Blockchain-Assisted Flexible Revocable Anonymous Authentication in Industrial Internet of Things

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

***Abstract*—In Industrial Internet of Things (IIoT) systems, data sharing between industrial departments is often utilized to optimize management models and improve decision-making efficiency. To enable secure data sharing, authentication between smart devices is critical. However, existing authentication schemes do not comprehensively consider data anonymity, data traceability, pseudonym management, and flexible revocation of devices, which cannot meet the needs of IIoT systems for security, real-time, and dynamicity. Therefore, we propose a blockchain-assisted lightweight authentication scheme. First, we design a lightweight authentication method based on Okamoto's protocol and elliptic curve cryptography, which achieves fast authentication of smart devices while ensuring data anonymity and traceability. Second, we design a two-level key derivation algorithm and combine it with blockchain technology to address the issue of pseudonym management. Smart devices can generate pseudonyms without requesting them from the key generation center and can be revoked flexibly. Third, security proof and analysis demonstrate that the proposed scheme achieves the security objectives and is resistant to various common attacks. Finally, the performance evaluation results show that our proposed scheme performs better than the other related schemes regarding computational and communication overheads.**

***Index Terms*—Authentication, consortium blockchain, Industrial Internet of Things (IIoT), key derivation.**

I. Introduction

In recent years, with the widespread use of new technologies such as cloud computing and edge computing in the Industrial Internet of Things (IIoT) [1], the interconnection between industrial departments has been promoted. In the IIoT system, there are many industrial services (e.g., different production tasks) [2], [3]. And smart devices from different industrial departments can collect corresponding real-time industrial data [4] according to the services they support. For example, in the steel-making process, smart devices with environmental monitoring capabilities need to collect data such as temperature and humidity in real time [5]. These industrial data are shared between smart devices [6], [7], which facilitates the dynamic optimization of production strategies, resulting in product quality improvement and production cost reduction [8], [9].

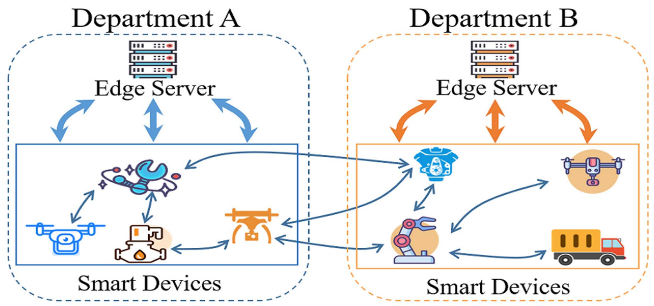


Fig. 1. Data sharing between two industrial departments.

Fig. 1 illustrates a typical data-sharing scenario between two industrial departments. The edge server manages the department's industrial services and smart devices [10], [11]. Smart devices are resource-limited (with limited computing power and storage capacity) [12], [13]. They can share real-time production information with other smart devices inside/outside the department [14]. In addition, when an smart device joins an industrial department, it can participate in multiple industrial services based on production needs. When the corresponding industrial service is completed, the smart device can exit the current industrial service or industrial department [15]. For example, a drone can join the transportation department to perform the task of transporting products. When that task is completed, the drone can leave the current department and join the quality monitoring department to perform product scanning and environmental monitoring tasks.

Although the above scenario can improve the flexible management of devices and the efficiency of production, frequent interactions between devices bring critical industrial data into insecure networks that are vulnerable to various attacks [16]. Specifically, an attacker can create chaos in industrial production by intercepting and tampering with the industrial data [17]. For

Received 28 April 2023; revised 2 September 2024; accepted 15 November 2024. Date of publication 21 November 2024; date of current version 25 December 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62272002, Grant 62202005, Grant 62372002 and Grant 62325209, in part by the Fundamental Research Funds for the Central Universities under Grant 2042023KF0203, 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. Recommended for acceptance by Dr. Zhen Wang. *(Corresponding author: Jie Cui.)*

Fengqun Wang, 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, Anhui University, Hefei 230039, China, and also with the Anhui Engineering Laboratory of IoT Security Technologies, Anhui University, Hefei 230039, China (e-mail: cuijie@mail.ustc.edu.cn).

Debiao He is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and also with the Shanghai Key Laboratory of Privacy Preserving Computation, MatrixElements Technologies, Shanghai 201204, China (e-mail: hedebiao@163.com).

Digital Object Identifier 10.1109/TNSE.2024.3503996

example, when information about production status is maliciously tampered with by an attacker and used by other devices, it can lead to severe damage to production machines or even the complete shutdown of an assembly line [18].

Authentication can effectively solve the above problems, but if the data anonymity is not guaranteed, it may expose the private information of IIoT [19]. For example, when the industrial data contains the real identity of a smart device, an attacker can mark that identity. Then, the attacker captures the data sent by that device for deeper analysis, thus obtaining privacy-sensitive information such as production steps and process parameters [20]. This threatens the security of industrial manufacturing [21]. However, absolute anonymity may lead to the sharing of malicious data. In this case, malicious devices could use anonymity to send false industrial data, intentionally interfering with the production process. For example, they may send false sensor data, causing the system to make error decisions. This malicious behavior will not only affect the normal operation of production, but also seriously impact product quality and production efficiency. Therefore, the introduction of data traceability mechanisms has become crucial. Through the tracing mechanism, system manager can open anonymous data, trace the device that sends malicious data, and take timely measures to prevent the threat to the production environment.

Many researchers are using elliptic curve cryptography to design pseudonym-based authentication schemes [22], [23]. On the one hand, pseudonym-based authentication schemes support the tracing and revocation of smart devices while ensuring data anonymity. On the other hand, compared to other typical authentication schemes (e.g., group signature-based authentication schemes), pseudonym-based authentication schemes are lightweight, which allows smart devices to process real-time data in time. However, these schemes still have the following challenges.

- The key generation center needs to pre-store many pseudonyms in the smart device. However, smart devices have limited storage capacity, and pre-storing pseudonyms increase the storage burden.
- Revocation mechanisms are rigid. This is because they usually only support revoking the real identity of a smart device, which means that the revoked smart device cannot continue to work in the IIoT system. This does not apply to the IIoT scenario we focus on, where smart devices can join other departments to perform new tasks after they leave the current department.

To address the above challenges, we propose a secure and efficient authentication scheme by using blockchain's decentralized and tamper-proof features. The contributions of the proposed scheme are as follows:

- We improve Okamoto's protocol and combine elliptic curve cryptography to design a lightweight authentication algorithm. The algorithm can fast authenticate smart devices while ensuring data anonymity and traceability.
- We design a two-level key derivation algorithm and combine it with blockchain to effectively manage certificates of smart devices, thereby reducing the storage burden on smart devices while enabling flexible revocation of smart devices. Under the supervision of blockchain, smart devices can derive pseudonyms by themselves.
- The security proof and analysis show that the proposed scheme is superior to other related schemes in terms of security and functionality. Experimental results demonstrate that the proposed scheme has advantages over other related schemes in terms of computational and communication overhead.

