Efficient Revocable Cross-Domain Anonymous Authentication Scheme for IIoT

Mingwei Zeng, Jie Cui, *Senior Member, IEEE*, Qingyang Zhang, *Member, IEEE*, Hong Zhong, *Member, IEEE*, and Debiao He, *Member, IEEE*

***Abstract*—The rapid evolution of the Industrial Internet of Things (IIoT) has necessitated increased device interactions across various management domains. This entails devices from different domains collaborating on the same production task. This poses significant challenges for the dynamics of cross-domain authentication schemes. Traditional cross-domain authentication schemes struggle to support seamless switching between domains and face difficulties when accommodating devices that join and leave the same domain. Moreover, these schemes suffer from intricate interactions and suboptimal efficiency. To address these issues, we propose a dynamic group signature scheme based on a dynamic accumulator and a non-interactive zero-knowledge proof. We integrated this scheme with blockchain technology to construct an efficient revocation cross-domain authentication scheme. The proposed scheme enables cross-domain anonymous authentication with simple interactions and provides an efficient revocation function for illegal devices. This approach ensures conditional privacy-preserving and enables efficient member joining and exiting through a dynamic accumulator. It effectively addresses the dynamic requirements of devices involved in IIoT production and manufacturing processes. We prove the security of the proposed scheme using a random Oracle model and conduct thorough analyses to verify its resistance against various attacks. Furthermore, the experimental results demonstrate that the proposed scheme achieves better performance in terms of computational and communication costs.**

***Index Terms*—Cross-domain authentication, industrial Internet of Things (IIoT), dynamic accumulator.**

I. Introduction

THE Industrial Internet of Things (IIoT) [1] applies Internet of Things [2] technology to the industrial domain for production and practical purposes. It connects physical objects such as sensors and actuators to the internet, enabling real-time data collection, monitoring, analysis, and control functions. The optimization of production processes enhances efficiency, reduces costs, and improves product quality [3]. With the continuous development of the IIoT, industrial manufacturing is becoming increasingly complex and often requires cross-domain collaboration to process and manufacture a product [4]. Traditional cross-domain collaboration scenarios in the IIoT are illustrated in Fig. 1. The trusted authority is responsible for the issuance and maintenance of certificates for IIoT devices within the domain to facilitate identity authentication and privacy protection [5]. Meanwhile, IIoT devices from different domains collaborate with edge servers to accomplish industrial production tasks [6], with edge servers processing and analyzing data collected from IIoT devices at locations closer to the data source, thereby enabling real-time monitoring, rapid response, and more efficient data processing [7].

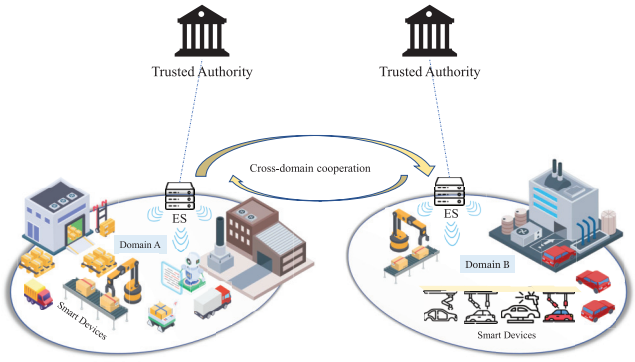


Fig. 1. Cross-domain collaboration in industrial Internet of Things.

Unauthorized access to IIoT systems may result in the leakage of production data and sensitive information, thereby affecting the accuracy and reliability of production processes. Identity authentication mechanisms validate the legitimacy of user identities, ensuring that only authorized devices can access the system. This prevents the leakage of private information and unauthorized tampering and ensures the continuity and stability of the production process. However, current IIoT systems still face challenges with cross-domain authentication.

First, IIoT devices from different domains are typically provided by various manufacturers and suffer from a mutual lack of trust. However, they must collaborate to complete essential production tasks. If IIoT devices authenticate and communicate using their real identities, malicious adversaries

Received 20 May 2024; revised 10 October 2024 and 27 November 2024; accepted 9 December 2024. Date of publication 25 December 2024; date of current version 8 January 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62272002, Grant 62202005, Grant 62372002, and Grant U24A20243; in part by the Natural Science Foundation of Anhui Province, China, under Grant 2208085QF198; 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 article and approving it for publication was Dr. Andrew Clark. *(Corresponding author: Jie Cui.)*

Mingwei Zeng, Jie Cui, Qingyang Zhang, and Hong Zhong are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, and Anhui Engineering Laboratory of IoT Security Technologies, 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, and also with Shanghai Key Laboratory of Privacy Preserving Computation, MatrixElements Technologies, Shanghai 201204, China (e-mail: hedebiao@163.com).

Digital Object Identifier 10.1109/TIFS.2024.3523198

can collect and analyze the data transmitted by these devices, potentially resulting in privacy breaches [8] in devices or even IIoT. For privacy, these devices aim to authenticate their identities without revealing their real identities during cross-domain authentication. Therefore, it is necessary to provide anonymous cross-domain authentication. In addition, compromised IIoT devices may send malicious data to disrupt normal production processes. Hence, it is essential to have methods that acquire the real identities of IIoT devices while ensuring conditional privacy-preserving.

In addition, to coordinate production relationships across different domains, some IIoT devices may need to move between domains to better accomplish cross-domain production tasks [9]. In the production process, there may be a need to introduce advanced equipment or replace damaged equipment. Therefore, to adapt to this dynamism, a cross-domain authentication scheme should provide functionality for both device addition and revocation. Certain scenarios demand a high feedback time from IIoT devices. For instance, in real-time monitoring systems within smart factories, promptly collecting and processing various sensor data is essential for monitoring the equipment status, production processes, and environmental conditions. Any delay can lead to production failure or safety risks. Therefore, cross-domain authentication schemes should consider the efficiency issues related to device addition and revocation.

Therefore, we propose a cross-domain authentication scheme that achieves conditional privacy-preserving, supports dynamic joining and facilitates efficient revocation of IIoT devices. Numerous studies have explored cross-domain authentications. We will proceed to introduce and summarize the relevant works on cross-domain authentication in the IIoT.

A. Related Work

Wang et al. [10] abstracted the authentication relationships among smart devices as an undirected graph and transformed the authentication issue into one of signature transitivity by combining digital signature and dynamic accumulator technology. This enables cross-domain identity authentication by computing the signatures of relevant edge devices. Chen et al. [11] proposed an attribute-based signature (ABS) using an attribute tree for access policies, improving private key generation and signature verification efficiency. They proved the scheme's unforgeability and anonymity in the standard model and reduced the computational burden on IoT devices. Millán et al. [12] proposed a solution where a third-party Certificate Authority (CA) issues certificates for different domains to achieve cross-domain authentication, resolving issues related to key pre-distribution and management. However, this approach creates a reliance on the third-party CA, rendering it vulnerable to potential attacks [13]. Li et al. [14] introduced a decentralized attribute-based server-assisted signature (DABSAS) scheme for anonymous IoT authentication. This approach decreases IoT device overhead with server support and proves security under the co-CDH assumption. Hao et al. [9] used IBC-based authentication methods to eliminate reliance on third-party TA and solved complex certificate management issues in traditional PKI systems. However, this study does not consider the issue of revoking illegal devices. Jia et al. [15] and Zhong et al. [16] also use similar methods to arrive at cross-domain authentication. Xue et al. [17] utilized a combination of multi-blockchains to achieve authentication within and across domains, employing chameleon hashes to minimize storage overhead. However, its system model and authentication process are complex, and the aforementioned schemes do not implement conditional privacy-preserving.

Xiong et al. [18] proposed an authentication scheme with conditional privacy-preserving using the Chinese Remainder Theorem (CRT). The domain manager broadcasts domain keys to authorized users within the domain using CRT and can update domain keys through lightweight operations. Zhang et al. [19] introduced an anonymous authentication scheme based on Merkle hash trees and group signature. After the initial authentication, they expedited subsequent authentication processes by storing partial information on the server, enabling one-time authentication for multiple accesses. However, these methods do not address the issue of revoking group members. After considering both administrative domains and geographical domains, Cheng et al. [20] proposed an identity-based signature scheme that does not require bilinear pairings. This scheme achieves conditional privacy-preserving and enhances system security through the design of a dynamically sparse Merkle tree structure. Wang et al. [21] introduced edge device-assisted authentication and proposed a blockchain-based identity authentication scheme to accommodate IIoT environments with multiple recipients, significantly reducing the computational overhead of IIoT devices. However, this scheme requires extensive interaction between entities for key negotiation before cross-domain authentication from domain A to domain Chen et al. [22] proposed a lightweight correctness verification scheme based on a multi-Merkle hash tree to address the issue of slow authentication response due to the low throughput of the blockchain. They also designed an anonymous cross-domain authentication protocol based on public key infrastructure to protect user privacy. However, these schemes lack efficient revocation mechanisms for illegal users and overlook the dynamic nature of IIoT devices.

Kang et al. [23] developed a traceable and forward-secure attribute-based signature (TFS-ABS) scheme with constant-size flexible threshold predicates. This scheme addresses signature misuse and key exposure issues in existing ABS schemes, providing a new solution for anonymous authentication. Shen et al. [24] proposed an efficient blockchain-assisted identity authentication scheme that combines blockchain with identity-based cryptography to achieve identity authentication and key negotiation. This mechanism ensures device anonymity through pseudonyms, and ensures security by periodically updating pseudonyms. However, frequent pseudonym changes can lead to exponentially increasing maintenance costs as IoT device numbers grow. Similarly, many other schemes also utilize pseudonyms for anonymous authentication [16], [25], [26], [27]. Tong et al. [8] introduced a completely cross-domain authentication mechanism leveraging blockchain technology, CCAP, which realizes conditional privacy-preserving, facilitating cross-domain authentication among IoT devices between various

cryptosystem trust domains. It achieved conditional privacy-preserving through zero-knowledge proof and Shamir secret sharing. Qikun et al. [28] introduced a dynamic cross-domain authentication scheme based on bilinear mapping, where group keys are dynamically negotiated by CAs from various domains during authentication. However, it requires pre-determination of devices engaging in cross-domain communication, and the time complexity of key negotiation is O(n). Wang et al. [10] introduced accumulator to facilitate dynamic membership changes, allowing only fixed-size accumulator values to be stored on the blockchain without storing user certificate information, thus reducing blockchain storage pressure. However, the computational complexity of accumulator values and corresponding evidence is proportional to the number of domain members, making it unsuitable for IIoT scenarios with numerous IIoT devices. The aforementioned schemes fall short of effectively addressing the low-latency demands within the IIoT.

The aforementioned cross-domain authentication schemes have not comprehensively considered the dynamic nature of IoT devices, conditional privacy-preserving for device identities across different domains, and the low-latency requirements in certain scenarios.

B. Our Contribution

We propose a group signature based on a dynamic accumulator and non-interactive zero-knowledge proof that achieves conditional privacy-preserving and supports rapid membership addition and efficient revocation. We proposed a cross-domain authentication scheme by integrating this signature into a blockchain. This integration simplifies the authentication process and eliminates dependency on third-party trusted authorities (TAs). The main contributions of this study are as follows:

- In response to the dynamic properties exhibited by devices in IIoT environments, we propose an efficient revocation dynamic group signature scheme. Compared with traditional group signature schemes, the proposed scheme provides efficient revocation functionality, facilitating swift group member registration and efficient group member revocation.
- We introduce a blockchain-assisted cross-domain authentication scheme that considers the privacy-preserving issue in the IIoT. This approach enables anonymous authentication without using pseudonyms, thereby avoiding the various costs associated with pseudonym maintenance. In addition, specific entities can increase the anonymity of IIoT devices, ensuring conditional privacy-preserving, which is more suitable for IIoT environments.
- We establish the provable security of the proposed scheme and conduct a comprehensive security analysis. Our investigation confirms that our approach meets essential security properties, including anonymity, traceability, and unlinkability. Simulation experiment results indicate that our scheme exhibits lower computational and communication overheads compared to contrast schemes.

