}||t_{esa}) \oplus H_3(e(h_{eid,b}, y_b)^r)$, $C_{b,2} = H_2(C_0, C_{b,1})^r$.
6) ES^A generates $Hdr = (C_0, C_b)$ and sets metadata $CT = (Serv, EID_a, Hdr, t_{esa})$.
7) ES^A uploads CT to the blockchain.

Remark 2: (Our Proposed Scheme is Extensible and Flexible): If l ESs besides ES^A subscribe to the same service, ES^A runs broadcast encryption to generate $CT = (Serv, EID_a, Hdr, t_{esa})$, where $Hdr = (C_0, C_1, \ldots, C_b, \ldots, C_l) = (C_0, C_{1,1}, C_{1,2}, \ldots, C_{b,1}, C_{b,2}, \ldots, C_{l,1}, C_{l,2})$. In addition, if a new ES_{new} subscribes to the service $Serv$, the ES^A can only generate $C_{new} = \{C_{new,1}, C_{new,2}\}$ without changing the entire ciphertext.

D. Sign the Request Message

When the smart device SD_i wants to request corresponding data of ES^A, it signs the request message and sends it to the ES^B. The process of signing a request message is as follows.

1) SD_i generates the shared secret key SSK_i for ES^B by calculating $SSK_i = (LPK_{esb})^{lsk_i}$.
2) SD_i selects an anonymous secret key $ask_i \in \mathbb{Z}_q^*$, and then computes anonymous public key $APK_i = g^{ask_i}$.
3) SD_i calculates $TSK_i = (LPK_{esb})^{ask_i}$ as a temporary secret key, which SD_i and ES^B can obtain by calculation. Then, SD_i calculates $PID_i = LPK_i + TSK_i$ as its pseudonym.
4) SD_i chooses current timestamps t_i and then calculates $\theta_i = H_1(W, t_i, M_{Serv}, TSK_i, SSK_i, APK_i, PID_i, EID_a)$, where M_{Serv} indicates data request message. Subsequently, SD_i obtains the signature by calculating $\sigma_i = ask_i + w \cdot \theta_i$.
5) SD_i sends msg_i to ES^B, where $msg_i = \{M_{Serv}, APK_i, PID_i, t_i, \sigma_i, EID_a\}$.

Remark 3: DAC^B does not need to participate in the authentication process of smart devices online. Because SD_i can generate pseudonyms by itself and without requesting and storing large numbers of pseudonyms from DAC^B.

E. Verify the Request Message

Upon receiving the message msg_i from SD_i, ES^B performs the following steps to verify the legitimacy of the requested message.

1) ES^B checks the freshness of timestamp t_i and discards the message if the timestamp has been expired. Otherwise, ES^B computes $TSK_i' = (APK_i)^{lsk_{esb}}$.
2) ES^B obtains the long public key of SD_i by calculating $LPK_i = PID_i - TSK_i'$. Then ES^B checks whether the LPK_i exists in the local database. If LPK_i does not exist, ES^B discards the message msg_i directly. Otherwise, ES^B performs the next step.
3) ES^B calculates $SSK_i' = (LPK_i)^{lsk_{esb}}$.
4) ES^B queries W based on M_{Serv} and then calculates $\theta_i' = H_1(W, t_i, M_{Serv}, TSK_i', SSK_i', APK_i, PID_i, EID_a)$.
5) ES^B verifies whether the equation $g^{\sigma_i} = APK_i \cdot (W)^{\theta_i'}$ holds. If it does not hold, ES^B discards msg_i directly; otherwise, ES^B accepts it. The correctness is proved as follows.

$$g^{\sigma_i} = g^{ask_i + w\cdot\theta_i} = g^{ask_i} \cdot (g^{\omega})^{\theta_i} = APK_i \cdot W^{\theta_i}. \quad (1)$$

The Proposed Scheme Supports Batch Validation: After checking, suppose there are still n messages that satisfy the timestamp to be valid. Assume that the n messages are $msg_1 = \{M_{Serv}, APK_1, PID_1, t_1, \sigma_1, EID_a\}$, $msg_2 = \{M_{Serv}, APK_2, PID_2, t_2, \sigma_2, EID_a\}, \ldots, msg_n = \{M_{Serv}, APK_n, PID_n, t_n, \sigma_n, EID_a\}$. ES^B performs batch verification after obtaining $\theta_i (i = 1, 2, \ldots, n)$.

1) To ensure non-repudiation [31], ES^B chooses a vector $v = \{v_1, v_2, \ldots, v_n\}$, where $v_i \in [1, 2^{\tau}]$ and τ is a small integer.

2) ES^B batch verifies these messages by calculating the following equation.

$$g^{\sum_{i=1}^{n}(v_i \cdot \sigma_i)} = \prod_{i=1}^{n}(APK_i^{v_i}) \cdot W^{(\sum_{i=1}^{n}(v_i \cdot \theta_i))}. \tag{2}$$

If this equation holds, ES^B accepts these messages; otherwise, ES^B drops these messages.

The correctness of equation (2) is shown in equation (3).

$$\begin{aligned} g^{\sum_{i=1}^{n}(v_i \cdot \sigma_i)} &= g^{\sum_{i=1}^{n}(v_i \cdot (ask_i + \omega \cdot \theta_i))} \\ &= g^{\sum_{i=1}^{n}(v_i \cdot ask_i + v_i \cdot \omega \cdot \theta_i))} \\ &= g^{\sum_{i=1}^{n}(v_i \cdot ask_i)} \cdot g^{\sum_{i=1}^{n}(v_i \cdot \omega \cdot \theta_i))} \\ &= \prod_{i=1}^{n}(APK_i^{v_i}) \cdot W^{(\sum_{i=1}^{n}(v_i \cdot \theta_i))}. \end{aligned} \tag{3}$$

Remark 4: Based on the CDH problem, we can know that only ES^B can verify access requests from SD_i. Because verifying the legitimacy of access request needs TSK_i and SSK_i, and besides SD_i, only ES^B can obtain these two keys by computation.

F. Transformation Key Generation and Data Sharing

Suppose the access request of SD_i is legitimate. In that case, ES^B obtains data (C_0, C_b, t_{esa}) in the blockchain through the service $Serv$ and EID_a, then ES^B performs the following steps to generate the corresponding transformation key and transformed ciphertext.