The rest of this paper is organized as follows. Section II presents existing authentication schemes. Section III offers the prerequisites. Section IV presents the system overview. Section V details the proposed scheme. Section VI presents the security proof and analysis. Section VII offers the performance evaluation. Section VIII concludes the paper.

II. Related Work

In this section, we introduce existing research related to authentication schemes.

In [14], Cui et al. proposed a cross-domain authentication scheme. In this scheme, they used pseudonyms to achieve anonymous communication between cross-domain devices. Like most traditional schemes [24], [25], this scheme imposes significant storage overhead on the smart device when it comes to pseudonym management because the smart device needs to pre-store many pseudonyms. In [26], Yang et al. proposed a conditional privacy-preserving authentication scheme considering the pseudonym management problem. The scheme uses the idea of key derivation to reduce the storage pressure on the device. Unfortunately, the device has the challenge of complicated operations and frequent interactions between entities when requesting a new pseudonym. In [27], Lin et al. combined blockchain technology and key derivation protocols to design a lightweight anonymous authentication scheme. The scheme is used in IIoT system with high demand for real-time performance. However, the revocation mechanism of the scheme is still not flexible enough; e.g., it cannot be adapted to the scenario where smart devices join/leave a certain industrial department anonymously. In [28] and [23], the authors designed a lightweight authentication scheme using elliptic curve cryptography. In these two schemes, the authors store the system secret key in the device, allowing the device to generate pseudonyms. However, keeping the system secret key in the device is not appropriate. Firstly, in the event of an attack on the device, the system secret key could be compromised, threatening the entire system's security. Secondly, once the device has been revoked, the system secret key in all devices will need to be updated. Finally, a device with a system secret key can generate pseudonyms without restriction, which poses a significant challenge for device revocation and supervision. In addition, some researchers proposed other schemes to ensure the security of IIoT systems. For example, Karati et al. [29] proposed a secure and efficient authentication scheme for IIoT environments. The scheme ensures data integrity but not anonymity of the smart device, which may compromise the privacy of the IIoT. In [30], Zhang et al. proposed an anonymous authentication scheme for IIoT. In this scheme, group signatures are mainly used for smart devices to access the server anonymously. In [31], Li et al. proposed a blockchain-based aggregate signature for securing privacy-sensitive data in IIoT. Although schemes [30] and [31] can effectively guarantee data anonymity and traceability without storing massive pseudonyms, they all contain many time-consuming cryptographic operations (e.g., bilinear pairing). Therefore, they are unsuitable for IIoT scenarios with high data density and high requirements for real-time data.

Through the above analysis, the existing schemes cannot simultaneously satisfy the following four conditions while ensuring lower computation and communication overhead: (1) guaranteeing data anonymity; (2) ensuring data traceability; (3) solving the pseudonym management problem and reducing the storage overhead of smart devices; and (4) achieving flexible revocation of smart devices, i.e., using different revocation mechanisms according to different needs. Therefore, we are motivated to design a lightweight anonymous authentication scheme that simultaneously considers the above four conditions.

III. Preliminaries

This section introduces elliptic curve cryptosystem and Okamoto's protocol.

A. Elliptic Curve Cryptosystem

Let $\mathbb{F}_p$ be a finite field, where p is a prime number. Elliptic curve point E over $\mathbb{F}_p$ be defined by the equation: $y^2 = x^3 + ax + b \bmod p$, where $a, b \in \mathbb{F}_p$. O is an infinity point, O and other points on E form an elliptic curve group $\mathbb{G}$, where the order is q and the generator is P. The main properties of $\mathbb{G}$ are listed below:

- Elliptic Curve Diffie-Hellman Problem (ECDHP): $x, y \in \mathbb{Z}_q^*$, and $X = xP, Y = yP$, where $X, Y \in \mathbb{G}$. Given X and Y, for a probabilistic polynomial-time (PPT) adversary, it is hard to get xyP.
- Elliptic Curve Discrete Logarithm problem (ECDLP): $x \in \mathbb{Z}_q^*$, $Q = xP$, where $P, Q \in \mathbb{G}$ on curve E. For a PPT adversary, given Q, it is hard to obtain x.

B. Okamoto's Protocol

The Okamoto's protocol is an example of the Sigma protocol. Let P is a generator of the elliptic curve E and $Q \in \mathbb{G}$ be some arbitrary group element, where Q publicly available for all parties. There are $\pi \in \mathbb{Z}_q^*$, $\omega \in \mathbb{Z}_q^*$, and $C = \omega \cdot P + \pi \cdot Q$ (C reveals no information about ω and a random secret ω' can not open C as $\omega' \neq \omega$ unless one can solve the ECDLP [32]). In the Okamoto's protocol, prover can convince a verifier that he knows π and ω, without revealing anything about π and ω. The description of this protocol between prover A and verifier B is presented in the following steps:

- A $\rightarrow$ B: A first selects two random number $\omega_t \in \mathbb{Z}_q^*$ and $\pi_t \in \mathbb{Z}_q^*$, then A calculates the commitment $C_t = \omega_t \cdot P + \pi_t \cdot Q$ and sends the commitment to B.
- B $\rightarrow$ A: B selects a challenge $e \in \mathbb{Z}_q^*$ and sends it to A.
- A $\rightarrow$ B: A computes $\omega_z = \omega_t + e \cdot \omega, \pi_z = \pi_t + e \cdot \pi$, and sends the response (ω_z, π_z) to B.
- B: B checks if $\omega_z \cdot P + \pi_z \cdot Q = C_t + e \cdot C$. If so, B accepts ω_z and π_z; otherwise, B rejects them.

The proposed scheme uses Okamoto's protocol to protect the authorization credential information of smart devices. To enhance the verification efficiency of credentials, we transform Okamoto's protocol into a non-interactive protocol.

C. Blockchain

Blockchain is essentially a distributed database, and its security is guaranteed mainly by cryptography and consensus protocols [33]. Blockchain can be divided into three types: public blockchain, consortium blockchain, and private blockchain. In the proposed scheme, we use a consortium blockchain accessible only to authorized organizations. In addition, consortium blockchain can provide the following properties.

- *Decentralization:* The blockchain is maintained by all organizations together, not a single one.
- *Transparency:* Data in the blockchain is transparent to authorized organizations.
- *Tamper proofing:* Once data are written to the blockchain, no organization can tamper with them.

Smart contracts in blockchain are essentially an automatically executable computer program. Specifically, the content of a smart contract can be defined in advance, and when the conditions for triggering a smart contract are met, the program will be executed automatically.

The decentralized and tamper-proof features of blockchain ensure transparency and security in data management, thereby reducing the risks of improper revocation or tampering. In our proposed scheme, the blockchain is regarded as a trusted platform for storing public information such as temporary certificates and anonymous certificates of smart devices. The verifier can determine whether the smart device is legitimate by checking whether the smart device's temporary certificate and anonymous certificate are valid.

IV. System Overview

This section shows the involved system model and assumptions, together with attack model and design objectives.