C. Paper Outline

Our study is structured as follows: Section II introduces the fundamental cryptographic knowledge utilized in this study. Section III presents the system model, while Section IV provides a detailed explanation of the proposed group signature. In Section V, we elaborate on the details of our cross-domain authentication scheme, followed by a security proof and analyses of the proposed scheme in Section VI. Lastly, Section VII contrasts our scheme with others through experimental comparisons.

II. Preliminaries

In our proposed cross-domain authentication scheme, we'll utilize existing concepts like bilinear mapping and dynamic accumulator, defined as follows:

A. Bilinear Mapping

Let G_1 be an additive cyclic group defined over a prime-order finite field $\mathbb{F}_q$, while G_2 and G_T are respectively additive and multiplicative cyclic groups defined over different extension fields of $\mathbb{F}_q$. Suppose e is a mapping from G_1, G_2 to G_T, denoted as $e : G_1 \times G_2 \rightarrow G_T$. It is called a bilinear mapping if e satisfies the following properties.

Bilinearity: For any $k, h \in \mathbb{Z}_q$ and $P_1 \in G_1$, $P_2 \in G_2$, if $e(kP_1, hP_2) = e(khP_1, P_2) = e(P_1, khP_2) = e(P_1, P_2)^{kh}$ holds, then we say that the mapping e satisfies bilinearity.

Non-degeneracy: There exists $R \in G_1$ and $S \in G_2$ satisfy $e(R, S) \neq 1_{G_T}$ (1_{G_T} represents the identity of G_T).

Computability: For any $R \in G_1$ and $S \in G_2$, the computation of $e(R, S)$ is efficient.

B. Dynamic Accumulator

We introduce the dynamic accumulator proposed by Nguyen et al. [29], which has been proven to possess collision resistance. Suppose G_1 is a cyclic group of order p generated by P_1. Two functions f and g are defined as follows: $f_x(u, s) = u(s + x)$, $g(u) = uP_1$(where x is a secret value).

Dynamic accumulator is defined as $\mathcal{DA} = \{Z_p^*, G_1, f_x, g\}$. For a set $S = \{s_1, \ldots, s_n\}$, its accumulator value is defined as $ACC_S = g(f_x(u, S)) = (u \prod_{i=0}^{n}(s_i + x))P_1$. For an element $s_i \in S$, its proof of membership in the set S corresponding to ACC_S is $W_i = (s_i + x)^{-1} ACC_S$. In addition, a dynamic accumulator $\mathcal{DA}$ satisfies the following properties:

Quasi Commutativity: $\mathcal{DA}$ satisfies quasi commutativity if, for any $s_1, \ldots, s_n \in \mathbb{Z}_p^*$, the equation $f_x(f_x(f_x(u, s_1), s_2), \ldots, s_n) = f_x(f_x(f_x(u, s_{i_1}), s_{i_2}), \ldots, s_{i_n}) = u \prod_{i=0}^{n}(s_i + x)$ holds, where $(i_1, \ldots, i_n)$ is an arbitrary permutation of $(1, 2, \ldots, n)$.

Efficient Addition: If we add s' to the set S. The value of the accumulator ACC' and the corresponding proof can be calculated as follows: $ACC' = (s' + x)ACC_S$, $W' = ACC_S$.

Efficient Deletion: If we delete s' from the set S. The value of the accumulator ACC' and the corresponding proof can be calculated as follows: $ACC' = (s' + x)^{-1} ACC_S$, $W' = \emptyset$.

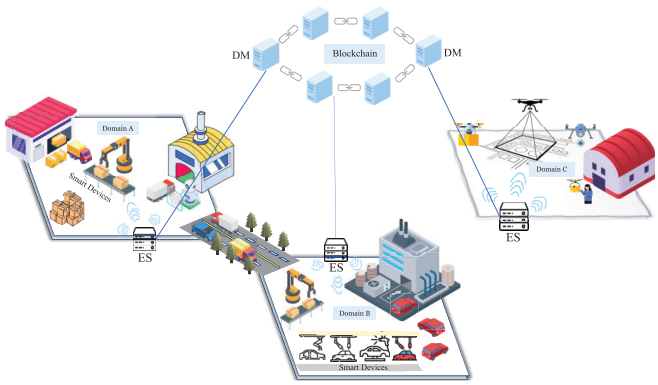


Fig. 2. Cross-domain authentication system model for IIoT.

III. System Architecture

A. System Model

Complex Industrial Internet of Things (IIoT) production tasks often require collaboration among multiple factories across different domains. IIoT devices in different domains exchange production information over the network, which enables better coordination of production tasks and improves overall efficiency. To establish trust between IIoT devices across different domains and ensure the security of production data, IIoT devices perform identity authentication with the assistance of edge servers before communication. The system model is shown in Fig. 2.

- **DM:** Each domain has a trusted domain manager (DM) responsible for generating the system parameters needed for authentication, as well as issuing certificates and domain member proofs to IIoT devices that join the domain. They are also responsible for managing the IIoT devices within the domain, including identity tracking and revoking access for malicious users.
- **BC:** The blockchain is composed of DMs from different domains who maintain the blockchain using smart contracts [30], [31]. This ensures that the blockchain always stores the latest authentication parameters. IIoT devices can query the blockchain to update their domain member proofs or to retrieve the necessary parameters for authentication.
- **SD:** Smart Devices are resource-constrained IIoT devices responsible for industrial production. When communicating with smart devices from different domains, they generate the necessary authentication messages and, upon successful authentication, engage in encrypted communication with devices from other domains.
- **ES:** Edge Servers deployed in smart factories are equipped with strong computational and storage capabilities. They offer processing power and storage resources to nearby IIoT devices, enabling more efficient completion of identity authentication and production tasks.

B. Threat Model

In the proposed scheme, edge servers are treated as semi-honest entities. This implies that while they will faithfully execute the protocol, they might try to deduce the privacy information of other participants by analyzing the data generated during the execution of the protocol. In contrast, IIoT devices are regarded as untrusted entities. Together with external attackers, they may seek to achieve unauthorized authentication through various tactics. For instance, they might impersonate legitimate devices to gain successful authentication or create information that can pass authentication without being traced back to the attacker's real identity. These adversarial capabilities align with the Dolev and Yao [32] threat model, which posits that attackers can monitor public channels, modify transmitted messages, impersonate legitimate entities, and generate arbitrary messages to interact with legitimate users. However, attackers are unable to guess the random numbers selected during the protocol, solve the assumed hard problems, or access the private information stored in trusted entities.

C. Security Objectives

Cross-domain authentication for IIoT devices should meet the following security objectives to ensure the safety of the production process:

- **Anonymity:** The proposed solution should guarantee that the identity information of IIoT devices remains confidential, thus safeguarding their identity privacy and ensuring anonymity throughout the authentication process.
- **Full-traceability:** Unregistered devices should not be able to forge messages that successfully pass authentication, which requires the protocol to ensure non-forgeability. Additionally, the system must be able to trace illegal IIoT devices, necessitating that the authentication scheme supports traceability. Legitimate IIoT devices should not be able to collude to create a signature that cannot be traced back to any of their identities, thus preventing coalition attacks. Bellare et al. [33] proposed a stronger security goal known as "full-traceability". They demonstrated that this concept not only includes non-forgeability and traceability but also offers resistance against coalition attacks.
- **Conditional Privacy-preserving:** Absolute anonymity could lead to compromised IIoT devices launching malicious attacks on the system, with no way to trace the attacker's identity. Therefore, there should be a method to reveal the real identity of devices to enable conditional privacy-preserving.
- **Revocability:** To prevent compromised industrial IoT devices from sending false production information or executing malicious production operations that disrupt normal industrial processes, an identity revocation function for malicious users should be provided to promptly revoke the permissions of these devices.
- **Unlinkability:** The authentication scheme should ensure that adversaries cannot determine whether two messages originate from the same entity.
- **Non-repudiation:** IIoT devices must be unable to deny having transmitted a message, ensuring that the origin of the message can be verified and attributed to the device that sent it.

TABLE I
GROUP SIGNATURE SYMBOLS

Notations	Description
q	A large prime number
G_1	A additive cyclic group defined on $\mathbb{F}_q$
G_2	A additive cyclic group
G_T	A multiplicative Cyclic Group
P_1	The generator of G_1
P_2	The generator of G_2
P_{pub}	Group public key
$H(\cdot)$	A secure hash function
ACC_j	The value of the dynamic accumulator
C_i	Group member certificate
W_i	Proof of group members
D_A, D_B	Domain A (B)
DM_A, DM_B	Domain manager in domain A (B)
SD_i^A, SD_i^B	A certain IIoT device in domain A (B)

IV. PROPOSED GROUP SIGNATURE SCHEME

Chaum et al. first proposed the concept of group signature in 1991. It satisfies the following requirements: A member of the group can sign messages on behalf of the group while remaining anonymous, and anyone can verify the correctness of the signature using the group's public key. Additionally, there exists a group manager who can use the group private key to de-anonymize group members and obtain their real identities. Over the years, various group signature schemes have been proposed, but efficiently revoking group members' group signatures remains an unresolved issue. We propose a group signature scheme based on dynamic accumulator and bilinear mapping. Compared with traditional group signatures, the proposed group signature achieves conditional privacy-preserving and enables efficient group numbers revocation.

The group signature algorithm we propose consists of seven algorithms: *Setup*, *Join*, *Sign*, *Verify*, *Open*, *Revoke*, and *Update*. Table I displays the symbols used in our scheme.

***Setup*$(\mathbf{1}^{\lambda}) \rightarrow$ (*params*, *gsk*)**: To generate a group signature, the group manager (GM) first creates a group. He performs the setup algorithm by providing a security parameter λ, which generates the group's public parameters and group private key, thereby creating the group.

Suppose G_1 is an additive cyclic group defined in the finite field $\mathbb{F}_q$, where q is a large prime number. e is a bilinear mapping: $e : G_1 \times G_2 \rightarrow G_T$, where P_1 and P_2 are generators of G_1 and G_2, respectively. The specific process is as follows:

1) Randomly generate two numbers $x, r \in_R \mathbb{Z}_q^*$ and calculate $P_{pub} = xP_1$ and $ACC_0 = rP_2$.
2) Choose a collision-resistant hash function $H(\cdot)$.
3) Create an list L. The elements stored in L have the following format: $< id_i, H(id_i)P_1, join/delete, ACC_j >$.
4) The GM issues the public group parameters *params* : $\{q, G_1, G_2, G_T, e, P_1, P_2, P_{pub}, ACC_0, H(\cdot)\}$ and he saves x as the group private key *gsk*.

***Join*$(id_i, gsk, params) \rightarrow (C_i, W_i)$**: To generate a group signature, User U_i needs to first register with the group manager. The group manager will issue a certificate and membership proof to validate the user as a legitimate group member. The specific process is as follows:

1) User U_i sends his real identity id_i to the group manager.
2) The GM computes $s_i = H(id_i)$ and further calculates the group certificate for user U_i as $C_i = (s_i + x)^{-1}P_2$. Then, the GM adds s_i to the accumulator, updates its value as $ACC_j = (s_i + x)ACC_{j-1}$, and provides the group membership proof $W_i = ACC_{j-1}$ for user U_i. Finally, the GM stores $< id_i, H(id_i)P_1, join, ACC_j >$ as an entry in L and sends $\{C_i, W_i\}$ to user U_i.
3) Upon receiving the message, U_i computes $s_i = H(id_i)$, and $gsk[i] = (s_i, C_i, W_i)$ is assigned as the private key of the group member.