1) ES^B calculates $RSK'_{a,b} = (LPK_{esa})^{lsk_{esb}}$, $r'_{a,b} = H_1(RSK'_{a,b}, Serv)$ and $SK'_{a,b} = (h'_{eid,b})^{r'_{a,b}}$, where $h'_{eid,b} = H_2(EID_b)$.
2) ES^B calculates $r_{b,i} = H_1(SSK'_i, Serv)$ and $SK_{b,i} = (h_i)^{r_{b,i}}$, where $h_i = H_2(APK_i)$.
3) ES^B chooses current timestamp t_{esb} and then calculates $TK_{b\to i} = SK_{a,b} \cdot g^{H_1(SK_{b,i}, Serv, t_{esb})}$ as the transformation key.
4) ES^B calculates $C'_0 = e(g, C_0)$ and $C'_{b,2} = e(C_0, TK_{b\to i})$.
5) ES^B generates the transformed ciphertext $CT_{b\to i} = (C'_0, C_{b,1}, C'_{b,2})$ and then sends $\{CT_{b\to i}, t_{esb}\}$ to SD_i.

G. Data Decryption

Upon receiving the message $\{CT_{b\to i}, t_{esb}\}$ from the ES^B, SD_i performs decryption operations to obtain the corresponding original data. The specific performing steps are as follows.

1) SD_i calculates $h'_i = H_2(APK_i)$, $r'_{b,i} = H_1(SSK_i, Serv)$ and $SK'_{b,i} = (h'_i)^{r'_{b,i}}$.
2) SD_i calculates $(k||index||h_{ed}||t_{esa}) = H_3(\frac{C'_{b,2}}{(C'_0)^{H_1(SK'_{b,i}, Serv, t_{esb})}}) \oplus C_{b,1}$. The correctness is shown in equation (4).

$$\begin{aligned} &H_3(\frac{C'_{b,2}}{(C'_0)^{H_1(SK'_{b,i}, Serv, t_{esb})}}) \oplus C_{b,1} \\ &= H_3(\frac{e(C_0, TK_{b\to i})}{(e(g, C_0))^{H_1(SK_{b,i}, Serv, t_{esb})}}) \\ &\quad \oplus ((k||index||h_{ed}||t_{esa}) \oplus H_3(e(h_{eid,b}, y_j)^r)) \\ &= H_3(\frac{e(g^r, (h_{eid,b})^{r_{a,b}} \cdot g^{H_1(SK_{b,i}, Serv, t_{esb})})}{(e(g, g^r))^{H_1(SK_{b,i}, Serv, t_{esb})}}) \\ &\quad \oplus ((k||index||h_{ed}||t_{esa}) \oplus H_3(e(h_{eid,b}, g^{r_{a,b}})^r)) \\ &= H_3(e(g^r, (h_{eid,b})^{r_{a,b}}) \oplus (k||index||h_{ed}||t_{esa}) \\ &\quad \oplus H_3(e(h_{eid,b}, g^{r_{a,b}})^r)) \\ &= (k||index||h_{ed}||t_{esa}) \end{aligned} \tag{4}$$

3) SD_i uses $index$ to obtain ED in the CS and then calculates $h_{ed} = H_4(ED)$. Finally, SD_i determines whether the equation $C'_0 = e(g, g)^{H_1(index, k, t_{esa}, h_{ed})}$ holds. If yes, SD_i accepts $\{k, ED\}$ and calculates $Dec_k(ED)$ to obtain original data m.

Remark 5: (Efficiency Improvement): To further improve efficiency, some cryptographic operations can be performed offline in our proposed scheme. For example, in the data encryption and storage phase, the generation of $\{h_{eid}, RSK_{a,b}, r_{a,b}, SK_{a,b}, y_b\}$ doesn't need the online original data m, so ES^A can generate $\{h_{eid}, RSK_{a,b}, r_{a,b}, SK_{a,b}, y_b\}$ in advance and store them for the future generation of online metadata CT.

VI. Security Proof and Analysis

In the proposed scheme, we design an anonymous authentication algorithm and a data sharing protocol. This data sharing protocol combines broadcast encryption and proxy re-encryption techniques. The broadcast encryption and proxy re-encryption involved in the proposed scheme are improvements on scheme [14] and scheme [26], so the security proof for this part can refer to [14] and [26]. The signatures in the anonymous authentication algorithm are lightweight and without pairing. For the signature algorithm, there are three types of adversaries. The first type of adversary does not have w, lsk_i, and lsk_{esb}. The second type of adversary has w but no lsk_i and lsk_{esb}. The third type of adversary has lsk_{esb}, but no w and lsk_i. According to CDH problem and Remark 4, the first type of adversary and the second type of adversary cannot compute TSK_i and SSK_i, so they cannot successfully forge a valid signature of SD_i. Therefore, in the security proof, we focus on the third type of adversary.

A. Security Model

The security of the proposed signature algorithm is defined by a game played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. This game has the following queries.

- *Setup-query:* For this query, $\mathcal{C}$ generates system secret key and some public parameters. Then, $\mathcal{C}$ sends public parameters to $\mathcal{A}$.
- *H_1-query:* When $\mathcal{A}$ makes this query with a random string str, challenger $\mathcal{C}$ returns the corresponding Hash value to $\mathcal{A}$.
- *Sign-query:* When $\mathcal{A}$ makes this query with $< M_{Serv}, LPK_i, SSK_i, LPK_{esb}, t_i, EID_a >$, the challenger $\mathcal{C}$ generates the corresponding signature tuple msg_i and then returns it to $\mathcal{A}$.

After the above queries is executed, $\mathcal{A}$ forges a signature σ_i' associated with $< M_{Serv}, LPK_i, SSK_i, LPK_{esb}, t_i, EID_a >$. $\mathcal{A}$ wins the game if the following conditions hold.

- $\mathcal{A}$ did not invoke the *Sign-query* corresponding to M_{Serv}.
- The forged signature σ_i' is valid after verification.

Definition 1: The proposed signature algorithm is secure against existential forgery under an adaptive chosen message attack if the advantage of breaking the proposed signature is negligible for any polynomial-time adversary $\mathcal{A}$.

B. Security Proof

According to *Definition 1*, this subsection demonstrates that our proposed signature algorithm is secure against the adaptive chosen message attack.

Theorem 1: If the DL problem is intractable, the proposed signature algorithm is secure in the random oracle model.

Proof: If the polynomial-time $\mathcal{A}$ can forge a legal message $msg_i = \{PID_i, t_i, \sigma_i, M_{Serv}, EID_a, APK_i\}$ with a non-negligible advantage ϵ, then the $\mathcal{C}$ has the ability to solve the DL problem executed by $\mathcal{A}$ as a subroutine with ϵ. Given (g, g^{ω}) as the instance of DL problem. $\mathcal{C}$ responds the oracle queries by $\mathcal{A}$ as follows.