A. System Model and Assumptions

We consider a smart factory as an example in IIoT. In a smart factory, each smart device supports different services, and only the services provided by smart devices with corresponding authorization credentials are trusted.

1) Entity Overview: As shown in Fig. 2, the system model consists of four entities, namely, Key Generation Center (KGC), Edge Servers (ES), Blockchain (BC), and Smart Devices (SD).

- **KGC**: It is a trusted entity, which is mainly responsible for the registration of IIoT entities and the deployment of smart contracts.
- **ES**: It has powerful computing and storage capabilities, which is not only the manager of an industrial department but also a member of BC.

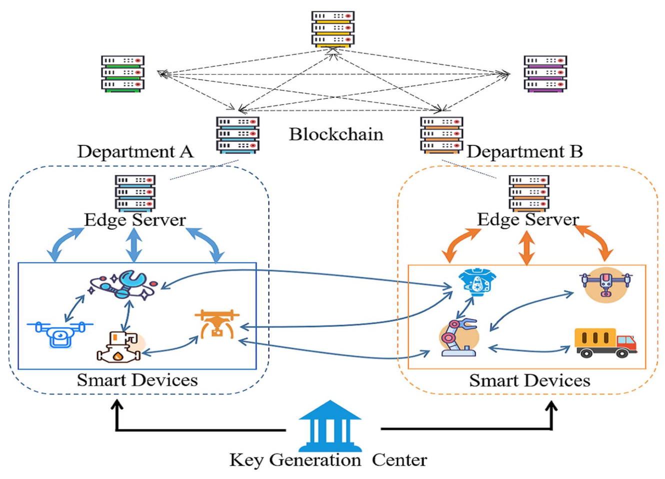


Fig. 2. Blockchain-assisted system model in IIoT.

- **BC**: It is a consortium blockchain maintained by ESs, which is responsible for storing public information of IIoT systems (e.g., anonymous certificates of SD).
- **SD**: It has limited computing and storage resources; it broadcasts the corresponding data to other SDs and ESs according to the service authorization.

In the blockchain-based system model, we consider three key features: (1) An SD can support multiple types of services. (2) Before SD participates in industrial services, it must join the corresponding industrial department. (3) ES acts as the manager of each industrial department, and if an SD wants to join this industrial department, it needs to be authorized by the corresponding ES.

B. Threat Model

In the proposed scheme, adversaries can be divided into external attackers and internal attackers. Internal attackers refer to the IIoT entities involved in the authentication process, i.e., edge servers and smart devices. External attackers refer to IIoT entities that do not participate directly in the authentication process. Both types of attackers can launch passive and active attacks. When launching passive attacks, the attacker mainly eavesdrops on the communication channels between industrial entities and tries to obtain confidential information. For example, the attacker can analyze the monitored data to mine the real identity of the signer. When launching an active attack, the attacker can access the communication channel between industrial entities and intercept, tamper, forge, and replay the packets transmitted by that communication channel.

C. Design Objectives

In this subsection, we present the proposed scheme's design objectives in terms of security and functional objectives.

1) Security Objectives:

- **Message integrity and authentication:** To ensure the security of IIoT systems, it is necessary to guarantee the integrity of the message and to ensure that the receiver can verify the validity of the message. Once a message is tampered with, the receiver should be able to find it in time.
- **Message anonymity:** To protect the privacy of SD, the real identity of SD should be hidden.
- **Unlinkability:** No malicious attacker can link two messages sent by the same SD.
- **Identity traceability:** ES/KGC can trace the identities of SDs according to different needs.
- **Resistance to various attacks:** To secure IIoT systems, our scheme should provide resistance to various common attacks, such as resist replay attacks, impersonation attacks, and modification attacks.

2) Functional Objectives:

- **Key derivation:** To reduce the storage pressure on SD, the proposed scheme should provide SD with the ability to derive pseudonyms/temporary identities and corresponding keys.
- **Batch authentication:** To improve the efficiency of authentication, the proposed scheme should be able to authenticate multiple messages simultaneously.
- **Flexible revocation:** The proposed scheme should provide a flexible revocation mechanism, i.e., it can revoke the SD's different certificates/identities based on specific needs.

D. Security Definition

Definition 1 (Existential Unforgeability Against 100.hosen Message Attacks, EU-CMA): In the EU-CMA security model, a signature scheme is (t, q_s, ϵ)-secure, which means in time t, no adversary $\mathcal{A}$ can win the game with advantage ϵ. In the game, $\mathcal{A}$ needs to perform q_s signature queries. In the queries process, the σ_{d_i} indicates a valid signature corresponding to the data d_i.

The game is an interaction between a PPT adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. Specifically, first, given a secure parameter, $\mathcal{C}$ generates a secret/public key pair (sk, pk) and sends pk to $\mathcal{A}$. Then, $\mathcal{A}$ adaptively selects data and makes signature queries to $\mathcal{C}$. Once the signature query corresponding to d_i is received, $\mathcal{C}$ returns the corresponding signature σ_{d_i} to $\mathcal{A}$. Finally, $\mathcal{A}$ selects a new data d_i' and returns the corresponding forged signature σ'_{d_i}.

V. Proposed Scheme

In this section, we first present the scheme overview and then describe the specifics of the scheme. The main notations used in the scheme are listed in Table I.

A. Scheme Overview

In this subsection, we first present the design goals and the issues considered in the scheme. Subsequently, we introduce the high-level workflow of the design scheme.

1) High-Level Idea: In the proposed scheme, our target is to achieve lightweight anonymous authentication between smart devices. In the scheme design process, we focus on the following four issues. Firstly, the proposed scheme meets the security objectives, including data anonymity and traceability. Second, the proposed scheme is lightweight enough to enable fast

TABLE I
NOTATIONS AND DEFINITIONS USED

Notations	Definitions
KGC	Key generation center
ES_j	$j-th$ edge server
SD_i	$i-th$ smart device
BC	Blockchain
msk/P_{pub}	System master secret/public key
ID_j	Real identity of ES_j
lsk_j/LPK_j	Long secret/public key of ES_j
RID_i	Real identity of SD_i
lsk_i	Long secret key of SD_i
tsk_i^x/TPK_i^x	$x-th$ temporary secret/public key of SD_i
pdi_i^x	$x-th$ private derivation information of SD_i
TI_i^x	$x-th$ temporary identity of SD_i
tc_i^x	$x-th$ temporary certificate of SD_i
$ask_{i,a}^y/APK_{i,a}^y$	$y-th$ anonymous secret/public key of TI_i^a
$pdi_{i,a}^y$	$y-th$ private derivation information of TI_i^a
$PID_{i,a}^y$	$y-th$ pseudonym identity of TI_i^a
$ac_{i,a}^y$	$y-th$ anonymous certificate of TI_i^a
Enc/Dec	Symmetric encryption/decryption

device authentication. Third, the proposed scheme supports key derivation to ease the storage pressure on smart devices. Finally, the proposed scheme can achieve flexible revocation of smart devices according to specific needs.