***Sign*$(s_i, C_i, W_i, params, M) \rightarrow (\sigma)$**: After successfully executing the *Join* algorithm, user U_i becomes a legitimate group member. Subsequently, he can use their certificate and group membership proof to generate a group signature σ. The specific process is as follows:

1) Generate a random number $u \in_R Z_q^*$ and calculate: $T_1 = uP_1$; $T_2 = uT_1$; $T_3 = s_iP_1 + uP_{pub}$ and $A_1 = uW_i$; $A_2 = uC_i$
2) Generate two random numbers $r_u \in_R Z_q^*, r_s \in_R Z_q^*$ and calculate: $R_1 = r_uP_1$; $R_2 = r_uT_1$; $R_3 = e(P_{pub}, A_1 + A_2)^{r_u} e(T_1, A_1 + A_2)^{r_s}$; $R_4 = r_sP_1 + r_uP_{pub}$
3) Calculate: $c = H(M, T_1, T_2, T_3, A_1, A_2, R_1, R_2, R_3, R_4)$ and $s_u = r_u + cu$; $s_s = r_u + cs_i$
4) Then the user U_i signs the message M as follows:$\sigma = \{M, T_1, T_2, T_3, A_1, A_2, c, s_u, s_s\}$

***Verify*$(M, \sigma, params) \rightarrow (1/\perp)$**: To verify the legitimacy of the signature σ, any verifier can first compute $\hat{R}_1$, $\hat{R}_2$, $\hat{R}_3$, and $\hat{R}_4$ as follows:

$$\hat{R}_1 = s_uP_1 - cT_1; \hat{R}_2 = s_uT_1 - cT_2 \tag{1}$$

$$\hat{R}_3 = e(P_{pub}, A_1 + A_2)^{s_u} e(T_1, A_1 + A_2)^{s_s} e(T_2, ACC_j + P_2)^{-c} \tag{2}$$

$$\hat{R}_4 = s_sP_1 + s_uP_{pub} - cT_3 \tag{3}$$

He checks if $c \stackrel{?}{=} H(M, T_1, T_2, T_3, A_1, A_2, \hat{R}_1, \hat{R}_2, \hat{R}_3, \hat{R}_4)$ holds. If the above checks pass, the verification is successful, and return 1; otherwise, the verification fails, and return $\perp$.

***Open*$(T_1, T_3, gsk, L) \rightarrow (id_*)$**: In this stage, the group manager can use his private key *gsk* to reveal the real identity of a group member. He first verify whether $Verify(M, \sigma, params) = 1$ holds. If the equation does not hold, He will refuse to open the signature. Otherwise, he calculate $s_*P_1 = T_3 - xT_1$ and query L to obtain $< id_*, s_*P_1, join, ACC_k >$ based on s_*P_1, thus obtain id_*.

***Revoke*$(id_*) \rightarrow (ACC_{j+1}, L)$**: The group manager runs the *Open* to obtain the real identity id^* of the message sender. Subsequently, using this identity, the GM efficiently revokes

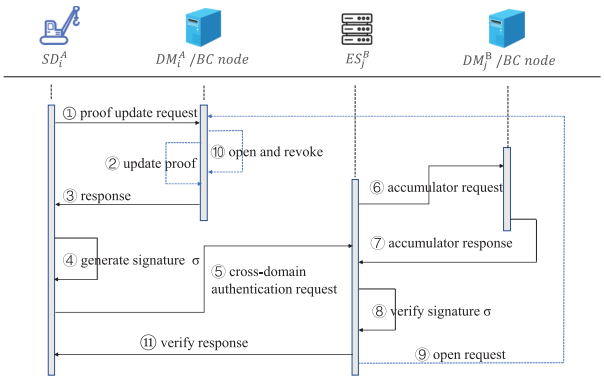


Fig. 3. IIoT cross-domain authentication process.

the user from the group by modifying the value of the accumulator. The specific process is as follows:

First, the GM runs $Open(T_1, T_3, gsk, L) \rightarrow (id_*)$ to obtain id_*. Then, the GM calculates $ACC_{j+1} = (s_* + x)^{-1} ACC_j$ and stores $< id_*, s_* P_2, delete, ACC_{j+1} >$ in the list L. Finally, replace ACC_j in *params* with the latest ACC_{j+1}.

***Update*(L) → (W_i')**: Due to the dynamic nature of group members, the value of the accumulator keeps changing. Therefore, when starting a new round of authentication, the user needs to update its group members evidence. The specific operation is as follows:

1) Suppose U_i last updated the evidence with a list $L = \{l_1, \ldots, l_k\}$, and the current list L contains a total of n items $\{l_1, \ldots, l_k, l_{k+1}, \ldots, l_n\}$. Let $L_{sub} = \{l_{k+1}, \ldots, l_n\}$. The GM traverses L_{sub} from the end backwards. For each entry $l_k =< id_k, s_k, s_k P_1, flag_k, ACC_j >$, if $id_k \neq id_i$, two sets S_{join} and S_{delete} are created according to the following rules:
   - If $flag_k = join$ in l_k, then add s_k to the set S_0.
   - If $flag_k = delete$, then add s_k to the set S_1.
2) if $id_k = id_i$ or L_{sub} has been fully traversed, *GM* computes w and updates U_i's member proof as $W_i' = wW_i$.

$$w = \prod_{i \in S_0} (s_i + x)^{-1} \prod_{j \in S_1} (s_j + x)$$

If a group member id_* is revoked, then there exists an entry $l_* =< id_*, H(id_*)P_1, \text{delete}, ACC_k >$ in L_{sub}. Consequently, when GM traverses L_{sub} up to l_*, the traversal will terminate, and at this point, the updated W' is evidently invalid.

V. Cross-Domain Authentication Scheme

The cross-domain authentication scheme we propose is based on the construction of group signature described above. Each domain's domain manager creates a group and acts as the group manager to issue certificates and domain membership proofs to IIoT devices that wish to join the domain. At the same time, the domain manager, as a node in the blockchain, updates the latest accumulator value ACC_j and public parameters of the domain on the blockchain for authentication.

Fig. 3 illustrates the process of cross-domain authentication between the device SD_i^A in domain D_A and the IIoT device in domain D_B, which consists of three main stages: system initialization, cross-domain authentication, and tracking and revocation of illegal devices. Below, we will sequentially introduce the content of these three parts.

A. System Initialization

Without loss of generality, we illustrate the system initialization process in domain A. The domain manager DM_A first generates the necessary parameters for group signatures and publishes the group public key GPK^A on the blockchain. The specific process is as follows:

The DM_A selects two additive cyclic groups G_1^A, G_2^A and a multiplicative cyclic group G_T^A over a finite field $\mathbb{F}_q$, and chooses a bilinear map $e : G_1^A \times G_2^A \rightarrow G_T^A$. Then, two random numbers x and r are generated, and $P_{pub}^A = xP_1^A$ and $ACC_0^A = rP_2^A$ are computed. Finally, the DM_A selects a secure hash function and updates the domain's group public key $GPK^A = \{q^A, G_1^A, G_2^A, G_T^A, e, P_1^A, P_2^A, P_{pub}^A, ACC_0^A, H(\cdot)\}$ to the blockchain via a smart contract. The DM_A creates a list L that can be used to track the identities of malicious users and update the latest evidence of group members. The elements stored in L have the following format: $< id_i, H(id_i)P_1, join/delete, ACC_j >$. The GM_A stores this list and the group private key in his memory.

Suppose a set of IIoT devices required for a production task is denoted as $SD_i^A (i = 1, 2, \ldots, n)$, each of these IIoT devices needs to join the group created by the domain manager. Firstly, SD_i^A generates its own identity id_i^A and sends it as a registration request to the domain manager DM_A.

The DM_A computes $s_i = H(id_i^A)$, $C_i = (s_i + x)^{-1} P_2^A$ and $ACC_j = (s_i + x) ACC_{j-1}$, $W_i = ACC_{j-1}$. And sends (s_i, W_i, C_i) to SD_i^A, and the latest accumulator value ACC_j is updated to the blockchain. Meanwhile, a group membership change notification is published on the blockchain.

Upon receiving the message from the group manager, IIoT device SD_i^A stores $gsk[i] = (s_i, W_i, C_i)$ as its domain member private key for subsequent signing operations.

B. Cross-Domain Authentication

Suppose IIoT devices from two different domains wish to engage in cross-domain collaboration. They initiate cross-domain authentication as follows. Without loss of generality, let's assume that a smart device SD_i^A from domain D_A intends to perform cross-domain authentication with a device SD_j^B from domain D_B. Devices in domain A query the blockchain during their idle time. If they observe a membership change notification for their domain on the blockchain, the device will send a group membership evidence update request to the domain manager. Upon receiving the request, DM_A will partition the relevant records in the list L into two sets S_0, S_1 using the *Update* algorithm. Then, domain manager DM_A compute $w = \prod_{i \in S_0} (s_i + x)^{-1} \prod_{j \in S_1} (s_j + x)$, and the updated evidence $W_i' = wW_i$ is sent to the IIoT device SD_i^A.

The IIoT device SD_i^A randomly selects a message M and generates a random number u. It then calculates $T_1 = uP_1^A$, $T_2 = uT_1$, $T_3 = s_i P_1^A + uP_{pub}^A$ and $A_1 = uW_i$, $A_2 = uC_i$. Then two random numbers $r_u \in_R \mathbb{Z}_q^*$ and $r_s \in_R \mathbb{Z}_q^*$ are generated by SD_i^A and compute:

1) $R_1 = r_u P_1^A$; $R_2 = r_u T_1$
2) $R_3 = e(P_{pub}^A, A_1 + A_2)^{r_u} e(T_1, A_1 + A_2)^{r_s}$

3) $R_4 = r_s P_1^A + r_u P_{pub}$
4) $c = H(M, T_1, T_2, T_3, A_1, A_2, R_1, R_2, R_3, R_4)$
5) $s_u = r_u + cu;\ s_s = r_u + cs_i$

The Industrial Internet of Things device sends the signature $\sigma = \{M, T_1, T_2, T_3, A_1, A_2, c, s_u, s_s\}$ of message M to the edge device ES_B in domain B.

The edge device ES_B in domain B first requests the blockchain to obtain the latest accumulator value from domain A. Then, it calculates:

1) $\hat{R_1} = s_u P_1^A - cT_1;\ \hat{R_2} = s_u T_1 - cT_2$
2) $\hat{R_3} = e(P_{pub}^A, A_1 + A_2)^{s_u} e(T_1, A_1 + A_2)^{s_s} e(T_2, ACC_j + P_2^A)^{-c}$
3) $\hat{R_4} = s_s P_1^A + s_u P_{pub}^A - cT_3$
4) $c' = H(M, T_1, T_2, T_3, A_1, A_2, \hat{R_1}, \hat{R_2}, \hat{R_3}, \hat{R_4})$

Finally, the edge device ES_B verifies whether the equation $c' \stackrel{?}{=} H(M, T_1, T_2, T_3, A_1, A_2, \hat{R_1}, \hat{R_2}, \hat{R_3}, \hat{R_4})$ holds. If it does, the authentication of device A is successful. The edge device responds with an authentication success message to the device SD_i^A in domain A. Subsequently, devices in domain A can engage in cross-domain communication.

C. Tracking and Revoking of Illegal Devices

If during communication between IIoT devices from different domains, an IIoT device in domain B detects anomalies in a device from domain A (for example, device SD_i^A fails to send messages as required or sends malicious messages), the edge device can send an identity disclosure request σ to the blockchain. Upon receiving this request, the domain manager in the corresponding domain will process it. First, it verifies whether the device is indeed violating the rules. If the device is indeed malicious, the domain manager executes the following algorithm to determine the real identity of the signer.