- ***Setup-query:*** Upon receiving the setup query from $\mathcal{A}$, $\mathcal{C}$ sets $\omega \in \mathbb{Z}_q^*$ as the secret key and computes $W = g^{\omega}$. Then $\mathcal{C}$ returns the public parameters $\{q, \mathbb{G}, \mathbb{G}_T, e, g, H_1, W\}$ to $\mathcal{A}$.
- ***H_1-query:*** $\mathcal{C}$ presets a map Map_{H_1}. When $\mathcal{A}$ invokes this query with $\{W, t_i, M_{Serv}, TSK_i, SSK_i, APK_i, PID_i, EID_a\}$, $\mathcal{C}$ checks whether the tuple $< W, t_i, M_{Serv}, TSK_i, SSK_i, APK_i, PID_i, EID_a >$ is already contained in the map Map_{H_1} or not. If so, $\mathcal{C}$ sends $h_1 = H_1(W, t_i, M_{Serv}, TSK_i, SSK_i, APK_i, PID_i, EID_a)$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ chooses a random number $\tau_{H_1} \in \mathbb{Z}_q^*$ and then adds $< W, t_i, M_{Serv}, TSK_i, SSK_i, APK_i, PID_i, EID_a >$ into the map Map_{H_1}. Finally, $\mathcal{C}$ sends τ_{H_1} to $\mathcal{A}$.
- ***Sign-query:*** When $\mathcal{A}$ invokes this query using $< M_{Serv}, LPK_i, SSK_i, LPK_{esb}, t_i, EID_a >$, $\mathcal{C}$ generates a random number $ask_i \in \mathbb{Z}_q^*$ and $ask_i = \sigma_i - \omega \cdot \theta_i$. Subsequently, $\mathcal{C}$ computes $APK_i = g^{ask_i}$, $TSK_i = LPK_{esb}^{ask_i}$, $PID_i = LPK_i + TSK_i$, $\theta_i = H_1(W, t_i, M_{Serv}, TSK_i, SSK_i, APK_i, PID_i, EID_a)$, and $\sigma_i = ask_i + \omega \cdot \theta_i$. Finally, $\mathcal{C}$ sends msg_i to $\mathcal{A}$, where $msg_i = \{M_{Serv}, APK_i, PID_i, t_i, EID_a, \sigma_i\}$.

After the above queries is executed, $\mathcal{A}$ outputs a forged signature tuple $msg_i = \{M_{Serv}, APK_i, PID_i, t_i, EID_a, \sigma_i\}$. And $\mathcal{C}$ determines whether the equation (5) holds.

$$g^{\sigma_i} = APK_i \cdot W^{\theta_i}. \tag{5}$$

If the equation (5) does not hold, $\mathcal{C}$ aborts the process. Based on the forking lemma [39], if $\mathcal{A}$ repeats the above step with a different choice of H_1, $\mathcal{A}$ can output another valid signature tuple $msg_i = \{M_{Serv}, APK_i, PID_i, t_i, EID_a, \theta_i', \sigma_i'\}$ with the probability $\epsilon' \geq (1/9)$ [40]. We easily obtain equation (6).

$$g^{\sigma_i'} = APK_i \cdot W^{\theta_i'}. \tag{6}$$

According to equations (5) and (6), we can obtain the equation (7).

$$g^{\sigma_i' - \sigma_i} = g^{(ask_i + \omega \cdot \theta_i') - (ask_i + \omega \cdot \theta_i)} = g^{\omega \cdot (\theta_i' - \theta_i)}. \tag{7}$$

Based on the equation (7), we can get the equation (8).

$$\sigma_i' - \sigma_i = \omega \cdot (\theta_i' - \theta_i). \tag{8}$$

According to the equation (8), $\mathcal{A}$ outputs $(\sigma_i' - \sigma_i) \cdot (\theta_i' - \theta_i)^{-1}$ as a solution for the given instance of the DL problem. $\mathcal{C}$ can solve the DL problem depending on whether the following two events occur simultaneously.

- E_{Serv} indicates that $Serv$ is equal to $Serv'$.
- E_{forge} indicates that $\mathcal{A}$ can forge two legitimate signatures.

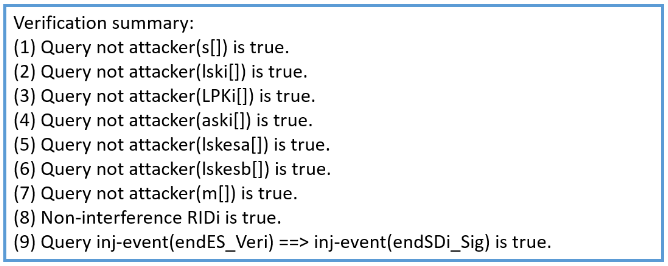
Let N_{H_1} indicates the number of H_1-query executed, then $Prob[E_{Serv}] = \frac{1}{N_{H_1}}$ and $Prob[E_{forge|E_{Serv}}] \geq \frac{1}{9} \cdot \epsilon$. Finally, we can obtain that

$$\begin{aligned} Prob[E_{forge} \wedge E_{Serv}] &= Prob[E_{forge}|E_{Serv}] \cdot Prob[E_{Serv}] \\ &\geq \frac{1}{9} \cdot \epsilon \cdot \frac{1}{N_{H_1}}. \end{aligned} \tag{9}$$

In summary, $\mathcal{C}$ can solve the DL problem with a non-negligible advantage $\frac{\epsilon}{9N_{H_1}}$. However, this contradicts the fact that the DL problem is hard to solve in polynomial time. Therefore, the signature algorithm is secure.

C. Security Analysis

1) ***Confidentiality:*** The original data is always stored and transmitted in the ciphertext. According to the security of broadcast encryption and proxy re-encryption, the corresponding original data is not available to entities other than authorized ESs and SDs.
2) ***Access control:*** Firstly, the security of broadcast cryptography ensures that only ESs subscribed to the relevant IIoT service can convert the ciphertext using proxy re-encrypt. Secondly, when an access request is received from SD_i, ES^B can verify the legitimacy of SD_i's identity and the validity of the request, i.e., only for smart devices where LPK_i is legitimate and does have access credentials ω, ES^B will return the corresponding transformed ciphertext. Therefore, our proposed scheme achieves efficient access control.
3) ***Anonymity:*** In our proposed scheme, SD uses a dynamically updated pseudonym $PID_i = LPK_i + (LPK_{esb})^{ask_i}$ in each communication, and LPK_i is hidden in PID_i. The attacker cannot obtain lsk_{esb} and cannot break the CDH problem, so the attacker cannot obtain the real identity of the smart device. Therefore, our proposed scheme achieves the anonymity of smart devices.
4) ***Un-linkability:*** In each request for data, SD_i generates the corresponding ask_i as an anonymous secret key and the corresponding pseudonym $PID_i = LPK_i + (LPK_{esb})^{lsk_i}$. Because ask_i is random and unlinkable, two messages of the same smart device are unlinkable.



```
Verification summary:
(1) Query not attacker(s[]) is true.
(2) Query not attacker(lski[]) is true.
(3) Query not attacker(LPKi[]) is true.
(4) Query not attacker(aski[]) is true.
(5) Query not attacker(lskesa[]) is true.
(6) Query not attacker(lskesb[]) is true.
(7) Query not attacker(m[]) is true.
(8) Non-interference RIDi is true.
(9) Query inj-event(endES_Veri) ==> inj-event(endSDi_Sig) is true.
```

Fig. 3. Obtained results from the ProVerif tool.