2) High-Level Workflow: As shown in Fig. 3, the whole authentication process can be divided into the following four phases.

1) **System initialization.** This phase is the foundation of the entire scheme, where KGC initializes the IIoT system and issues keys for each industrial entity. Notably, at this phase, SD can use the parameters distributed by KGC to derive a series of temporary identities and corresponding public/secret keys, which are used when the SD joins industrial departments.
2) **Joining request sending and verification.** Before an SD wants to join a new industry department, it needs to send a join request and be authenticated by the corresponding ES.
3) **Anonymous public and secret key derivation.** In this phase, SD uses the private derivation information to derive a series of pseudonym and anonymous public/secret keys, which are used to sign messages when the SD performs the corresponding industrial tasks. ES uses the private derivation information to derive a series of anonymous public keys.
4) **Message signing and verification.** In this phase, the message sender (an SD) signs with the derived secret key, and the corresponding message receiver (another SD) implements fast authentication with the assistance of the blockchain.

Note that in our proposed scheme, KGC/ES can trace to the corresponding identity of the SD and revokes it on demand. The reason is that we design a two-level key derivation algorithm, which is the basis for achieving effective management of pseudonyms and flexible revocation of devices simultaneously. The main idea behind the two-level key derivation is shown in Fig. 4. Specifically, the smart device can derive multiple temporary identities and corresponding keys based on the real identity. Also, the smart device can derive multiple pseudonyms and corresponding keys based on a temporary identity.

B. System Initialization

The KGC generates the system parameters, and then generates authorization credentials based on the service type and hides these credentials. Subsequently, during the registration phase, KGC issues the corresponding required secret and public parameters for ES and SD.

1) System Parameters Generation: Given the parameters of elliptic curve $(\mathbb{G}, q, P)$, KGC generates system parameters and initializes a consortium blockchain.

1) The KGC chooses a random number $msk \in \mathbb{Z}_q^*$ as the master secret key, and then calculates the corresponding system public key $P_{pub} = msk \cdot P$.
2) The KGC selects two hash functions: $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$, $H_2 : \{0,1\}^* \rightarrow \{0,1\}^*$.
3) The KGC keeps msk secretly and publishes the system public parameters $\{\mathbb{G}, q, \mathbb{Z}_q^*, P_{pub}, H_1, H_2\}$ via public channels.
4) The KGC selects preset ESs to construct a consortium blockchain and then deploys smart contracts (including smart contracts related to operations such as writing and querying smart device certificates) into the blockchain. Note that these ESs have been registered and are legitimate.

2) Authorization Seeds Generation: IIoT system has multiple services. For each service, KGC generates the corresponding authorization seeds and hides them.

1) Assume that the service is $Serv_s \in \{0,1\}^*$, the KGC chooses two number $\omega_s, \pi_s \in \mathbb{Z}_q^*$ and then sets ω_s as authorization seed of $Serv_s$.
2) KGC computes $C_s = \pi_s \cdot P + \omega_s \cdot P_{pub}$. Then KGC keeps ω_s and π_s secretly.
3) KGC computes the corresponding assisted secret key $as_{Serv_s} = H_1(Serv_s, msk)$, which is used to assist ES in verifying whether the authorization credential of the SD is legitimate.

Remark 1: In the proposed scheme, the KGC is fully trusted and can not be compromised. The msk is secret, no IIoT entities other than KGC can find a number $msk' \in \mathbb{Z}_q^*$ and computes $P_{pub} = msk' \cdot P$. Therefore, for other entities, the C_s reveals no information about ω_s.

3) Registration for IIoT Entities: To protect the privacy of ES and SD, the registration process of ES and SD is done through a secure and private channel.

1) ES_j submits a registration request $\{ID_j, Serv_s\}$ to KGC, where $ID_j \in \{0,1\}^*$ indicates the real identity of the ES_j and $Serv_s$ indicates the service that ES_j interested in. According to the request, the KGC computes $lsk_j = H_1(ID_j||msk)$ and $LPK_j = lsk_j \cdot P$, where lsk_j and LPK_j indicate the long secret key and long public key of ES_j respectively. Then KGC stores $\{lsk_j, LPK_j, as_{Serv_s}, C_s\}$ into the ES. Finally, KGC