First, the domain manager obtains a signature $\sigma = \{M, T_1, T_2, T_3, A_1, A_2, c, s_u, s_s\}$. Then, it checks whether $Verify(M, \sigma, params) \stackrel{?}{=} 1$ holds. If it does, the domain manager uses its private key to compute $s_* P_1^A = T_3 - xT_1$. Then, it searches table L to find the entry $< id_*, s_* P_1^A, join, ACC_k >$ based on $s_* P_1^A$, thus obtaining the true identity id_* of the signer.

If the IIoT device behavior is particularly malicious, the domain manager can compute $ACC_{j+1} = (s_* + x)^{-1} ACC_j$ to revoke the unauthorized user. Then, the domain manager stores $< id_*, s_* P_1^A, delete, ACC_{j+1} >$ in L and publishes a notification of group membership changes to the blockchain.

VI. Security Proof and Analyses

We first demonstrate the correctness, completeness, and honest-verifier zero-knowledge properties of the proposed group signature. Based on these properties, we will subsequently prove the scheme's security. Bellare et al. [33] defined three essential security properties for group signatures: correctness, full-anonymity, and full-traceability. Boneh et al. [34] introduced CPA-full-anonymity as a weakened form of full-anonymity. We will show that the proposed group signature scheme satisfies both CPA-full-anonymity and full-traceability.

A. Security Model

Informally, anonymity requires that an adversary cannot deduce the identity of the signer from the signature. In contrast to standard anonymity, full-anonymity grants the adversary stronger capabilities: they can collude with group members (gaining access to their private keys) and determine the signer of an existing signature (gaining access to the Open Oracle). Our group signature adopts CPA-full-anonymity as defined by Boneh et al. [34], where the adversary does not have access to the Open Oracle and, therefore, cannot obtain the group private key. In practice, as the group private key is generally well-protected, this assumption is reasonable within standard security models [34].

Experiment $\mathrm{Exp}_{\mathcal{GS},\mathcal{A}}^{\text{Anon-b}}(\lambda)$

$params \leftarrow \textbf{\textit{Setup}}(1^\lambda)$; $Let\ [n] = \{1, \ldots, n\}$
$gsk[k] \leftarrow \textbf{\textit{Join}}(id_k, gmsk, params); (k \in [n])$
Let $U \subset Z_n,\ k \in U$
While $i < q_s$ **do**
  $\sigma_i \leftarrow \textbf{\textit{Sign}}(gsk[k], params, m)$
End while
$\mathcal{A}(1^\lambda, gsk[k], \{\sigma_i\}) \rightarrow (St, m, ID_0, ID_1)$
$b \in \{0, 1\}\ \ \sigma_b \leftarrow \textbf{\textit{Sign}}(gsk[b], params, m)$
$b' \leftarrow \mathcal{A}(m, \sigma_b, St)$
If $b' = b$ **then return** 1
else return 0
End if

To formally define CPA-full-anonymity, we define the following experiment:

The process begins with the setup algorithm, which generates the public parameters for the group signature scheme. Following this, private keys are generated for each group member, along with the group's public and private keys.

The adversary is allowed to collude with any number of group members, gaining access to their private keys generated in the previous step. Using these keys, the adversary selects a message m for which it wants a signature and can produce a polynomial number of valid signatures on m. With this information, the adversary selects two identities, ID_0 and ID_1, to maximize its chances of success in the next step. It then submits the tuple $(m, \mathrm{ID}_0, \mathrm{ID}_1)$ to a challenger C, retaining all gathered information in its internal state St.

The challenger randomly selects $b \in \{0, 1\}$ and signs the message m using the private key gsk[b] associated with IDb, producing a signature σ_b, which is sent back to the adversary. Using its internal state St and the signature σ_b, the adversary makes a guess b'. If $b' = b$, the adversary "wins" the game, and the experiment outputs 1; otherwise, it outputs 0.

The adversary's advantage in breaking CPA-full-anonymity is defined as follows:

$$Adv(\lambda) = |\ Pr[Exp_{\mathcal{GS},\mathcal{A}}^{\text{Anon-b}}(\lambda) = 1] - 1/2\ |$$

If the adversary's advantage $Adv(\lambda)$ in winning this experiment is negligible, then we say that the proposed group signature scheme satisfies CPA-full anonymity.

The security definition for full-traceability in the proposed scheme is identical to that proposed by Bellare et al. [33], and thus will not be repeated here.

B. Security Assumption

In order to prove the security of the proposed scheme, we will first introduce two common security assumptions in this section, facilitating the subsequent security proofs.

1) The q-Strong Diffie-Hellman Assumption (q-SDH): The Strong Diffie-Hellman Assumption on the cyclic groups (G_1, G_2) is defined as follows: for a given tuple $(g_2, g_1, g_1^x, g_1^{x^2}, \ldots, g_1^{x^q})$, there does not exist an algorithm $\mathcal{A}$ that can, in polynomial time t, with a negligible probability ϵ, obtain a tuple $(s, A) = (s, g_2^{1/(x+s)})$. That is:

$$\Pr[\mathcal{A}(g_2, g_1, g_1^x, g_1^{x^2}, \ldots, g_1^{x^q}) \rightarrow (s, g_2^{1/(x+s)})] \leq \epsilon.$$

2) RXDH Assumption: Assuming $e : G_1 \times G_2 \rightarrow G_T$ is a bilinear map, the Randomized XDH assumption states that for any randomly chosen $a, b, z, z_i, r_i \in \mathbb{Z}_q^*$, the tuples $(P_1, aP_1, bP_1, abP_1, br_iP_1)$ and $(P_1, aP_1, bP_1, zP_1, z_iP_1)$ are computationally indistinguishable, where $i = 1, 2, \ldots, p$.

C. Lemma

To prove the security of the proposed scheme, we first prove the correctness of the following lemma. Assuming $P_{\text{pub}} = xP_1$; $ACC_j = (s_i + x)W_i$; $C_i = (s_i + x)^{-1}P_2$. The proposed signature proves to the verifier that signer possesses W_i, C_i, and s_i satisfying $e(P_{\text{pub}}, W_i + C_i) = e(T_2, ACC_j + P_2)$.

The proposed group signature scheme can actually be viewed as a special type of non-interactive zero-knowledge proof, used to demonstrate to the verifier that one possesses the group member private key $gsk[i] = (s_i, C_i, W_i)$. Specifically, non-interactive zero-knowledge proofs demonstrate three things to the verifier:

1) The group member possesses the secret key s_i corresponding to the identity id_i.
2) The group member holds the group member certificate C_i corresponding to the identity id_i, indicating that he has previously registered with the group manager.
3) The group member has the accumulator evidence W_i associated with the identity id_i, showing that the group member is still a legitimate group member and has not been revoked by the group manager.

Next, we will prove that the proposed group signature scheme satisfies the following properties:

- Correctness. It means that correctly generated group signatures are always successfully verified by verifiers.
- Zero knowledge under honest verifiers. It means that the proposed group signature scheme can be simulated.
- Soundness. This property means that the proposed group signature scheme exists extractors.

Lemma 1: The proposal group signature is correctness.

The proposed group signature scheme possesses correctness. In other words, if a signer possesses (W_i, C_i, s_i) and correctly executes the *Sign* algorithm to obtain the signature σ, then $Verify(\sigma)$ will always return 1.

- We first prove equation $s_uP_1 = (r_u + cu)P_1 = R_1 + cT_1$ hold. According to $s_u = r_u + cu$, we can conclude that $s_uP_1 = (r_u + cu)P_1 = (r_uP_1 + cuP_1) = R_1 + cT_1$. So $\hat{R}_1 = s_uP_1 - cT_1$. Similarly, we can prove that $s_uT_1 = (r_u + cu)T_1 = R_2 + cT_2$ hold. Therefore, we can prove $\hat{R}_2 = s_uT_1 - cT_2$ hold.
- Next, we set $W_i = w_iP_2$ and $C_i = c_iP_2$, we can get:

$$\begin{aligned}
&e(P_{pub}, A_1 + A_2)^{s_u} \cdot e(T_1, A_1 + A_2)^{s_s} \\
&= R_3 \cdot e(P_{pub}, A_1 + A_2)^{cu} \cdot e(T_1, A_1 + A_2)^{cs_i} \\
&= R_3 \cdot e(xP_1, u(w_i + c_i)P_2)^{cu} \cdot e(uP_1, u(w_i + c_i)P_2)^{cs_i} \\
&= R_3 \cdot e(P_1, P_2)^{(w_i+c_i)xucu} \cdot e(P_1, P_2)^{(w_i+c_i)u^2cs_i} \\
&= R_3 \cdot e(P_1, P_2)^{[(w_i+c_i)cu^2(x+s_i)]} \\
&= R_3 \cdot e(u^2P_1, w_i(x + s_i))P_2 + c_i(x + s_i)P_2)^c
\end{aligned}$$

  Due to $w_iP = W_i = ACC_{j-1}$ and $c_i = (x + s_i)^{-1}$, then $w_i(x + s_i)P_2 = ACC_j$ and $c_i(x + s_i)P_2 = P_2$. Thus, the original expression is equal to $R_3 \cdot e(T_2, ACC_j + P_2)^c$ and equation $\hat{R}_3 = e(P_{pub}, A_1 + A_2)^{s_u} \cdot e(T_1, A_1 + A_2)^{s_s} \cdot e(T_2, ACC_j + P_2)^{-c}$ hold.
- Finally, we prove $s_sP_1 + s_uP_{pub} = R_4 + cT_3$ hold. Due to $s_sP_1 + s_uP_{pub} = (r_s + cs_i)P_1 + (r_u + cu)P_{pub} = R_4 + cT_3$ So there is: $\hat{R}_4 = s_sP_1 + s_uP_{pub} - cT_3$ hold.

In summary, we have proven that $\hat{R}_i = R_i$ for $i = 1, 2, 3, 4$. Therefore, $c' = H(M, T_1, T_2, T_3, A_1, A_2, \hat{R}_1, \hat{R}_2, \hat{R}_3, \hat{R}_4) = c$, which means our signature scheme is correct.

Lemma 2: There exists a simulator Sim that can simulate the proposed signature without knowing (s_i, W_i, C_i), producing a group signature indistinguishable from a real one.

The simulator randomly generates $u, s \in \mathbb{Z}_q^*$, and randomly selects $W, C \in_R G_2$ and calculates $T_1 = uP_1$, $T_2 = uT_1$, $T_3 = sP_1 + P_{\text{pub}}$, $A_1 = uW$, $A_2 = uC$. Assuming the hardness of the RXDH and the discrete logarithm problem holds, from the adversary's perspective, the simulated (T_1, T_2, T_3) are elements randomly chosen from G_1, while (A_1, A_2) are elements randomly chosen from G_2. In the real group signature scheme, $(T_1, T_2, T_3, A_1, A_2)$ are randomized using the random number u, thus they are perceived by the adversary as elements randomly selected from the respective groups. Thus, the tuple $(T_1, T_2, T_3, A_1, A_2)$ generated by the simulator and a tuple generated by a real group member are indistinguishable.

For any given tuple $(T_1, T_2, T_3, A_1, A_2)$, the simulator randomly generates $c, s_u, s_s \in_R \mathbb{Z}_q^*$ and computes: $R_1 = s_uP_1 - cT_1, R_2 = s_uT_1 - cT_2$, $R_4 = s_sP_1 + s_uP_{\text{pub}} - cT_3$ and $R_3 = e(P_{\text{pub}}, A_1 + A_2)^{s_u}e(T_1, A_1 + A_2)^{s_s}e(T_2, ACC_j + P_2)^{-c}$. Thus, A tuple $(T_1, T_2, T_3, A_1, A_2, c, s_u, s_s)$ was simulated by the simulator. We can easily verify that the simulated tuple satisfies all verification equations (eq. 1) (eq. 3) and (eq. 2). In the real group signature, $s_u = r_u + cu$ and $s_s = r_u + c_s$, where u is random, making s_u and s_s random numbers. In the random oracle model, the hash function result is seen as a random number, hence s_u, s_s, and c are perceived as random by the adversary. In the simulation scheme, we randomly select s_u, s_s, and c, thus the simulated s_u, s_s, and c are indistinguishable from those in the real scheme.