5) ***Resistance replay attack:*** The message sent by SD_i contains a timestamp t_i; ES determines whether a received message is a replay message by checking whether the timestamp has expired.
6) ***Resistance modification attack:*** Once the ES discovers that the message has been tampered with, i.e., if the message verification fails, the ES will drop the message directly.
7) ***Resistance impersonation attack:*** According to the security proof, we can know that the attacker cannot obtain lsk_i or ω, so the attacker cannot generate a legitimate signature $\sigma_i = \omega + ask_i \cdot \theta_i$.

D. Formal Analysis Using Proverif

To further demonstrate the security of our scheme, we use ProVerif, an automated tool for formal analysis and verification of security protocols. The source code is opened,[1] and the results are shown in Fig. 3.

In Fig. 3, (1)-(7) indicate that no attacker can obtain $\{s, lsk_i, LPK_i, ask_i, lsk_{esa}, lsk_{esb}, m\}$. In Fig. 3, (8) shows the result for two observation equivalents. The result indicates RID_i is anonymous, i.e., the smart device is anonymous. In Fig. 3, (9) shows the result of two injective correspondence assertions. This result indicates that our protocol enables the verification of ES^B to SD_i.

E. Comparison of Security and Functionality Features

We introduce the related schemes [11], [13], [32], [33], and [34] into the IIoT scenarios we focus on and compare them in terms of security and functionality. As shown in Table II, our proposed scheme is more advantageous compared to the related schemes [11], [13], [33], [34]. Although scheme [32] satisfies all the security and functionality requirements, our proposed scheme requires lower computational overhead and communication overhead compared to that scheme.

VII. Performance Analysis

A. Experimental Setting

We compare the proposed scheme with some related schemes [11], [13], [32], [33], [34]. To make the performance evaluation more fair and convenient, we introduce the algorithms from related schemes into our proposed system model. That is, ES^A performs the data encryption operation, ES^B performs the message verification and data re-encryption operation, and SD_i performs the message signature and data decryption operation.

TABLE II
Comparison of Security and Functionality Features

	[13]	[11]	[32]	[33]	[34]	Our
Confidentiality	✓	✓	✓	×	×	✓
Anonymity	✓	✓	✓	✓	✓	✓
Un-linkability	×	✓	✓	✓	✓	✓
Access control	✓	✓	✓	✓	✓	✓
Resistance replay attack	×	✓	✓	✓	×	✓
Resistance modification attack	×	✓	✓	✓	✓	✓
Resistance impersonation attack	×	✓	✓	✓	✓	✓
Pseudonyms generated by SD	×	×	✓	✓	✓	✓

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

TABLE III
Execution Time of Several Operations

Symbol	Description	Time$_1$ (ms)	Time$_2$ (ms)
T_{bp}	Bilinear pairing operation	10.932	54.387
T_m	Scale multiplication operation in $\mathbb{G}$	5.619	26.113
T_{sm}	Multiplication with small factor in $\mathbb{G}$	0.443	2.080
T_e	Exponentiation operation in $\mathbb{G}_T$	4.971	24.046
T_h	One-way hash operation	0.014	0.027
T_{mtp}	MapToPoint operation in $\mathbb{G}$	0.621	3.302
T_a	Point addition in $\mathbb{G}$	0.010	0.056
T_{gtmul}	Multiplication operation in $\mathbb{G}_T$	0.018	0.100

Time$_1$: The execution time on the PC. Time$_2$: The execution time on the Raspberry Pi 4.

To evaluate the computational costs of the protocols proposed in each scheme, we use the MIRACL cryptography library [41] to obtain cryptographic operations' execution time on the PC (Intel Core i5-7500 CPU @3.4GHz, 16GB RAM, and the Ubuntu 18.04.3 operation system) and the Raspberry Pi 4 (1.5GHz CPU, 4GB RAM, and the Debian GNU/Linux 11 operation system). The execution time on the PC and Raspberry Pi 4 indicates the execution time required by ES and SD, respectively. In the MIRACL library, to realize the symmetric bilinear pairs $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ and achieve 128-bit security level, we choose super-singular curve, eta_T pairing embedding degree 4. The hash function we use is SHA256. Table III lists the execution time of the basic cryptographic operations on each entity.

To evaluate the performance of on-chain operations, we build the Hyperledger platform and the Ethereum platform, respectively. The Hyperledger platform consists of ten PCs with the same performance. Each PC is running Ubuntu 18.04.3, equipped with an Intel Core i7-11700 CPU @2.50GHz and 16GB of RAM. Eight of these ten PCs act as peer nodes, one as an orderer node, and one as a client node. The Ethereum platform is deployed on a PC with the same performance as the Hyperledger platform. Specifically,

[1] https://github.com/ahufqwang/BSCDDS

TABLE IV
LATENCY AND THROUGHPUT

	Send rate (TPS)	Max latency (s)	Min latency (s)	Average latency (s)	Throughput (TPS)
Write	328.4	2.09	0.05	0.11	326.2
Query	3208.3	0.03	0.00	0.01	3208.2

we use Ganache[2] to construct a simulated Ethereum blockchain.

B. On-Chain Overhead

We implement write and query operations on the Hyperledger and Ethereum platforms to evaluate the on-chain overhead based on the experimental settings.

On the Hyperledger platform, we set the number of worker processes to 5, the test time to $300s$, the rate controller to fixed load, and the maximum transaction load to 50. The experimental results are shown in Table IV. Note that the "TPS" indicates "Transactions Per Second". From Table IV, we can see that the average time latency of write and query is 0.11 s and 0.01 s, respectively. Note that the time delay of the query operation is the lowest. This is because the retrieval in the blockchain is performed in the local ledger. In addition, when the sending rate is 328.4 TPS and 3208.3 TPS, the corresponding throughput is 326.2 TPS and 3208.2 TPS, respectively. The throughput completion rates are calculated to be $326.2/328.4 \approx 99.330\%$ and $3208.2/3208.3 \approx 99.997\%$. That is, the throughput completion rates are all more than 95%, which meets actual application requirements.