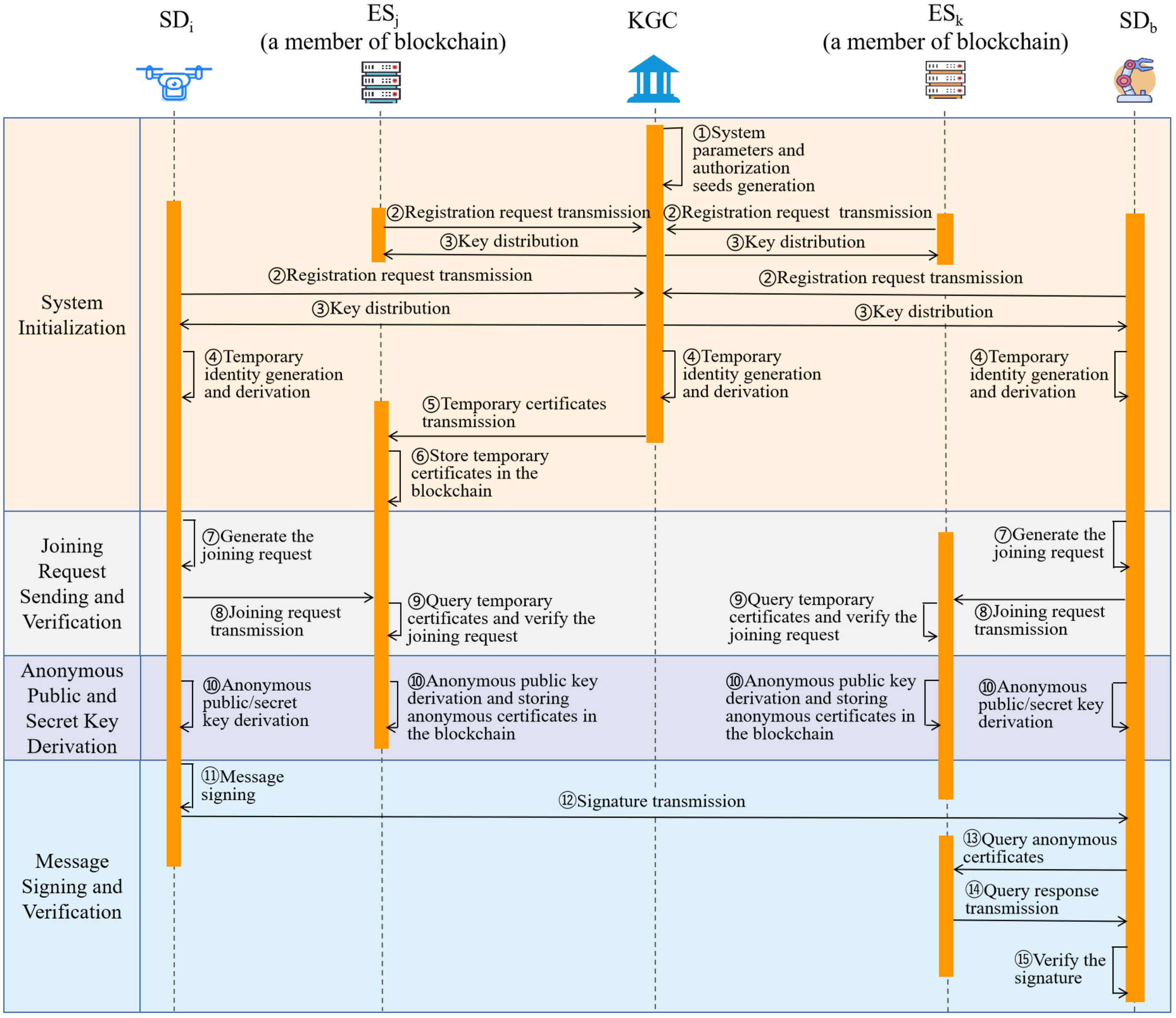

SD_i
ES_j (a member of blockchain)
KGC
ES_k (a member of blockchain)
SD_b
System Initialization
Joining Request Sending and Verification
Anonymous Public and Secret Key Derivation
Message Signing and Verification
①System parameters and authorization seeds generation
②Registration request transmission
③Key distribution
④Temporary identity generation and derivation
⑤Temporary certificates transmission
⑥Store temporary certificates in the blockchain
⑦Generate the joining request
⑧Joining request transmission
⑨Query temporary certificates and verify the joining request
⑩Anonymous public key derivation and storing anonymous certificates in the blockchain
⑩Anonymous public/secret key derivation
⑪Message signing
⑫Signature transmission
⑬Query anonymous certificates
⑭Query response transmission
⑮Verify the signature

Fig. 3. Authentication process overview.

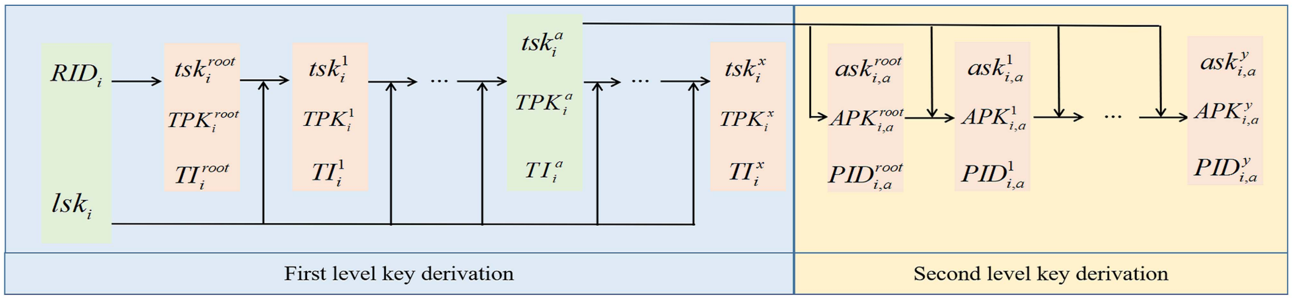

RID_i
lsk_i
tsk_i^root TPK_i^root TI_i^root
tsk_i^1 TPK_i^1 TI_i^1
tsk_i^a TPK_i^a TI_i^a
tsk_i^x TPK_i^x TI_i^x
ask_{i,a}^root APK_{i,a}^root PID_{i,a}^root
ask_{i,a}^1 APK_{i,a}^1 PID_{i,a}^1
ask_{i,a}^y APK_{i,a}^y PID_{i,a}^y
First level key derivation
Second level key derivation

Fig. 4. Two-level key derivation.

assists ES_j to join the blockchain according to the deployment strategy of the blockchain.

2) SD_i submits a registration request $\{RID_i, Serv_s\}$ to KGC, where $RID_i \in \{0,1\}^*$ indicates the real identity of SD_i and $Serv_s$ indicates the service that SD_i interested in. According to the request, the KGC computes $lsk_i = H_1(RID_i||msk)$, where lsk_i indicates the long secret key of SD_i. For $Serv_s$, KGC first chooses a current timestamps t_k, then calculates $n_i = H_1(RID_i||Serv_s||t_k||msk)$, $an_i = H_1(n_i||as_{Serv_s})$, $\omega_i = \omega_s \cdot an_i$, and $\pi_i = \pi_s \cdot an_i$, where ω_i is a temporary authorization credentials corresponding to $Serv_s$ for SD_i. Finally, the KGC stores $\{lsk_i, n_i, \omega_i, \pi_i\}$ into SD_i.

Remark 2: To further protect SD's privacy, for a service, SD's authorization credential is one-time, and once used, SD discards the authorization credential. Therefore, SD needs to periodically obtain new authorization credentials from KGC.

Remark 3: In the proposed scheme, for the same service, although the authorization seed is the same, different SD have different assisted parameters an_i, resulting in the service authorization credentials are not the same for different SDs supporting the same service. Therefore, revoking an SD does not affect the use of service authorization credentials for other SDs.

4) Temporary Identity Generation and Derivation for SD: To ensure the anonymity of SD_i, when it wants to join a new industrial department, it needs to use a new temporary identity to hide the real identity RID_i.

The temporary identity generation and derivation are as follows.

1) SD_i computes the root private derivation information $pdi_i^{root} = H_1(lsk_i \cdot P_{pub})$, root temporary secret key $tsk_i^{root} = H_1(lsk_i || (lsk_i \cdot P_{pub}))$, root temporary public key $TPK_i^{root} = tsk_i^{root} \cdot P$, and root temporary identity $TI_i^{root} = RID_i \oplus H_2(tsk_i^{root} \cdot P_{pub})$. If SD_i wants to obtain $x-th$ temporary secret key tsk_i^x and corresponding parameters, it computes $h_i^x = H_1(pdi_i^{x-1} || TPK_i^{x-1})$, $pdi_i^x = H_1(pdi_i^{x-1} || (lsk_i \cdot P_{pub}) || TPK_i^{x-1})$, $tsk_i^x = tsk_i^{x-1} \cdot h_i^x$, $TPK_i^x = tsk_i^x \cdot P$, and $TI_i^x = RID_i \oplus H_2(tsk_i^x \cdot P_{pub})$. Note that when $x = 1$, then $pdi_i^{x-1} = pdi_i^{root}$, $tsk_i^{x-1} = tsk_i^{root}$, and $TPK_i^{x-1} = TPK_i^{root}$.
2) For KGC, it has all the parameters of SD_i so that it can generate the corresponding parameters $(tsk_i^x, pdi_i^x, TPK_i^x, TI_i^x)$ in the same way as SD_i. Then, KGC calculates temporary certificate $tc_i^x = H_1(TI_i^x || TPK_i^x)$. Finally, KGC marks the temporary certificate as valid and stores it in the blockchain.