In conclusion, the simulator simulates variables generated during a real group signature process without knowing the

secrets (s_i, W_i, C_i). If the discrete logarithm problem and the RXDH problem are hard, the tuples simulated by the simulator $(T_1, T_2, T_3, A_1, A_2, c, s_u, s_s)$ are indistinguishable from the proposed group signature.

Lemma 3: There exists an extractor for the proposal group signature scheme.

The signer generates $(T_1, T_2, T_3, A_1, A_2, R_1, R_2, R_3, R_4)$. Assuming the extractor receives two challenges, c and c', (We will later describe how to obtain c and c') it computes the corresponding s_u, s_s and s'_u, s'_s. Both of the responses provided above satisfy equations (eq. 1) to (eq. 2) as described.

For the convenience of narration, let: $\Delta c = c - c'$; $\Delta s_u = s_u - s'_u$ and $\Delta s_s = s_s - s'_s$ If the prover provides two zero-knowledge proofs that are both correct, then equation [1] holds for the challenge-response pairs c, s_u, s_s and c', s'_u, s'_s, i.e. $\hat{R_1} = s_u P_1 - cT_1$; $\hat{R_1} = s'_u P_1 - c'T_1$

1) Taking the difference of both sides of the equations (eq. 1), we have: $\Delta s_u P_1 = \Delta c T_1$, thus $T_1 = (\Delta s_u / \Delta c) P_1$. If let $\hat{u} = \Delta s_u / \Delta c$, then $T_1 = \hat{u} P_1$. Similarly, we have $\Delta s_u T_1 = \Delta c T_2$, then $T_2 = (\Delta s_u / \Delta c) T_1 = \hat{u}^2 P_1$.
2) Similarly, due to equations (eq. 2) hold, we can deduce $\Delta s_s P_1 + \Delta s_u P_{pub} = \Delta c T_3$. Transform the equation into: $T_3 = (\Delta s_s / \Delta c) P_1 + (\Delta s_u / \Delta c) P_{pub}$. If we let $\hat{s_i} = \Delta s_s / \Delta c$, then we obtain $T_3 = \hat{u} P_1 + \hat{s_i} P_{pub}$
3) For equation (eq. 3), applying the same method yields: $e(P_{pub}, A_1 + A_2)^{\Delta s_u} \cdot e(T_1, A_1 + A_2,)^{\Delta s_s} = e(T_2, ACC_j + P_2)^{\Delta c}$. We transformed the equation into: $e(T_2, ACC_j + P_2) = e(P_{pub}, A_1 + A_2)^{\Delta s_u / \Delta c} \cdot e(T_1, A_1 + A_2)^{\Delta s_s / \Delta c}$. Simplify it to get $e(T_2, ACC_j + P_2)^{\hat{u}^{-2}} = e(P_{pub}, A_1 + A_2)^{\hat{u}\hat{u}^{-2}} \cdot e(T_1, A_1 + A_2)^{\hat{s_i}\hat{u}^{-2}}$. So, we finally obtain equation $e(\hat{u}^{-2} T_2, ACC_j + P_2) = e(P_{pub}, \hat{u}^{-1} A_1 + \hat{u}^{-1} A_2) \cdot e(\hat{s_i} P_2, \hat{u}^{-1} A_1 + \hat{u}^{-1} A_2,)$

Let $W_i = \hat{u}^{-1} A_1$, $C_i = \hat{u}^{-1} A_2$, and $s_i = \hat{s_i}$. Then, we have $e(P_{pub}, W_i + C_i) = e(T_2, ACC_j + P_2)$. Therefore, the extractor *Ext* obtains tuple (W_i, C_i, s_i).

D. CPA-Full-Anonymity

We will prove that, under the random oracle model, anyone without the group private key cannot, within polynomial time and with non-negligible probability, obtain the real identity of signers from group signatures. Otherwise, there exists a algorithm which can exploit this to solve the RXDH problem. For convenience, we define $(P_1, aP_1, bP_1, zP_1, br_1 P_1, br_2 P_1)$ as a DH tuple if $z = ab$. Conversely, if $z \neq ab$, it is not considered a DH tuple.

Theorem 1: The proposed group signature achieves CPA-full-anonymity if the RXDH assumption holds.

Our proof commences with a game wherein Challenger $\mathcal{C}$ and adversary $\mathcal{A}$ interact. For ease of description, suppose the proposed group signature be $\sigma = (\sigma_1, c, \sigma_2)$, where $\sigma_1 = (T_1, T_2, T_3, A_1, A_2)$ and $\sigma_2 = (s_u, s_s)$.

1) Challenger $\mathcal{C}$ generates a random number $a \in_R \mathbb{Z}_q^*$ as the group private key and computes $P_{pub} = aP_1$. Then, using the Setup() method, challenger $\mathcal{C}$ obtains $Params = \{q, G_1, G_2, G_T, e, P_1, P_2, P_{pub}, ACC_0, H(\cdot)\}$. Then, $\mathcal{C}$ sends the group public key *Params* to adversary $\mathcal{A}$.
2) Assume the private keys of group members are $gsk[k]$. $\mathcal{C}$ sends the set of private keys $\{gsk[k]\}_{i=1}^{n}$ to $\mathcal{A}$, Using these private keys, $\mathcal{A}$ can generate up to q_s signatures, obtaining the signature set $\{\sigma_i\}_{i=1}^{q_s}$.
3) At any time, adversary $\mathcal{A}$ can query a random oracle $\mathcal{O}$ to obtain the hash value of message m. Initially, the challenger $\mathcal{C}$ creates an empty hash list L. When a new hash query m_i is received, $\mathcal{C}$ randomly selects an element $h_i \in \mathbb{Z}_p$ as the response and stores (m_i, h_i) in L. If the adversary's query m_* is in the list L, the existing value in the list is used as the response, ensuring the same query yields the same response.
4) Adversary $\mathcal{A}$ randomly selects two group members and sends the challenge $ID_0, ID_1 \in_R G_1$ to the challenger $\mathcal{C}$, where $ID_0 = H(id_0)P_1$ and $ID_1 = H(id_1)P_1$ represent the identities of the two group members. $\mathcal{C}$ randomly selects a bit $i \in_R \{0, 1\}$ and generates a simulated signature response to $\mathcal{A}$ as follows:
   $\mathcal{C}$ generates a random number $b \in_R \mathbb{Z}_q^*$ and a random element $Z \in_R G_1$, and computes $\sigma_1 = (T_1 = bP_1, T_2 = bT_1, T_3 = ID_c + Z, A_1 = bW_i, A_2 = bC_i)$. By Lemma 2, it can be concluded that there exists a simulator that can simulate $\sigma = (\sigma_1, c, \sigma_2)$ using only σ_1 without group member private key (s_i, W_i, C_i). Challenger $\mathcal{C}$ uses this simulator to generate σ and returns $\sigma = (\sigma_1, c, \sigma_2)$ as group signature for the identity information ID_i to $\mathcal{A}$. $\mathcal{C}$ compute R_1, R_2, R_3, R_4 using the simulated σ, and set the hash value at $(T_1, T_2, T_3, A_1, A_2, R_1, R_2, R_3, R_4)$ as c, which is then stored in the hash list L. If $\mathcal{A}$ has queried this hash value before and provided a response $c' \neq c$ at that time, then the game fails.
5) Adversary $\mathcal{A}$ outputs its guess i' for i. If $i' = i$, the challenger outputs 1; otherwise, it outputs $\perp$.

We now construct an algorithm $\mathcal{B}$ that leverages adversary $\mathcal{A}$ to solve the RXDH hard problem. The input to algorithm $\mathcal{B}$ is $(P_1, T_1, T_2, T_3, A_1, A_2)$, where $\{T_i\}_{i=1}^{3}$ and $\{A_i\}_{i=1}^{2}$ is random element in G_1. $\mathcal{B}$ first calls $\mathcal{A}$ to obtain random elements $ID_0, ID_1 \in_R G_1$, then randomly selects $i \in \{0, 1\}$. Next, B uses the simulator from Lemma 2 to generate a simulated group signature σ for ID_i based on input and sends σ to $\mathcal{A}$. Then, $\mathcal{A}$ guesses i' based on the aforementioned game. If $i' = i$, algorithm B outputs 1, indicating that it believes input is a valid RXDH tuple; otherwise, it outputs 0.

Let event A represent "algorithm B output 1". Then the advantage of $\mathcal{B}$ in solving the RXDH hard problem is:

$$Adv_{RXDH}^{\mathcal{B}}(A) = \mid Pr[A|Z = abP_1] - Pr[A|Z \neq abP_1] \mid$$

Now, we analyze the probability calculation method for $\mathcal{B}$ to solve the RXDH hard problem based on the above game.

If the challenger $\mathcal{C}$ and adversary $\mathcal{A}$ successfully complete the simulation process [35], we deem $\mathcal{C}$'s simulation as successful; otherwise, it is considered a failed simulation. Let event S denote "challenger $\mathcal{C}$ conducted a successful simulation", then according to the law of total probability, we have:$Pr[A|Z = abP_1] = Pr[A|Z = abP_1 \wedge S] \cdot Pr[S] + Pr[A|Z = abP_1 \wedge \overline{S}] \cdot Pr[\overline{S}]$; $Pr[A|Z \neq abP_1] = Pr[A|Z \neq abP_1 \wedge S] \cdot Pr[S] + Pr[A|Z \neq abP_1 \wedge \overline{S}] \cdot Pr[\overline{S}]$

Let event B represent "the event that $\mathcal{A}$ outputs $i' = i$ in the above game". For convenience of description, we assume $P_S = Pr[S]$, $P_T = Pr[B|Z = abP_1]$ and $P_F = Pr[B|Z \neq abP_1]$. When $Z = abP_1$ and the simulation is successful, the simulated scheme is the real scheme. Therefore, the probability of event B occurring is the same as the probability of adversary $\mathcal{A}$ breaking the real scheme. Therefore, we have:

$$P_T = Pr[B|Z = abP_1] = Pr[A|Z = abP_1 \wedge S]$$

Similarly, only when the above game succeeds does adversary $\mathcal{A}$ output a guess i', making event B potentially possible. Therefore, we have:

$$P_F = Pr[B|Z \neq abP_1] = Pr[A|Z \neq abP_1 \wedge S]$$

Therefore, the advantage of algorithm $\mathcal{B}$ in solving the RXDH hard problem is given by:

$$Adv^{\mathcal{B}}_{RXDH}(A) = | P_S \cdot (P_T - P_F) |$$

In the above game, $\mathcal{C}$ generates a signature for ID_i using the simulator in Lemma 2, ensuring it is indistinguishable from those produced by the real group signature scheme. Now, let's analyze the probability of $\mathcal{B}$ solving the RXDH hard problem based on the aforementioned game. The elements chosen by $\mathcal{C}$, b and Z, have two possibilities: either $Z = abP_1$ or $Z \neq abP_1$:

- If $Z = abP_1$, $\sigma = (\sigma_1, c, \sigma_2)$ generated by challenger $\mathcal{C}$ in the game is produced according to the simulator described in Lemma 2, thus indistinguishable from the real scheme. Let event D represent "$\mathcal{A}$ breaks the proposed scheme". Assuming that $\mathcal{A}$ can break our proposed scheme with a non-negligible advantage ϵ, then $\epsilon = 2(Pr[D] - 1/2)$. Therefore, we have $Pr[D] = 1/2 + \epsilon/2$. Since the simulated scheme by challenger $\mathcal{C}$ is indistinguishable from the genuine scheme, we have $P_T = Pr[D] = 1/2 + \epsilon/2$ and $P_T = Pr[Exp^{\text{Anon-b}}_{\mathcal{GS},\mathcal{A}}(\lambda) = 1]$.
- If $Z \neq abP_1$, both b and Z are randomly generated. Thus, T_3 essentially corresponds to $\mathcal{C}$ encrypting identity id_c with a one-time pad. According to the construction method of signature σ by the simulator, σ_2 is randomly generated. In this case, Since the signature σ contains no information about identity id_i, and $\mathcal{A}$ gains no advantage in guessing i, we have $P_F = 1/2$.