On the Ethereum platform, gas is usually the unit to measure workloads. Through the experiment, the results show that the gas usage for write operation is 580395 and the gas usage for query operation is 49743. In addition, we record the time of invoking the smart contract corresponding to write operations and query operations. The results show that the time delay of invoking the corresponding smart contract is low. Specifically, the time to invoke a write operation is about 0.069 s and the time to invoke a query operation is about 0.041 s.

C. Computational Overhead

Table V shows the cryptographic operations performed by each scheme. It is worth noting that some cryptographic operations in our proposed scheme can be executed offline. To make the comparison fairer, Table V only records the cryptographic operations that are executed online in all schemes. The "-" in Table V indicates no cryptographic operations.

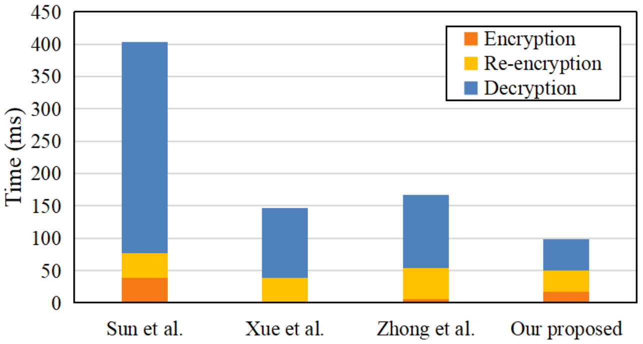


Fig. 4. The computational overhead comparison of encryption, proxy re-encryption, and decryption.

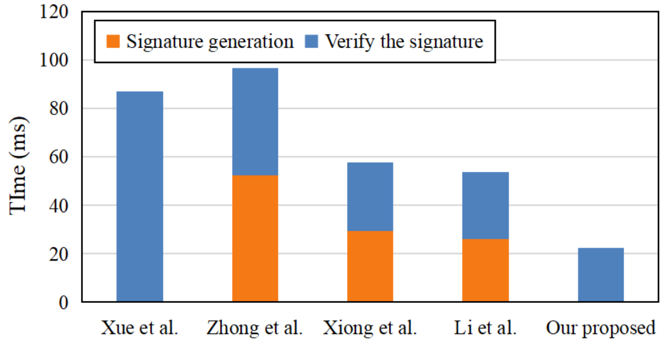


Fig. 5. The computational overhead of signing and verifying.

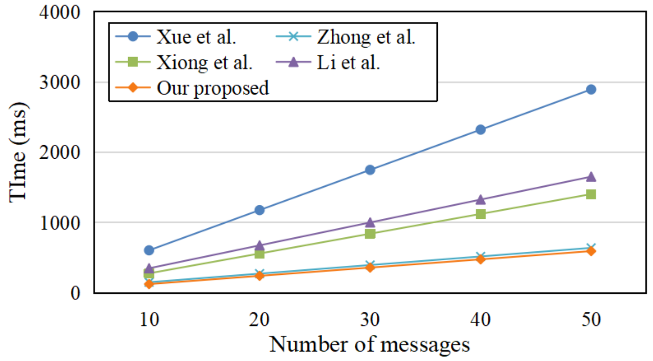


Fig. 6. The computational overhead of batch verification.

Fig. 4 shows the computational overhead comparison of encryption, proxy re-encryption, and decryption. In scheme [13], the time overhead required for encryption, proxy re-encryption, and decryption is $6T_m+T_e+6T_h+2T_a \approx 38.789\ ms$, $3T_{bp}+T_m+2T_{gtmul} \approx 38.451\ ms$, and $4T_{bp}+4T_m+T_{mtp}+4T_{gtmul}+3T_h+2T_a \approx 325.895\ ms$ respectively. In scheme [11], the time overhead required for proxy re-encryption and decryption is $2T_{bp}+T_e+2T_m+T_{gtmul} \approx 38.091\ ms$ and $2T_{bp}+T_{gtmul} \approx 108.874\ ms$ respectively. In scheme [32], the time overhead required for encryption, proxy re-encryption, and decryption is $T_m+T_h+T_{gtmul} \approx 5.651\ ms$, $2T_{bp}+3T_m+2T_e+T_h+T_a+2T_{gtmul} \approx 48.723\ ms$, and $2T_{bp}+T_{mtp}+2T_{gtmul} \approx 112.276\ ms$ respectively.

In our proposed scheme, the computational overhead of performing encryption is $2T_m+T_e+T_{mtp}+3T_h \approx 16.872\ ms$,

[2] https://trufflesuite.com/ganache/

TABLE V

COMPUTATION COST COMPARISON

Scheme	ES^A	ES^B			SD_i	
	Encryption	Re-encryption	Verify the Signature	Batch Verification (n messages)	Signature Generation	Decryption
Sun *et al.* [13]	$6T_m + T_e + 6T_h + 2T_a$	$3T_{bp} + T_m + 2T_{gtmul}$	-	-	-	$4T_{bp} + 4T_m + T_{mtp} + 4T_{gtmul} + 3T_h + 2T_a$
Xue *et al.* [11]	-	$2T_{bp} + T_e + 2T_m + T_{gtmul}$	$2T_{bp}+8T_m+4T_e + 4T_a + 4T_{gtmul}+T_h$	$10nT_m + 2T_{bp} + 3T_e + 2nT_{sm} + 4nT_a + nT_h + 4T_{gtmul}$	T_h	$2T_{bp} + T_{gtmul}$
Zhong *et al.* [32]	$T_m + T_h + T_{gtmul}$	$2T_{bp} + 3T_m + 2T_e + T_h + T_a + 2T_{gtmul}$	$3T_{bp} + 2T_m + 3T_h + T_a + T_{gtmul}$	$3T_{bp} + 2nT_m + 2nT_{sm} + (4n - 3)T_a + 3nT_h + T_{gtmul}$	$2T_m + 2T_h + 2T_a$	$2T_{bp} + T_{mtp} + 2T_{gtmul}$
Xiong *et al.* [33]	-	-	$2T_{bp} + T_m + T_{mtp} + T_a + T_h$	$2nT_{bp} + nT_m + nT_{mtp} + nT_a + nT_h$	$T_{mtp} + T_m$	-
Li *et al.* [34]	-	-	$2T_{bp} + T_m + T_a + T_h$	$(2n + 2)T_{bp} + (n + 1)T_m + nT_e + (n + 2)T_h + (3n - 2)T_a + (2n - 2)T_{gtmul}$	$T_m + T_h + T_a$	-
Our proposed	$2T_m + T_e + T_{mtp} + 3T_h$	$2T_{bp} + T_{mtp} + 2T_m + 2T_h + T_a$	$4T_m + 2T_a + T_h$	$(2n + 2)T_m + 2nT_a + nT_{sm} + nT_h$	T_h	$2T_e+4T_h+T_{gtmul}$