Remark 4: During SD's registration, it can negotiate with KGC on the frequency of updating temporary certificates. Subsequently, KGC can regularly generate multiple temporary certificates and store them on the blockchain. Therefore, during the generation of temporary certificates, SD and KGC do not need to communicate. On the one hand, in traditional anonymity schemes, SDs need to request and store a large number of temporary information from the KGC in advance. However, in our proposed scheme, SDs can derive their own temporary identities and corresponding parameters, thus saving storage space and reducing storage pressure. On the other hand, the KGC calculates the temporary certificates of the SD and stores them in the blockchain. Only the temporary certificates stored in the blockchain are legitimate, thus enabling effective supervision of SDs and preventing them from performing key derivation under unrestricted conditions.

C. Joining Request Sending and Verification

In this phase, SD_i wants to join a new industrial department, it needs to send a joining request to the corresponding ES (e.g., ES_j). After receiving the request, ES_j verifies whether the temporary identity (e.g., TI_i^a) of the SD_i is legitimate and whether there is authorization with the corresponding service (e.g., $Serv_s$).

1) Send Joining Request: When SD_i wants to send a join request, it will perform the following steps.

1) SD_i first selects three random number $r_i \in \mathbb{Z}_q^*$, $\varrho_i^1 \in \mathbb{Z}_q^*$, and $\varrho_i^2 \in \mathbb{Z}_q^*$. Then SD_i calculates $e = H_2(r_i \cdot LPK_j)$, $C_i = \pi_i \cdot P + \omega_i \cdot P_{pub}$, $\widetilde{C}_i = \varrho_i^1 \cdot P + \varrho_i^2 \cdot P_{pub}$ and $\alpha_i = H_1(t_i^1 || e || C_i || \widetilde{C}_i)$, where t_i^1 indicates current timestamp. Subsequently, SD_i computes $\varphi_i^1 = \varrho_i^1 + \pi_i \cdot \alpha_i$ and $\varphi_i^2 = \varrho_i^2 + \omega_i \cdot \alpha_i$.
2) SD_i first calculates $R_i = r_i \cdot P$, $\beta_i = H_1(t_i^1 || c || R_i || TPK_i^a)$, where $c = Enc_e(\varphi_i^1 || \varphi_i^2 || C_i || \widetilde{C}_i || n_i || TI_i^a || TPK_i^a || Serv_s)$. Then SD_i computes $\sigma_{req} = r_i + tsk_i^a \cdot \beta_i$, where σ_{req} indicates the join request signature.
3) SD_i sends $M_i^1 = \{\sigma_{req}, t_i^1, c, R_i\}$ to ES_j.

2) Verify Request: Once ES_j receives the M_i^1, it performs the following steps.

1) ES_j checks the freshness of timestamp t_i^1, if it has not been expired, ES_j computes $e' = H_2(lsk_j \cdot R_i)$, $(\varphi_i^1 || \varphi_i^2 || C_i || \widetilde{C}_i || n_i || TI_i^a || TPK_i^a || Serv_s) = Dec_{e'}(c)$, and $tc_i^{a'} = H_1(TI_i^a || TPK_i^a)$. Then ES_j queries $tc_i^{a'}$ in the blockchain, and if it does not exist, ES_j discards the message directly. Otherwise, ES_j performs the next step.
2) ES_j calculates $\beta_i' = H_1(t_i^1 || c || R_i || TPK_i^a)$ and verifies whether the (1) is holds. If not, ES_j discards the message directly; otherwise, ES_j performs the next step.
$$\sigma_{req} \cdot P = R_i + \beta_i' \cdot TPK_i^a. \tag{1}$$
3) ES_j computes $an_i' = H_1(n_i || as_{serv_s})$ and $C_s' = C_i / an_i'$. Then ES_j queries the local database for whether $\{C_s', Serv_s\}$ is existed. If not, ES_j drops this message. Otherwise, ES_j computes $\alpha_i' = H_1(t_i^1 || e' || C_i || \widetilde{C}_i)$ and verifies whether the (2) is holds. If not, ES_j discards the message directly. Otherwise, SD successfully joins the department and can participate in industrial services $Serv_s$ in the long term.
$$\varphi_i^1 \cdot P + \varphi_i^2 \cdot P_{pub} = \widetilde{C}_i + \alpha_i' \cdot C_i \tag{2}$$

Remark 5: An SD can participate in multiple services in multiple departments. For example, a drone can participate in production monitoring services in multiple industrial departments.

D. Anonymous Public and Secret Key Derivation

In this phase, the SD_i and ES_j will generate private derivation information $pdi_{i,a}^{root}$, and then derive the corresponding public and private keys according to $pdi_{i,a}^{root}$.

1) Generate Root Private Derivation Information: Before key derivation, ES_j and SD_i need to generate the root key derivation information and the corresponding root anonymous public/private key.

1) ES_j calculates $temp_{i,a} = H_1(n_i || H_2(lsk_j \cdot TPK_i^a))$ and the root private derivation information $pdi_{i,a}^{root} = H_1(temp_{i,a} || TPK_i^a)$.

2) SD_i calculates $temp_{i,a} = H_1(n_i||H_2(tsk_i^a \cdot LPK_j))$ and uses the same way as ES_j to generate $pdi_{i,a}^{root}$.
3) ES_j computes the root anonymous public key $APK_{i,a}^{root} = temp_{i,a} \cdot TPK_i^a$.
4) SD_i calculates the root anonymous secret key $ask_{i,a}^{root} = tsk_i^a \cdot temp_{i,a}$, $APK_{i,a}^{root} = ask_{i,a}^{root} \cdot P$, $PID_{i,a}^{root} = TI_i^a \oplus H_2(ask_{i,a}^{root} \cdot LPK_j)$.

2) Anonymous Public Key Derivation: According to the SD_i and corresponding service, ES_j derives the $y-th$ anonymous public key for SD_i.

1) ES_j computes $\mu_{i,a}^y = H_1(pdi_{i,a}^{y-1}||APK_{i,a}^{y-1})$, $pdi_{i,a}^y = H_1(pdi_{i,a}^{y-1}||(lsk_j \cdot TPK_i^a)||APK_{i,a}^{y-1})$, and the $y-th$ anonymous public key $APK_{i,a}^y = \mu_{i,a}^y \cdot APK_{i,a}^{y-1}$. Note that when $y = 1$, then $pdi_{i,a}^{y-1} = pdi_{i,a}^{root}$, $APK_{i,a}^{y-1} = APK_{i,a}^{root}$.
2) ES_j computes the $y-th$ pseudonym $PID_{i,a}^y = TI_i^a \oplus H_2(lsk_j \cdot APK_{i,a}^y)$ and corresponding anonymous certificate $ac_{i,a}^y = H_1(PID_{i,a}^y||Serv_s||APK_{i,a}^y||LPK_j)$.
3) ES_j marks the anonymous certificate as valid and stores it in the blockchain.