In the game between $\mathcal{C}$ and $\mathcal{A}$, if a hash collision occurs, the game fails. Assuming $\mathcal{A}$ performs a total of q_H hash queries. Due to the probability of a hash collision occurring being no higher than $q_H/2^\lambda$, the probability of a successful simulation is no less than $(1 - q_H/2^\lambda)$. So, the advantage of $\mathcal{B}$ in solving the RXDH problem is: $Adv^{\mathcal{B}}_{RXDH}(A) = | P_S \cdot (P_T - P_F) | = P_S \cdot | (Pr[Exp^{\text{Anon-b}}_{\mathcal{GS},\mathcal{A}}(\lambda) = 1] - 1/2) | = P_S \cdot Adv(\lambda) \geq P_S \cdot | (1/2 + \epsilon/2 - 1/2) | = (1 - q_H/2^\lambda) \times \epsilon/2$. The time cost for the $\mathcal{B}$ to perform the aforementioned simulation is $O(1)$.

Thus, we have demonstrated that the simulation of the challenger in the game is indistinguishable from the real group signature scheme, and the probability of successful simulation is $(1 - q_H/2^\lambda)$. Furthermore, if $\mathcal{A}$ can break the anonymity of our scheme with a non-negligible advantage ϵ (in this case, $Adv(\lambda) = \epsilon/2$ is also non-negligible) within polynomial time t (denoted as (t, ϵ)), then $\mathcal{B}$ can use $\mathcal{A}$ to solve the RXDH hard problem with an advantage of $(t + O(1), (1 - q_H/2^\lambda)\epsilon/2)$. Therefore, if the RXDH problem is hard, $Adv(\lambda)$ is negligible, and our scheme satisfies CPA-full-anonymity.

E. Full-Traceability

In the Industrial Internet of Things, the vast number of devices and the limited resources of most devices can lead to incidents where some devices are attacked and send unauthorized data, disrupting industrial production. To address this, our solution offers a traceability feature for unauthorized devices, enabling effective management of these entities. Full-traceability is a stronger security goal [33] that encompasses several key properties, including unforgeability, resistance to coalition attacks, and traceability.

We now prove that the proposed group signature scheme possesses full-traceability as defined by Bellare et al. [33].

Forking Lemma [36]: Let $\mathbb{F}$ be a set with at least 2 elements, i.e. $|\mathbb{F}| \geq 2$, and q be a positive integer. Randomly select $m \in_R \mathbb{Z}_q$ and $(f_1, \ldots, f_q) \in_R \mathbb{F}$. For a probabilistic polynomial-time algorithm $\mathcal{A}$, defined as $(J, \sigma) = \mathcal{A}(m, f_1, \ldots, f_q)$, we define $acc = \Pr[J \geq 1 | \mathcal{A}(m, f_1, \ldots, f_q)]$. An intuitive interpretation of the above is: after conducting q hash and signature queries, the adversary obtains q corresponding $(f_1, \ldots, f_q)$. Then, based on this information, the adversary uses algorithm $\mathcal{A}$ to forge a signature on m with probability acc (where $J = 0$ indicates failure to forge). We define a forking algorithm $F_{\mathcal{A}}(m)$ associated with $\mathcal{A}$ as follows:

1) Randomly generate $\rho \in_R \{0, 1\}$ and $(f_1, \ldots, f_q) \in_R \mathbb{F}$.
2) Run algorithm $\mathcal{A}$ to obtain $(J, \sigma) = \mathcal{A}(m, f_1, \ldots, f_q, \rho)$. If $J = 0$, return $(0, \perp, \perp)$.
3) Randomly generate $(f'_J, \ldots, f'_q) \in_R \mathbb{F}$.
4) Run algorithm $\mathcal{A}$ again to obtain a new results $(J', \sigma') = \mathcal{A}(m, f_1, \ldots, f_{J-1}, f'_J, f'_q, \rho)$. If $f_J \neq f'_J \wedge J = J'$, return (J, σ, σ'); otherwise, return $(0, \perp, \perp)$.

Here, $\sigma = (\sigma_1, c, \sigma_2)$ and $\sigma' = (\sigma_1, c', \sigma'_2)$ are both valid signatures on message m, with $c' \neq c$ and $\sigma_2 \neq \sigma'_2$. If we let $succ = Pr[J = 1 : m \in_R \mathbb{Z}_q; (J, \sigma, \sigma') = F_{\mathcal{A}}(m)]$, and the signature σ can be simulated without knowledge of the signing key, and the distributions of the simulated signature and the genuine signature are indistinguishable, Bellare et al. [37] proved that $succ \geq acc(\frac{acc}{q} - \frac{1}{|\mathbb{F}|}) = \varepsilon$, that is:

$$\Pr[J = 1 : m \in_R \mathbb{Z}_q; (J, \sigma, \sigma') = F_{\mathcal{A}}(m)] \geq \varepsilon.$$

Theorem 2: If the 1-SDH assumption holds, then the proposed group signature is fully-traceable.

Our proof commences with a game wherein $\mathcal{C}$ and $\mathcal{A}$ interact. $\mathcal{C}$ first generates the necessary public parameters for the group signature and issues group member private keys $gsk[i]$ for group member. Then, $\mathcal{A}$ can collude with any number of group members to attempt to forge a signature that can not be opened or cannot be traced back to any specific group member identity. To achieve this goal, $\mathcal{A}$ can run polynomial number of hash and signature queries at any time. Finally, the $\mathcal{A}$ outputs a forged signature $\sigma = (\sigma_1, c, \sigma_2)$.

Setup Phase: The challenger $\mathcal{C}$ generates a random number $x \in_R \mathbb{Z}^*_q$ as the group private key and computes $P_{\text{pub}} = xP_1$. Subsequently, using the Setup() method, it generates the

other parts of the group public key, obtaining $params = \{q, G_1, G_2, G_T, e, P_1, P_2, P_{\text{pub}}, ACC_0, H(\cdot)\}$.

Suppose the set of group members is $\mathbb{U} = \{U_1, U_2, \ldots, U_n\}$ and the identity of U_i is id_i. In this stage, the challenger $\mathcal{C}$ generates the group private key $gsk[i] = (s_i, W_i, C_i)$ for each group member U_i, where $s_i = H(id_i)$. Then, it computes $acc = \prod_{i=1}^{n}(s_i + x)$ and updates the accumulator value to $ACC_j = accACC_0$. Finally, $\mathcal{C}$ sends $Params = \{q, G_1, G_2, G_T, e, P_1, P_2, P_{\text{pub}}, ACC_j, H(\cdot)\}$ to $\mathcal{A}$.

Collusion Phase: The adversary $\mathcal{A}$ colludes with any number of group members to obtain the group private keys $gsk[i]$ of any number of group members. Let $\mathbb{L} = \{U_1, U_2, \ldots, U_t\}$ denote the set of group members colluding with the adversary.

Query Phase: To grant the adversary $\mathcal{A}$ with substantial capabilities, we assume that $\mathcal{A}$ can conduct polynomially many hash queries and signature queries at any time. For each distinct query, $\mathcal{C}$ responds as follows:

- **Hash queries:** The adversary can conduct a polynomial number of hash queries at any time, and the challenger $\mathcal{C}$ maintains an initially empty list H. When the adversary $\mathcal{A}$ queries a message m_i for hashing, it selects a random element $h_i \in_R \mathbb{Z}_q^*$ as the response and adds (m_i, h_i) to the list H. If the message m_i queried by the adversary already exists in H, then the response to $\mathcal{A}$ is the corresponding h_i stored in H.
- **Signature queries:** The $\mathcal{A}$ can conduct a polynomial number of signature queries at any time. Suppose the $\mathcal{A}$ wants to query a group member with identity id^* for a signature on message M. $\mathcal{C}$ responds to the $\mathcal{A}$'s query as follows: $\mathcal{C}$ randomly selects an $s_* \in_R \mathbb{Z}_q^*$ as the hash value for id^* and stores (id_*, s_*) in the hash list H. Then, leveraging Lemma 2, the $\mathcal{C}$ computes $\sigma_1 = (T_1, T_2, T_3, A_1, A_2)$ with (s_*, W_*, C_*), followed by using the simulator to generate a simulated signature $\sigma = (\sigma_1, c, \sigma_2)$. Finally, let c be the hash value at $M, T_1, T_2, T_3, A_1, A_2, R_1, R_2, R_3, R_4$. If the $\mathcal{A}$ has previously conducted a hash query here and received a value c', where $c' \neq c$, the game fails. Otherwise, the signature σ is returned to $\mathcal{A}$ as the result of the signature query. Since s_* is randomly chosen, the way $\mathcal{C}$ generates σ is indistinguishable from the simulator's generation in Lemma 2, thus the response σ to this signature query is indistinguishable from the genuine scheme.

Forgery Phase: The adversary outputs a forged group signature $\sigma = \{\sigma_1, c, \sigma_2\}$. If the forged signature satisfies the following properties, then the adversary wins the game:

- The challenger runs the *Open* algorithm and obtains $Open(\sigma) = \perp$ or $Open(\sigma) = id^*$, and id_* is not the identity of any group member. We refer to the adversary who produces such a signature as a Type I forger.
- The challenger runs the *Open* algorithm and obtains $Open(\sigma) = id^*$ where id_* is the identity of a legitimate group member $U_* \notin \mathbb{L}$. We refer to the adversary who produces such a signature as a Type II forger.

The analysis of the probability that the adversary $\mathcal{A}$ wins the above game proceeds as follows:

In the game between $\mathcal{C}$ and $\mathcal{A}$, if a hash collision occurs during signature queries, the simulation fails. Suppose $\mathcal{A}$ conducts a total of q_H hash queries, then the probability of a hash collision occurring is at most $q_H/2^{\lambda}$. Therefore, the probability of successful simulation is at least $(1 - q_H/2^{\lambda})$. Here, λ represents the size of elements in the hash space.

Suppose $\mathcal{A}$ conducts q_H hash queries and obtains results $\{h_1, h_2, \ldots, h_{q_H}\}$, along with q_S signature queries and obtains results $\{s_1, s_2, \ldots, s_{q_S}\}$. Let $\mathbb{F} = \{f_1, f_2, \ldots, f_{q_H+q_S}\}$, where $f_i = h_i$ for $i = 1, 2, \ldots, q_H$ and $f_{q_H+i} = s_i$ for $i = 1, 2, \ldots, q_S$. If the $\mathcal{A}$ can break the signature scheme in polynomial time t ($\epsilon = \Pr[J \geq 1 | \mathcal{A}(m, f_1, \ldots, f_q)]$) with a non-negligible probability $acc = \epsilon$, then the algorithm $\mathcal{B}$ can execute forking lemma [36] to obtain $(m, \sigma, \sigma') = F_A(M)$. Here, $\sigma = (\sigma_1, c, \sigma_2)$, $\sigma' = (\sigma_1, c', \sigma_2')$, satisfying $c' \neq c$ and $\sigma_2 \neq \sigma_2'$. Utilizing the extractor in Lemma 3, we derive $(s_i, W_i, C_i) = Ext(\sigma, \sigma')$, satisfying $e(P_{\text{pub}} + s_i P_1, W_i + C_i) = e(P_1, ACC_j + P_2)$. Simplifying this equation, we arrive at $e((x + s_i)P_1, (acc + 1)^{-1}(W_i + C_i)) = e(P_1, P_2)$. Let $A = (acc + 1)^{-1}(W_i + C_i)$, and thus $A = (\frac{1}{x+s_i})P_2$. This signifies that we have found a solution (s_i, A) to the 1-SDH problem.