which is higher than that in schemes [11] and [32], but through Fig. 4 we can find that the total time (including the time cost of encryption, proxy re-encryption, and decryption) is the lowest in our proposed scheme. For the proxy re-encryption, the computational overhead in our proposed scheme is about $2T_{bp}+T_{mtp}+2T_m+2T_h+T_a \approx 33.761\ ms$, which is $38.451 - 33.761 = 4.69\ ms$, $38.091 - 33.761 = 4.33\ ms$ and $48.723 - 33.761 = 14.962\ ms$ less than schemes [11], [13], [32], respectively. In our scheme, the time overhead of decryption is $2T_e + 4T_h + T_{gtmul} \approx 48.300\ ms$, which is $325.895 - 48.300 = 277.595\ ms$, $108.874 - 48.300 = 60.574\ ms$, and $112.276 - 48.300 = 63.976\ ms$ less than schemes [11], [13], [32], respectively. In addition, in our scheme, we can calculate that the total time (including the time cost of encryption, proxy re-encryption, and decryption) is $16.872 + 33.761 + 48.3 = 98.933\ ms$, which is about $98.933/(38.789+38.451+325.895) \approx 24.5\%$ of [13], $98.933/(38.091 + 108.874) \approx 67.3\%$ of [11], and $98.933/(5.651 + 48.723 + 112.276) \approx 59.4\%$ of [32].

Fig. 5 shows the computational cost of signing and verifying. Combining with Table V, we get that the time overhead required for signature generation and verification of scheme [11] is $T_h \approx 0.027\ ms$ and $2T_{bp} + 8T_m + 4T_e + 4T_a+4T_{gtmul}+T_h \approx 86.826\ ms$ respectively. In scheme [32], the time overhead required for signature generation and verification is $2T_m + 2T_h + 2T_a \approx 52.392\ ms$ and $3T_{bp} + 2T_m+3T_h+T_a+T_{gtmul} \approx 44.104\ ms$ respectively. In scheme [33], the time overhead required for signature generation and verification is $T_{mtp} + T_m \approx 29.415\ ms$ and $2T_{bp} + T_m + T_{mtp}+T_a+T_h \approx 28.128\ ms$ respectively. In scheme [34], the time cost required for signature generation and verification is $T_m + T_h + T_a \approx 26.196\ ms$ and $2T_{bp} + T_m + T_a + T_h \approx 27.507\ ms$ respectively. As can be seen from Fig. 5, the total computational overhead (including the computational overhead of signing and verification) is the lowest in our proposed scheme.

In addition, our scheme supports batch authentication and the computational cost of batch verification is shown in Fig. 6. From Table V, we can obtain that the computational overhead required for batch authentication in schemes [11], [32], [33], [34] and our scheme is $10nT_m + 2T_{bp} + 3T_e + 2nT_{sm} + 4nT_a + nT_h + 4T_{gtmul} \approx 57.13n + 36.849\ ms$, $3T_{bp} + 2nT_m + 2nT_{sm} + (4n - 3)T_a + 3nT_h + T_{gtmul} \approx 12.206n + 32.784\ ms$, $2nT_{bp} + nT_m + nT_{mtp} + nT_a + nT_h \approx 28.128nms$, $(2n + 2)T_{bp} + (n + 1)T_m + nT_e + (n + 2)T_h + (3n - 2)T_a + (2n - 2)T_{gtmul} \approx 32.534n + 27.455\ ms$, and $(2n+2)T_m+2nT_a+nT_{sm}+nT_h \approx 11.715n+11.238\ ms$. From the Fig. 6, we find that when the number of messages is 10, the time overhead in our proposed scheme is $128.388\ ms$, which is $608.149 - 128.388 = 479.761\ ms$, $154.844 - 128.388 = 26.456\ ms$, $281.28 - 128.388 = 152.892\ ms$, and $352.795 - 128.388 = 224.407\ ms$ less than [11], [32], [33], and [34], respectively. With the number of messages increasing, the time overhead required for batch authentication in our proposed scheme remains at the lowest level. When the number of messages is 50, the time overhead in our proposed scheme is $596.988\ ms$, which is $2893.349 - 596.988 = 2296.361\ ms$, $643.084 - 596.988 = 46.096\ ms$, $1406.6 - 596.988 =$

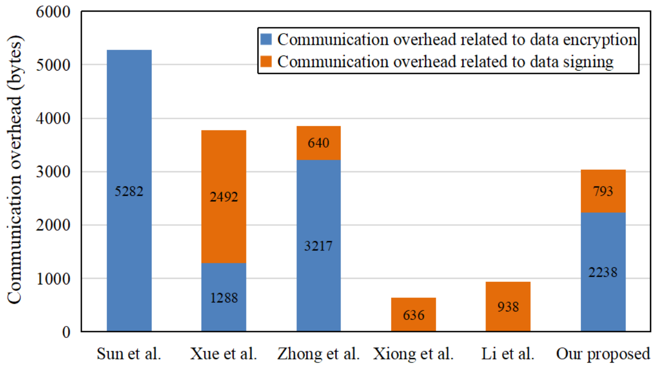


Fig. 7. The communication overhead comparison.

809.412 ms, and $1654.155 - 596.988 = 1057.167$ ms less than [11], [32], [33], and [34], respectively.

D. Communication Overhead

Based on the above setup, the elements in $\mathbb{Z}_q^*$, $\mathbb{G}$, and $\mathbb{G}_T$ are about 153 bytes, 306 bytes, and 612 bytes, respectively. Let the message be 20 bytes, the real identity of ES be 4 bytes, the $Serv$ be 4 bytes, and the timestamp be 4 bytes. In this subsection, we mainly compare the two aspects of data encryption and message signatures.

First, the data that the ES^A publishes is $CT = (Serv, EID_a, Hdr, t_{esa})$, which is about $4 + 4 + 805 + 4 = 817$ bytes. Second, the ES^B sends the re-encrypted data $\{CT_{b \rightarrow i}, t_{esb}\}$ to SD_i and the data is about $1417 + 4 = 1421$ bytes. Finally, the data sent by SD_i to the ES^B is $\{M_{Serv}, APK_i, PID_i, t_i, \sigma_i, EID_a\}$, which is about $20 + 306 + 306 + 4 + 153 + 4 = 793$ bytes. Therefore, the communication overhead associated with data encryption is about $817 + 1421 = 2238$ bytes and the communication overhead associated with message signing is about 793 bytes.