3) Anonymous Public and Secret Key Derivation: According to the ES_j and corresponding service, SD_i derives the $y-th$ anonymous secret key. Noting that the anonymous secret key of SD can only be derived by itself.

1) SD_i calculates $\mu_{i,a}^y = H_1(pdi_{i,a}^{y-1}||APK_{i,a}^{y-1})$, $pdi_{i,a}^y = H_1(pdi_{i,a}^{y-1}||(tsk_i^a \cdot LPK_j)||APK_{i,a}^{y-1})$, and the $y-th$ anonymous secret key $ask_{i,a}^y = \mu_{i,a}^y \cdot ask_{i,a}^{y-1}$. When $y = 1$, then $ask_{i,a}^{y-1} = ask_{i,a}^{root}$.
2) SD_i computes the $y-th$ pseudonym $PID_{i,a}^y = TI_i^a \oplus H_2(ask_{i,a}^y \cdot LPK_j)$.
3) SD_i computes the $y-th$ anonymous public key $APK_{i,a}^y = ask_{i,a}^y \cdot P$.

Remark 6: Similar to the temporary certificate generation, ES can periodically generate multiple anonymous certificates based on the characteristics of the industrial service to which SD subscribes, and then ES stores these anonymous certificates in the blockchain.

E. Message Signing and Verification

To ensure the data source is trusted and the data integrity, the sender (an SD) must sign the data before sending the message. After receiving the message, the receiver (another SD) needs to authenticate the message.

1) Message Signing: Assume that the message sender is SD_i. When data d_i is generated, SD_i will proceed with the following steps.

1) SD_i first chooses current timestamp t_i^2 and a random number $v_i \in \mathbb{Z}_q^*$. Then, SD_i computes $V_i = v_i \cdot P$ and $f_i = H_1(d_i||t_i^2||PID_{i,a}^y||APK_{i,a}^y||V_i||LPK_j)$.
2) SD_i calculates signature $\sigma_i = v_i + ask_{i,a}^y \cdot f_i$. Subsequently, SD_i broadcasts $M_i^2 = \{\sigma_i, d_i, t_i^2, Serv_s, PID_{i,a}^y, APK_{i,a}^y, V_i, LPK_j\}$ to IIoT system.

Remark 7: During the message signing process, the generation of some data (e.g., v_i, V_i) does not depend on the online message d_i. Therefore, when the computational density of smart devices is low, these data can be generated in advance, thereby improving the efficiency of the smart devices' signing process.

2) Message Verification: Assume that the message receiver is SD_b. Upon it receives the message, it will proceed the following steps.

1) SD_b checks the freshness of timestamp t_i^2, if it has been expired, SD_b discards the message directly. Otherwise, SD_b performs the next step.
2) SD_b computes $ac_{i,a}^{y'} = H_1(PID_{i,a}^y||Serv_s||APK_{i,a}^y||LPK_j)$ and then queries $ac_i^{y'}$ in the BC. If it does not exist, the SD_b drops this message; otherwise, SD_b performs the next step.
3) SD_b calculates $f_i' = H_1(d_i||t_i^2||PID_{i,a}^y||APK_{i,a}^y||V_i||LPK_j)$ and verifies whether the (3) is holds.

$$\sigma_i \cdot P = V_i + f_i' \cdot APK_{i,a}^y. \tag{3}$$

Batch Verification: Note that the proposed scheme supports batch verification. That is, a message receiver can simultaneously verify multiple messages. First, SD_b checks the timestamp in the received message; if the timestamp is expired, the SD_b directly discards the corresponding message. Suppose there are n timestamps still fresh, and the corresponding messages are $\{M_1^2, \ldots, M_i^2, \ldots, M_n^2\}$. To prevent non-repudiation attacks, the proposed scheme uses the small exponent test technology. SD_b randomly selects a vector $u = \{u_1, \ldots, u_i, \ldots, u_n\}$, where $u_i \in [1, 2^l]$, and l is a small random integer. Finally, SD_b calculates $f_i' = H_1(d_i||t_i^2||PID_{i,a}^y||APK_{i,a}^y||V_i||LPK_j)$ and whether the following (4) is holds. If it holds, the SD_b accepts these valid messages.

$$\left(\sum_{i=1}^{n}(u_i \cdot \sigma_i)\right) \cdot P = \sum_{i=1}^{n}(u_i \cdot V_i) + \sum_{i=1}^{n}(u_i \cdot f_i' \cdot APK_{i,a}^y). \tag{4}$$

F. Tracing and Revocation

In the proposed scheme, the tracing and revocation can be divided into three types according to different needs: (1) anonymous certificate tracing and revocation; (2) temporary certificate tracing and revocation; and (3) real identity tracing and revocation.

1) Anonymous Certificate Tracing and Revocation: When SD_i expires for a service (e.g., $Serv_s$) in the $j-th$ industry department, the specific revocation process is as follows.

1) SD_i actively sends TI_i^a to ES_j, or ES_j obtains the corresponding temporary identity by computing $TI_i^a = PID_{i,a}^y \oplus H_2(lsk_j \cdot APK_{i,a}^y)$.
2) Based on TI_i^a and $Serv_s$, ES_j stops generating the corresponding pseudonyms. Then ES_j packages the corresponding anonymous certificates of these pseudonyms into $< ac_{i,a}^* >$. Finally, ES_j sends revocation request with $< ac_{i,a}^* >$ to BC.
3) BC marks the $< ac_{i,a}^* >$ as invalid according to the revocation request.

2) Temporary Certificate Tracing and Revocation: When an SD (e.g., SD_i) wants to leave the $j-th$ industry department, the specific revocation process is as follows.

1) According to the temporary identity (e.g., TI_i^a) of the SD_i, the ES_j collaborates with BC to revoke corresponding anonymous certificates through anonymous certificate tracing and revocation.
2) ES_j sends revocation request with temporary certificate tc_i^a to BC. Then the BC marks the tc_i^a as invalid.

3) Real Identity Tracing and Revocation: Once SD_i is found to send an illegitimate message, the KGC, BC and the corresponding ES will collaborate to revoke the SD_i. The specific revocation process is as follows.

1) According to LPK_j, the receiver SD_b sends $\{PID_{i,a}^y, APK_{i,a}^y\}$ to the corresponding ES_j.
2) ES_j calculates $TI_i^a = PID_{i,a}^y \oplus H_2(lsk_j \cdot APK_{i,a}^y)$, and then sends $\{TI_i^a, TPK_i^a\}$ to KGC.
3) KGC first computes $RID_i = TI_i^a \oplus H_2(msk \cdot TPK_i^a)$, and then sends revocation request to BC with $< tc_i^* >$, where $< tc_i^* >$ is the temporary certificate set corresponding to RID_i. Next, KGC broadcasts $< TI_i^* >$ to the IIoT system, where $< TI_i^* >$ is the temporary identity set corresponding to RID_i.
4) BC marks the $< tc_i^* >$ as invalid according to the revocation request.
5) According to $< TI_i^* >$, the corresponding ES performs the anonymous certificate tracing and revocation.