It's clear that the time cost for the algorithm $\mathcal{B}$ to complete the above game is $O(q_H + q_S)$. Therefore, when the adversary can forge an untraceable group signature with (t, ϵ) advantage, the algorithm $\mathcal{B}$ can utilize the forking lemma to find two signatures σ and σ' with a probability not lower than $succ$. Thus, the advantage of the algorithm $\mathcal{B}$ in breaking the 1-SDH problem is not lower than $\epsilon \cdot (1 - \frac{q_H}{2^{\lambda}}) \cdot succ$. In other words, $Adv_{1-SDH}^{\mathcal{B}} \geq \epsilon \cdot (1 - \frac{q_H}{2^{\lambda}}) \cdot \epsilon \cdot (\frac{\epsilon}{q} - \frac{1}{h})$. As $(q \leq h = q_H + q_S)$, we have $(\frac{\epsilon}{q} - \frac{1}{h}) \geq \frac{\epsilon}{q_H + q_S}$. Thus, $Adv_{1-SDH}^{\mathcal{B}} \geq (1 - \frac{q_H}{2^{\lambda}}) \cdot \frac{\epsilon^3}{q_H + q_S}$. In conclusion, the algorithm $\mathcal{B}$ can solve the 1-SDH hard problem with $(t + O(q_H + q_S), (1 - \frac{q_H}{2^{\lambda}}) \cdot \frac{\epsilon^3}{q_H + q_S})$.

F. Security Analyses

We will now analyze in detail the security properties satisfied by our scheme:

1) Unforgeability: Unforgeability refers to the impossibility of any unregistered device to forge a correct identity authentication message. We have already demonstrated that our scheme possesses full-traceability, which implies its unforgeability [33] (refer to Chapter 3 of [33]).
2) Anonymity: Anonymity means that the real identities of both parties in message communication remain confidential during the authentication process for those without access to the system's private key. In the security proof section above, we have proved that our scheme achieves CPA-full-anonymity. Clearly, it provides anonymity.
3) Traceability: Traceability means the ability to trace the real identity of the message sender from the authentication message. Domain manager can use the domain private key to run the *open* algorithm and obtain the identity of SD_i, Thus our cross-domain authentication scheme satisfies traceability.
4) Revocability: Revocability refers to the existence of a method to revoke current legitimate users, preventing them from generating valid messages for identity authen-

TABLE II
SECURITY PROPERTY COMPARISON

SP \ Schemes	BASA [24]	CCAP [8]	IRBA [15]	Our Scheme
unforgeability	✓	✓	✓	✓
anonymity	✗	✓	✗	✓
raceability	✓	✓	✓	✓
revocability	✓	✓	✗	✓
conditional privacy-preserving	✗	✓	✗	✓
unlinkability	✓	✓	✓	✓
resistance coalition Attacks	✗	✗	✗	✓
non-repudiation	✓	✓	✓	✓

TABLE III
EXPERIMENTAL SETUP

Parameter	Definition
Processor	Intel Core i7-12700 2.10 GHz
RAM	16.0 GB
Implementation Language	C++
Hash Function	SHA256
Elliptic Curve	BLS12383
Signature & Certificate	ECDSA
Cryptographic Library	Miracl core [40]
Measurement Metric	Average execution time (running 100 times)

tication. In the proposed scheme, domain manager can use the *revoke* algorithm to revoke specific users, rendering them unable to generate valid group signatures and thus restricting their identity authentication. Therefore, our scheme achieves revocability.

5) Conditional Privacy-preserving: Conditional privacy-preserving refers to the protection of the identity of communication parties under certain conditions. However, once these conditions are breached, privacy will be exposed. In our scheme, the identity privacy of IIoT devices is protected as long as the group private key is unknown. However, individuals with the group private key can reveal their real identity through the *open* algorithm, thus achieving conditional privacy-preserving.
6) Unlinkability: Unlinkability refers to the inability of anyone to distinguish whether two messages come from the same sender unless they possess the keys required for traceability. Lemma 2 shows the proposed scheme is indistinguishable from signatures simulated by a simulator under the random oracle model. Thus, adversary cannot judge whether two messages come from the same sender, indicating that our scheme achieves unlinkability.
7) Resistance to Coalition Attacks: A coalition attack refers to a group of signers conspiring to generate a signature that cannot be traced back to any one of them [38]. In our full-traceability proof, we allow the adversary to query signatures of any group member, which is equivalent to permitting any number of signers to collude. Since our scheme achieves full-traceability, it is capable of resisting coalition attacks.
8) Non-repudiation: Non-repudiation refers to the inability of an entity to deny having sent a message. Since our scheme allows tracking the sender's real identity based on the message, it evidently satisfies non-repudiation.

We compare our scheme with the three recent cross-domain authentication schemes in IIoT [8], [24] [15], and the results are presented in Table II.

It is noteworthy that IRBA utilizes IBC cryptography for cross-domain authentication. During the signature verification process, the legitimacy of the signer's identity is confirmed using their public key (such as the signer's email), thereby preventing anonymity and conditional privacy-preserving. Additionally, the IRBA scheme does not provide revocation functionality for illegal devices. BASA introduces pseudo-anonymous identities to facilitate revocability, but still requires the use of the signer's public key (such as the signer's email) to verify the legitimacy of pseudo-anonymous identities, thus lacking anonymity and conditional privacy-preserving. All three comparison schemes require the use of the signer's public key during verification, allowing verifiers to determine whether two messages originate from the same sender, thereby failing to achieve unlinkability.

VII. EXPERIMENTAL EVALUATION

A. Experimental Setup

To assess the proposed scheme's practicality, we compared it with CCAP [8], BASA [24] and IRBA [15] in terms of computational and communication costs. Using C++, we implemented these schemes on a PC with an Intel Core i7-12700 2.10 GHz CPU and 16.0 GB RAM. We utilized the BLS12383 curve for bilinear pairings, SHA256 hash function, and Miracl core [39] library for cryptography. The specific experimental setup are presented in Table III.

We encapsulated the operations of each entity into a function to simulate the execution of the protocol, and measured the execution time of each entity during this process. The final experimental results were obtained by running each scheme 100 times and calculating the average values. In particular, the experimental settings for different schemes are as follows.

For CCAP, IIoT devices from different domains have their own certificates. The ECDSA algorithm was used to generate these certificates because this algorithm is widely used (e.g. it is the algorithm used in the SSL layer of HTTPS).

For the IRBA [15], the BLS12383 curve is used for bilinear pairings. This curve does not support symmetric bilinear pairings, so small modifications need to be made to the selection of group elements in the IRBA scheme. It is worth noting that this change has minimal impact on the performance.

B. Computation Cost

Firstly, we theoretically analyze the computational costs of each scheme. Due to the excessive number of operations involved in each scheme, we only count the number of time-consuming operations. For arithmetic operations, power operations, and hash functions of large integers, their costs can be ignored and will not be counted here. The following table IV provides the meanings represented by different symbols:

TABLE IV
THE SYMBOL DEFINITION USED

Symbol	Definition
G_T^m	The cost of point multiplication $g_{t1} \cdot g_{t2}$, where $g_{t1}, g_{t2} \in G_T$
G_T^p	The cost of exponential operation g_t^p , where $g_t \in G_T$
B	The cost of bilinear mapping $e(g_1, g_2)$
G_1^m	The cost of point multiplication aP_1, where $G_1 =< P_1 >$
G_2^m	The cost of point multiplication aP_2, where $G_2 =< P_2 >$
T_s	The time cost generating an ECDSA signature

TABLE V
TIME-CONSUMING CRYPTOGRAPHIC OPERATIONS

Scheme	SD_i^A	Aux_i^A	Aux_i^B	SD_i^B
BASA [24]	$2B + 2G_T^p$	$2B + G_T^p + G_T^m + G_2^m$	$2B + G_T^p + G_T^m + G_2^m$	-
CCAP [8]	$2T_s$	$8B + 21G_1^m + G_T^p$	$9B + 18G_1^m + 8G_T^p$	-
IRBA [15]	$4B + 2G_2^m$	-	$6B + 2G_T^p$	-
Our Scheme	$2B + 2G_T^p + 8G_1^m + 2G_2^m$	-	$3B + 3G_T^p + 7G_1^m$	-

Aux_i^A in the table represents the infrastructure that assists authentication in domain A, in addition to the IIoT devices themselves. For example, the Proxy Authentication Server (PAS) in the CCAP scheme or the Authentication Agent Server (AAS) in the BASA scheme. It is worth noting that the CCAP scheme can choose whether to perform anonymous authentication as needed. Due to the implementation of anonymous operations in our scheme, for fairness reasons, the following experiment uses an anonymous authentication version of the CCAP scheme. We theoretically calculate the cost of each scheme as shown in the table V:

According to the table, it can be seen that the most time-consuming operation performed by devices in domain A during the authentication process of the IRBA scheme is four bilinear pairings and two point multiplication operations on G_2 groups. The CCAP scheme introduces a new entity PAS (Proxy Authentication Server), and then transfers most calculations to PAS during the authentication process, which increases the interaction complexity in the authentication process. In order to achieve anonymity, the main time-consuming operations performed by this scheme include 8 bilinear pairings and 21 point multiplication operations on G_1. In contrast, our proposed scheme reduces the most time-consuming bilinear pairing to two, effectively reducing computational overhead.

In order to more accurately compare the computational costs of these schemes, we implemented each scheme in C++. This includes the details of each scheme, including arithmetic operations for large integers, hashing and so on. In particular, in CCAP [8] scheme, the cross-domain authentication verification operations are handled by the Proxy Authentication Server and the Verification Server. In contrast, for BASA [24], IRBA [15] and our scheme, the verification operations are carried out by the edge servers. The experimental results are shown in Fig. 4 and Table VI.

It can be seen that the total computational cost of the proposed scheme is the smallest. The CCAP and BASA solutions reduce the cost of IIoT devices by transferring computing to auxiliary devices. But this increases the interaction complexity and communication overhead during the authentication process and increases the security risk of the entire system.

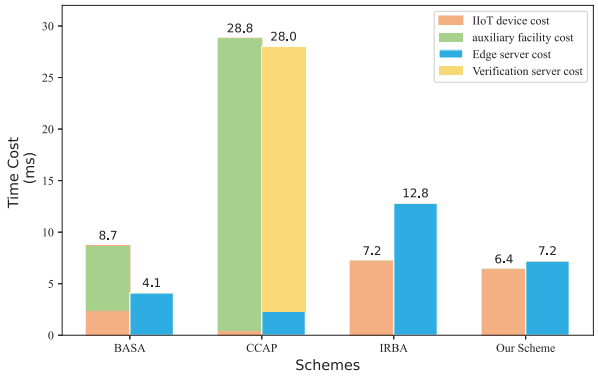


Fig. 4. Comparison of time costs for four protocols.

TABLE VI
COMPUTATION COST OF FOUR SCHEMES

Time(ms) / Phase / Scheme	Signature	Verification	Revoke	Authentication
BASA [24]	8.7	4.1	-	**12.8**
CCAP [8]	28.8	28.0	2.8	**56.8**
IRBA [15]	7.2	12.8	-	**20.0**
Our Scheme	6.5	7.2	0.5	**13.7**

C. Communication Cost

In this section, we will compare the communication costs of various solutions. We recorded the interaction between entities in each scheme during the authentication process, then counted the size of the data packets they transmitted, and recorded the number of communications. In the cryptographic tools we use, a large integer occupies 65 bytes, and points on G_1, G_2, and G_T are stored using point compression technology to reduce communication overhead. At this time, the points on them occupy 49 bytes, 98 bytes, and 294 bytes, respectively. In the CCAP scheme, certificates are required. In the experiment, we selected the commonly used ECDSA signature to generate certificates and used NIST256 for elliptic curves. At this point, the size of the certificate and the elements on the finite field are both 32 Bytes. Due to the fact that the size of message M is not fixed during the communication process, M is taken into account when calculating the communication overhead here.