Calculated by the above method, the results are shown in Fig. 7. Since scheme [13] does not provide data signing and authentication algorithms, there is no communication overhead related to data signature. Similarly, schemes [33] and [34] do not provide data encryption and decryption algorithms, so there is no communication overhead related to data encryption.

For data encryption, the communication overhead of our proposed scheme is about 2238 bytes, which is $2238/5282 \approx 42.4\%$ and $2238/3217 \approx 69.6\%$ of [13] and [32], respectively. Although the communication overhead of our proposed scheme is higher than that in scheme [11], the total communication overhead of our proposed scheme is lower.

For data signing, the communication overhead of our proposed scheme is about 793 bytes, which is about $793/2492 \approx 31.8\%$ and $793/938 \approx 84.5\%$ of that in scheme [11] and scheme [34], respectively. From Fig. 7, we find that the communication overhead of our proposed scheme is higher than that of scheme [32] and scheme [33]. However, the total communication overhead of our proposed scheme is lower than scheme [32].

Finally, the total communication overhead is $2238 + 793 = 3031$ bytes in the proposed scheme, which is about $3031/(2492 + 1288) \approx 80.2\%$ and $3031/(3217 + 640) \approx 78.6\%$ of the communication overhead in schemes [11] and [32], respectively.

VIII. Conclusion

This study proposes a blockchain-based, secure, and efficient data-sharing scheme for the IIoT scenario, where smart devices can only access cross-domain data with the assistance of edge servers. First, the scheme supports smart devices in generating pseudonyms independently without requiring online participation by domain authorization centers. This saves storage resources for smart devices and reduces the computational pressure on domain authorization centers. Second, the scheme combines broadcast encryption and proxy re-encryption techniques, which increases the scalability and flexibility of data sharing while ensuring its security. The security proof and analysis present that our scheme meets the security requirements. Performance comparisons show that the scheme achieves low computation and communication overheads. In our next study, we will design a blockchain-based flexible data-sharing scheme for mobile devices in the cross-domain IIoT.

Acknowledgment

The authors are very grateful to the anonymous referees for their detailed comments and suggestions regarding this paper.

References

[1] A. Corallo, M. Lazoi, M. Lezzi, and A. Luperto, "Cybersecurity awareness in the context of the industrial Internet of Things: A systematic literature review," *Comput. Ind.*, vol. 137, May 2022, Art. no. 103614. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361522000094

[2] L. D. 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.

[3] Y. Chi, Y. Dong, Z. J. Wang, F. R. Yu, and V. C. M. Leung, "Knowledge-based fault diagnosis in industrial Internet of Things: A survey," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 12886–12900, Aug. 2022.

[4] Y. Zhang, B. Li, J. Wu, B. Liu, R. Chen, and J. Chang, "Efficient and privacy-preserving blockchain-based multifactor device authentication protocol for cross-domain IIoT," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22501–22515, Nov. 2022.

[5] H. Zhong, C. Gu, Q. Zhang, J. Cui, C. Gu, and D. He, "Conditional privacy-preserving message authentication scheme for cross-domain industrial Internet of Things," *Ad Hoc Netw.*, vol. 144, May 2023, Art. no. 103137. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870523000574

[6] C. Huang et al., "Blockchain-assisted transparent cross-domain authorization and authentication for smart city," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17194–17209, Sep. 2022.

[7] M. Shen et al., "Blockchain-assisted secure device authentication for cross-domain industrial IoT," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 942–954, May 2020.

[8] P. Singh, M. Masud, M. S. Hossain, and A. Kaur, "Cross-domain secure data sharing using blockchain for industrial IoT," *J. Parallel Distrib. Comput.*, vol. 156, pp. 176–184, Oct. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S074373152100112X

[9] K. Miyachi and T. K. Mackey, "HOCBS: A privacy-preserving blockchain framework for healthcare data leveraging an on-chain and off-chain system design," *Inf. Process. Manage.*, vol. 58, no. 3, May 2021, Art. no. 102535. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457321000431

[10] K. Wang, Y. Yan, S. Guo, X. Wei, and S. Shao, "On-chain and off-chain collaborative management system based on consortium blockchain," in *Advances in Artificial Intelligence and Security*, X. Sun, X. Zhang, Z. Xia, and E. Bertino, Eds. Cham, Switzerland: Springer, 2021, pp. 172–187.

[11] K. Xue, X. Zhang, Q. Xia, D. S. L. Wei, H. Yue, and F. Wu, "SEAF: A secure, efficient and accountable access control framework for information centric networking," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun.*, Apr. 2018, pp. 2213–2221.

[12] J. Cui, F. Wang, Q. Zhang, C. Gu, and H. Zhong, "Efficient batch authentication scheme based on edge computing in IIoT," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 1, pp. 357–368, Mar. 2023.

[13] J. Sun, G. Xu, T. Zhang, X. Yang, M. Alazab, and R. H. Deng, "Verifiable, fair and privacy-preserving broadcast authorization for flexible data sharing in clouds," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 683–698, 2023.

[14] P. Xu, J. Li, W. Wang, and H. Jin, "Anonymous identity-based broadcast encryption with constant decryption complexity and strong security," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur.* New York, NY, USA: Association for Computing Machinery, 2016, pp. 223–233, doi: 10.1145/2897845.2897853.

[15] A. Fiat and M. Naor, "Broadcast encryption," in *Advances in Cryptology—CRYPTO' 93*, D. R. Stinson, Ed. Berlin, Germany: Springer, 1994, pp. 480–491.

[16] J. Kim, W. Susilo, M. H. Au, and J. Seberry, "Adaptively secure identity-based broadcast encryption with a constant-sized ciphertext," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 3, pp. 679–693, Mar. 2015.

[17] J. Li, L. Chen, Y. Lu, and Y. Zhang, "Anonymous certificate-based broadcast encryption with constant decryption cost," *Inf. Sci.*, vols. 454–455, pp. 110–127, Jul. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025518303244

[18] J. Hur, C. Park, and S. O. Hwang, "Privacy-preserving identity-based broadcast encryption," *Inf. Fusion*, vol. 13, no. 4, pp. 296–303, Oct. 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253511000145

[19] L. Chen, J. Li, and Y. Zhang, "Anonymous certificate-based broadcast encryption with personalized messages," *IEEE Trans. Broadcast.*, vol. 66, no. 4, pp. 867–881, Dec. 2020.

[20] L. Chen, J. Li, Y. Lu, and Y. Zhang, "Adaptively secure certificate-based broadcast encryption and its application to cloud storage service," *Inf. Sci.*, vol. 538, pp. 273–289, Oct. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S002002552030517X

[21] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Advances in Cryptology—EUROCRYPT'98*, K. Nyberg, Ed. Berlin, Germany: Springer, 1998, pp. 127–144.

[22] B. Chen, D. He, N. Kumar, H. Wang, and K. R. Choo, "A blockchain-based proxy re-encryption with equality test for vehicular communication systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2048–2059, Jul. 2021.

[23] K. O. O. Agyekum, Q. Xia, E. B. Sifah, C. N. A. Cobblah, H. Xia, and J. Gao, "A proxy re-encryption approach to secure data sharing in the Internet of Things based on blockchain," *IEEE Syst. J.*, vol. 16, no. 1, pp. 1685–1696, Mar. 2022.

[24] Y. Zhang and C. L. P. Chen, "Secure heterogeneous data deduplication via fog-assisted mobile crowdsensing in 5G-enabled IIoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 4, pp. 2849–2857, Apr. 2022.

[25] A. Manzoor, A. Braeken, S. S. Kanhere, M. Ylianttila, and M. Liyanage, "Proxy re-encryption enabled secure and anonymous IoT data sharing platform based on blockchain," *J. Netw. Comput. Appl.*, vol. 176, Feb. 2021, Art. no. 102917. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804520303763

[26] H. Lin, J. Shao, C. Zhang, and Y. Fang, "CAM: Cloud-assisted privacy preserving mobile health monitoring," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 6, pp. 985–997, Jun. 2013.

[27] X. Liu, Y. Zhang, B. Wang, and J. Yan, "Mona: Secure multi-owner data sharing for dynamic groups in the cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1182–1191, Jun. 2013.

[28] S. Jiang, J. Liu, L. Wang, Y. Zhou, and Y. Fang, "ESAC: An efficient and secure access control scheme in vehicular named data networking," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10252–10263, Sep. 2020.

[29] Y. Yang, L. Wei, J. Wu, C. Long, and B. Li, "A blockchain-based multidomain authentication scheme for conditional privacy preserving in vehicular ad-hoc network," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8078–8090, Jun. 2022.

[30] C. Lin, X. Huang, and D. He, "EBCPA: Efficient blockchain-based conditional privacy-preserving authentication for VANETs," *IEEE Trans. Depend. Secure Comput.*, vol. 20, no. 3, pp. 1818–1832, Jun. 2023.

[31] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2681–2691, 2015.

[32] H. Zhong, S. Zhang, J. Cui, L. Wei, and L. Liu, "Broadcast encryption scheme for V2I communication in VANETs," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 2749–2760, Mar. 2022.

[33] H. Xiong, Y. Wu, C. Jin, and S. Kumari, "Efficient and privacy-preserving authentication protocol for heterogeneous systems in IIoT," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11713–11724, Dec. 2020.

[34] T. Li, H. Wang, D. He, and J. Yu, "Permissioned blockchain-based anonymous and traceable aggregate signature scheme for industrial Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8387–8398, May 2021.

[35] W. Liang et al., "A secure fabric blockchain-based data transmission technique for industrial Internet-of-Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3582–3592, Jun. 2019.

[36] S. Yu and Y. Park, "A robust authentication protocol for wireless medical sensor networks using blockchain and physically unclonable functions," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20214–20228, Oct. 2022.

[37] Y. Yao, X. Chang, J. Misic, V. B. Misic, and L. Li, "BLA: Blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3775–3784, Apr. 2019.

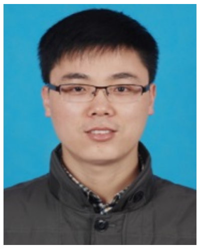
[38] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology—CRYPTO*, J. Kilian, Ed. Berlin, Germany: Springer, 2001, pp. 213–229.

[39] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–396, 2000.

[40] J. Li, Y. Ji, K. R. Choo, and D. Hogrefe, "CL-CPPA: Certificate-less conditional privacy-preserving authentication protocol for the Internet of Vehicles," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10332–10343, Dec. 2019.

[41] *Miracl Cryptographic SDK*. Accessed: Apr. 15, 2023. [Online]. Available: https://github.com/miracl/MIRACL

Fengqun Wang is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Anhui University, Hefei, China. His research interests include the IoT security, blockchain, and applied cryptography.



Jie Cui (Senior Member, IEEE) was born in Henan, China, in 1980. He received the Ph.D. degree from the University of Science and Technology of China in 2012. He is currently a Professor and the Ph.D. Supervisor with the School of Computer Science and Technology, Anhui University. He has more than 150 scientific publications in reputable journals (e.g., IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, IEEE Journal on Selected Areas in Communications, IEEE Transactions on Mobile Computing, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, IEEE Transactions on Vehicular Technology, IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Network and Service Management, IEEE Transactions on Industrial Informatics, IEEE Transactions on Industrial Electronics, IEEE Transactions on Cloud Computing, and IEEE Transactions on Multimedia), academic books, and international conferences. His current research interests include applied cryptography, the IoT security, vehicular ad hoc network, cloud computing security, and software-defined networking (SDN).

Qingyang Zhang (Member, IEEE) was born in Anhui, China, in 1992. He received the B.Eng. and Ph.D. degrees in computer science from Anhui University in 2014 and 2021, respectively. He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University. His research interests include edge computing, computer systems, and security.

Debiao He (Member, IEEE) received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, Wuhan, China, in 2009. He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University. His work has been cited more than 10000 times at Google Scholar. He has published more than 100 research papers in refereed international journals and conferences, such as IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, and Usenix Security Symposium. His main research interests include cryptography and information security, in particular cryptographic protocols. He was a recipient of the 2018 IEEE Systems Journal Best Paper Award and the 2019 IET Information Security Best Paper Award. He is on the Editorial Board of several international journals, such as *Journal of Information Security and Applications*, *Frontiers of Computer Science*, and *Human-Centric Computing and Information Sciences*.

Hong Zhong (Member, IEEE) was born in Anhui, China, in 1965. She received the Ph.D. degree in computer science from the University of Science and Technology of China in 2005. She is currently a Professor and the Ph.D. Supervisor with the School of Computer Science and Technology, Anhui University. She has over 200 scientific publications in reputable journals (e.g., IEEE Journal on Selected Areas in Communications, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Mobile Computing, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Multimedia, IEEE Transactions on Vehicular Technology, IEEE Transactions on Network and Service Management, IEEE Transactions on Cloud Computing, IEEE Transactions on Industrial Informatics, IEEE Transactions on Industrial Electronics, and IEEE Transactions on Big Data), academic books, and international conferences. Her research interests include applied cryptography, the IoT security, vehicular ad hoc network, cloud computing security, and software-defined networking (SDN).