Based on the above parameter settings, our proposed solution has a communication cost of $\sigma_A = \{M, T_1, T_2, T_3, A_1, A_2, c, s_u, s_s\} = (49\times3+98\times2+48\times3 = 487B)$ plus updated evidence. The total communication overhead for the W_i sent by the blockchain and the ACC_j sent by the blockchain during signature verification is $487 + 98 + 65 = 650B$. The communication cost of the IRBA scheme is $(M, \theta, \sigma, R, \epsilon, N) = (98 + 294 \times 2 + 49 + 48)$, which is 783B. Similarly, the communication cost of the CCAP scheme can be calculated as $(176n + 2438)$B, where n represents the number of members in the domain. The communication cost of the

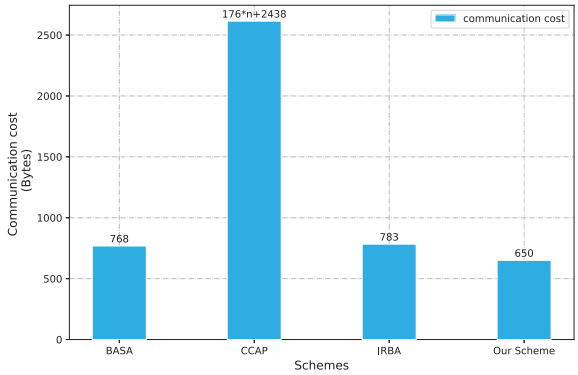


Fig. 5. Comparison of communication costs for four protocols.

BASA scheme is 768B. We will compare the communication costs of different schemes and obtain the following bar chart Fig. 5.

The CCAP scheme uses the Shamir secret sharing scheme to track the real identity of industrial IoT devices, thereby achieving conditional privacy-preserving. This leads to a linear correlation between the communication cost of the CCAP scheme and the number of domain members. As the number of domain members gradually increases, the cost of implementing anonymity in this scheme will significantly increase.

VIII. Conclusion

In this study, we propose a dynamic group signature scheme based on a dynamic accumulator and non-interactive zero-knowledge proof that achieves fast member joining and efficient revocation while ensuring conditional privacy-preserving. We combine this signature with blockchain to propose a dynamic cross-domain authentication scheme suitable for the IIoT. This scheme effectively protects the identity privacy of devices in industrial production and satisfies the dynamic requirements of IIoT devices in industrial scenarios. Under the RXDH and q-SDH assumptions, we demonstrated that the proposed scheme satisfies CPA-full-anonymity and full-traceability. Security analyses demonstrated that the proposed scheme can resist various attacks. Experimental results show that, compared to other relevant works, the proposed scheme has a lower computational and communication overhead and is more suitable for dynamic cross-domain authentication scenarios in the IIoT with high demands for privacy and efficiency. In future work, researchers can explore the development of pairing-free dynamic group signature schemes to further reduce the computational and communication overhead of IIoT devices. This approach could facilitate a reduced reliance on edge devices, lower the costs associated with actual deployment, accelerate their application in IIoT scenarios, and promote the advancement and progress of industrial IoT.

Acknowledgment

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

References

[1] S. Vitturi, C. Zunino, and T. Sauter, "Industrial communication systems and their future challenges: Next-generation Ethernet, IIoT, and 5G," *Proc. IEEE*, vol. 107, no. 6, pp. 944–961, Jun. 2019.

[2] H. Kopetz and W. Steiner, *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Cham, Switzerland: Springer, 2022.

[3] P. K. Malik et al., "Industrial Internet of Things and its applications in Industry 4.0: State of the art," *Comput. Commun.*, vol. 166, pp. 125–139, Jan. 2021.

[4] Q. Zhang, Y. Fu, J. Cui, D. He, and H. Zhong, "Efficient fine-grained data sharing based on proxy re-encryption in IIoT," *IEEE Trans. Depend. Secure Comput.*, vol. 21, no. 6, pp. 5797–5809, Nov. 2024.

[5] A. Badshah et al., "AAKE-BIVT: Anonymous authenticated key exchange scheme for blockchain-enabled Internet of Vehicles in smart transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 1739–1755, Feb. 2023.

[6] S. Sharma and V. K. Verma, "Security explorations for routing attacks in low power networks on Internet of Things," *J. Supercomput.*, vol. 77, no. 5, pp. 4778–4812, May 2021.

[7] L. Yang, Y. Liao, X. Cheng, M. Xia, and G. Xie, "Efficient edge data management framework for IIoT via prediction-based data reduction," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 12, pp. 3309–3322, Dec. 2023.

[8] F. Tong, X. Chen, K. Wang, and Y. Zhang, "CCAP: A complete cross-domain authentication based on blockchain for Internet of Things," *IEEE Trans. Inf. Forensic Security*, vol. 17, pp. 3789–3800, 2022.

[9] S.-G. Hao, L. Zhang, and G. Muhammad, "A union authentication protocol of cross-domain based on bilinear pairing," *J. Softw.*, vol. 8, no. 5, pp. 1094–1100, May 2013.

[10] L. Wang, Y. Tian, and D. Zhang, "Toward cross-domain dynamic accumulator authentication based on blockchain in Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 4, pp. 2858–2867, Apr. 2022.

[11] Y. Chen, J. Li, C. Liu, J. Han, Y. Zhang, and P. Yi, "Efficient attribute based server-aided verification signature," *IEEE Trans. Services Comput.*, vol. 15, no. 6, pp. 3224–3232, Nov. 2022.

[12] G. L. Millán, M. G. Pérez, G. M. Pérez, and A. F. G. Skarmeta, "PKI-based trust management in inter-domain scenarios," *Comput. Security*, vol. 29, no. 2, pp. 278–290, Mar. 2010.

[13] S. Matsumoto and R. M. Reischuk, "IKP: Turning a PKI around with decentralized automated incentives," in *Proc. IEEE Symp. Security Privacy (SP)*, Sep. 2017, pp. 410–426.

[14] J. Li, Y. Chen, J. Han, C. Liu, Y. Zhang, and H. Wang, "Decentralized attribute-based server-aid signature in the Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4573–4583, Mar. 2022.

[15] X. Jia et al., "IRBA: An identity-based cross-domain authentication scheme for the Internet of Things," *Electronics*, vol. 9, no. 4, p. 634, Apr. 2020.

[16] 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.

[17] L. Xue, H. Huang, F. Xiao, and W. Wang, "A cross-domain authentication scheme based on cooperative blockchains functioning with revocation for medical consortiums," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 3, pp. 2409–2420, Sep. 2022.

[18] H. Xiong, J. Chen, Q. Mei, and Y. Zhao, "Conditional privacy-preserving authentication protocol with dynamic membership updating for VANETs," *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 3, pp. 2089–2104, May 2022.

[19] Q. Zhang, J. Wu, H. Zhong, D. He, and J. Cui, "Efficient anonymous authentication based on physically unclonable function in industrial Internet of Things," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 233–247, 2023.

[20] G. Cheng et al., "Conditional privacy-preserving multi-domain authentication and pseudonym management for 6G-enabled IoV," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 10206–10220, 2024.

[21] F. Wang, J. Cui, Q. Zhang, D. He, C. Gu, and H. Zhong, "Blockchain-based lightweight message authentication for edge-assisted cross-domain industrial Internet of Things," *IEEE Trans. Depend. Secure Comput.*, vol. 21, no. 4, pp. 1–18, Jul. 2024.

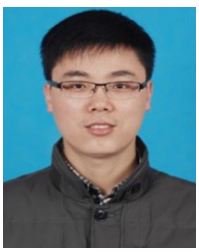
[22] J. Chen, Z. Zhan, K. He, R. Du, D. Wang, and F. Liu, "XAuth: Efficient privacy-preserving cross-domain authentication," *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 5, pp. 3301–3311, Sep./Oct. 2022.

[23] Z. Kang, J. Li, J. Shen, J. Han, Y. Zuo, and Y. Zhang, "TFS-ABS: Traceable and forward-secure attribute-based signature scheme with constant-size," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 9514–9530, Sep. 2023.

[24] 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.

[25] Q. Zhang, R. Wang, Y. Gan, and Y. Yin, "A cross-domain alliance authentication scheme based on bilinear group," *Appl. Math. Inf. Sci.*, vol. 8, no. 3, pp. 1313–1317, May 2014.
[26] J. Cui, N. Liu, Q. Zhang, D. He, C. Gu, and H. Zhong, "Efficient and anonymous cross-domain authentication for IIoT based on blockchain," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 2, pp. 899–910, Mar. 2023.
[27] M. Wang, L. Rui, Y. Yang, Z. Gao, and X. Chen, "A blockchain-based multi-CA cross-domain authentication scheme in decentralized autonomous network," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 3, pp. 2664–2676, Sep. 2022.
[28] Q. Zhang, Y. Gan, Q. Zhang, R. Wang, and Y. Tan, "A dynamic and cross-domain authentication asymmetric group key agreement in telemedicine application," *IEEE Access*, vol. 6, pp. 24064–24074, 2018.
[29] L. Nguyen, "Accumulators from bilinear pairings and applications," in *Topics in Cryptology—CT-RSA 2005*, A. Menezes, Ed., Berlin, Germany: Springer, 2005, pp. 275–292.
[30] M. Shen, J. Duan, L. Zhu, J. Zhang, X. Du, and M. Guizani, "Blockchain-based incentives for secure and collaborative data sharing in multiple clouds," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1229–1241, Jun. 2020.
[31] M. Shen, J. Zhang, L. Zhu, K. Xu, and X. Tang, "Secure SVM training over vertically-partitioned datasets using consortium blockchain for vehicular social networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5773–5783, Jun. 2020.
[32] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.
[33] M. Bellare, D. Micciancio, and B. Warinschi, "Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions," in *Advances in Cryptology—EUROCRYPT 2003*, E. Biham, Ed., Berlin, Germany: Springer, 2003, pp. 614–629.
[34] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology—CRYPTO 2004*, M. Franklin, Ed., Berlin, Germany: Springer, 2004, pp. 41–55.
[35] Y. M. F. Guo and W. Susilo, *Introduction to Security Reduction*, 1st ed., Cham, Switzerland: Springer, Jul. 2018.
[36] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–396, 2000.
[37] M. Bellare and G. Neven, "Multi-signatures in the plain public-key model and a general forking lemma," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, Oct. 2006, pp. 390–399.
[38] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme," in *Advances in Cryptology—CRYPTO 2000*, M. Bellare, Ed., Berlin, Germany: Springer, 2000, pp. 255–270.
[39] M. Scott. (2023). *Miracl Core*. [Online]. Available: https://github.com/miracl/core

Mingwei Zeng is currently a Research Student with the School of Computer Science and Technology, Anhui University. His research interests include the security of the industrial Internet of Things.



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 of the School of Computer Science and Technology, Anhui University. He has over 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 MOBILECOMPUTING, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON COMPUTERS, 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 networks, cloud computing security, and software-defined networking (SDN). He is on the Editorial Board of several international journals, such as *IET Communications*, *Security and Communication Networks*, and *Sensors*.

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. He has over 30 scientific publications in reputable journals (e.g., *Proceedings of the IEEE*, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and IEEE TRANSACTIONS ON COMPUTERS) and international conferences. His research interests include edge computing, computer systems, and security.

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 of 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 networks, cloud computing security, and software-defined networking (SDN).

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, and Shanghai Key Laboratory of Privacy Preserving Computation, MatrixElements Technologies, Shanghai, China. He has published over 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 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. His work has been cited more than 10000 times at Google Scholar. 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*.