Remark 8: To improve efficiency, our proposed scheme supports batch writes and batch reads in the blockchain. For example, in the message verification phase, when SD_b receives multiple messages, it can batch retrieve the anonymous certificates corresponding to these messages in the blockchain.

VI. Security Proof and Analysis

This section shows that the signatures satisfy existential unforgeability in the random oracle model using security proof. Subsequently, this section demonstrates that the proposed scheme meets security objectives and can withstand various attacks through security analysis. Finally, we compare the proposed scheme with other related schemes regarding security and functionality.

A. Security Proof

In the proposed scheme, we use two signatures, σ_{req} and σ_i, respectively. Since these two signatures are generated in the same way, we only prove that σ_i is unforgeable against the $\mathcal{A}$ based on Definition 1.

Theorem 1: If ECDLP is (ς', ϵ')-hard in $\mathbb{G}$, the signature algorithm is (ς, ϵ)-existential unforgeable against adaptively chosen message attacks in random oracle model, such that $\epsilon' \geq \epsilon_1 \cdot \left(\frac{\epsilon_1}{q_{h_1}} + \frac{1}{q}\right)$ and $\varsigma' \leq 2\varsigma + q_s(2t_{sm} + t_{pa})$, where $\epsilon_1 = \epsilon - \frac{q_s(q_{h_1}+q_s)}{q}$, q_{h_1} indicates the max times of H_1 query and q_s indicates the max times of sign query. In addition, t_{sm} indicates the time taken for the scalar multiplication operation in $\mathbb{G}$ and t_{pa} indicates the time taken for the point addition in $\mathbb{G}$.

Proof: If $\mathcal{A}$ can break the proposed scheme with probability ϵ, then $\mathcal{C}$ can solve the ECDLP run by $\mathcal{A}$ as a subroutine with the probability ϵ'. Given a group $\mathbb{G}$ and an ECDLP instance $\{P, Q = \alpha P | \alpha \in \mathbb{Z}_q^*, P \in \mathbb{G}, Q \in \mathbb{G}\}$, $\mathcal{C}$ responses the oracle queried by $\mathcal{A}$ as follows.

Setup-query: For the query, $\mathcal{C}$ sets the public key $PK_i \leftarrow Q$. Subsequently, $\mathcal{C}$ sends the public parameters $\{E, \mathbb{G}, P, PK_i\}$ to $\mathcal{A}$.

H_1-query: For the query, $\mathcal{C}$ presets a map Map_{H_1}, and the Map_{H_1} is empty at the beginning. When $\mathcal{A}$ makes the query with $< d_i, t_i^2, PID_{i,a}^y, PK_i, V_i, LPK_j >$, $\mathcal{C}$ checks whether the Map_{H_1} has the key $< d_i || t_i^2 || PID_{i,a}^y || PK_i || V_i || LPK_j >$. If so, $\mathcal{C}$ finds the corresponding value and returns it to $\mathcal{A}$. Otherwise, $\mathcal{C}$ chooses a random number $r_{H_1} \in \mathbb{Z}_q^*$, where r_{H_1} should satisfy $r_{H_1} \notin Map_{H_1}$. Finally, the $\mathcal{C}$ sets the value $Map_{H_1}(< d_i || t_i^2 || PID_{i,a}^y || PK_i || V_i || LPK_j >) \leftarrow r_{H_1}$ and returns r_{H_1} to $\mathcal{A}$.

Sign-query: For the query, $\mathcal{C}$ presets a map Map_{sig}, and the Map_{sig} is empty at the beginning. When $\mathcal{A}$ makes the query with $< d_i, t_i^2, PID_{i,a}^y, PK_i, LPK_j >$, $\mathcal{C}$ first randomly chooses a number $\sigma_i \in \mathbb{Z}_q^*$, and computes $V_i = \sigma_i \cdot P - r_{H_1} \cdot PK_i$. If $Map_{H_1}(< d_i || t_i^2 || PID_{i,a}^y || PK_i || V_i || LPK_j >)$ has already been defined, then $\mathcal{C}$ aborts this process, then returns $\perp$ and sets $bad \leftarrow true$. Otherwise, $\mathcal{C}$ returns $< \sigma_i, V_i >$ on $< d_i, t_i^2, PID_{i,a}^y, PK_i, LPK_j >$ to $\mathcal{A}$.

After executes the above-mentioned queries, adversary $\mathcal{A}$ outputs a forged signature $< \sigma_i, V_i >$ on $< d_i, t_i^2, PID_{i,a}^y, PK_i, LPK_j >$. If $\mathcal{A}$ has not invokes a *Sign-query* with the message $< d_i, t_i^2, PID_{i,a}^y, PK_i, LPK_j >$, then the forgery is non-trivial.

Let E_1 indicate that $\mathcal{C}$ does not abort during the signature simulation, and E_2 indicate that $\mathcal{A}$ outputs a non-trivial forgery. Therefore, the probability of $\mathcal{A}$ forges a valid signature $< \sigma_i, V_i >$ is $\epsilon_1 = Pr[E_1]Pr[E_2|E_1]$.

Claim 1: $Pr[E_1] = Pr[\neg bad] \geq 1 - \frac{q_s(q_{h_1}+q_s)}{q}$

Proof: If the tuple $< d_i, t_i^2, PID_{i,a}^y, PK_i, LPK_j >$ has already occurred during a previous H_1 queries, then $bad \leftarrow true$. There are at most $q_{h_1} + q_s$ entries, so for one *Setup-query*, the probability of E_1 is at most $\frac{(q_{h_1}+q_s)}{q}$. Therefore, for q_s queries, the probability of E_1 is at most $\frac{q_s(q_{h_1}+q_s)}{q}$.

Claim 2: $Pr[E_2|E_1] \geq \epsilon$

Proof: $Pr[E_2|E_1]$ indicates that the probability in the previous sign queries, $\mathcal{C}$ does not abort and $\mathcal{A}$ forges a signature successfully. Therefore, $\mathcal{A}$ will generates a non-trivial forgery with probability no less than ϵ.

In summary, the probability that $\mathcal{A}$ outputs a non-trivial forgery no less than $\epsilon_1 = \epsilon - \frac{q_s(q_{h_1}+q_s)}{q}$. For the same inputs on $< d_i, t_i^2, PID_{i,a}^y, PK_i, LPK_j >$, the $\mathcal{C}$ executes Forking algorithm [34] to obtain σ_i and σ_i^* with different outputs of H_1-*query*, which as presented in (5).

$$\begin{aligned} V_i &= \sigma_i \cdot P - r_{H_1} \cdot PK_i, \\ V_i &= \sigma_i^* \cdot P - r_{H_1}^* \cdot PK_i, \end{aligned} \tag{5}$$

where $\sigma_i \neq \sigma_i^*$ and $r_{H_1} \neq r_{H_1}^*$. According to (5), we could get

$$\alpha = \frac{\sigma_i - \sigma_i^*}{r_{H_1} - r_{H_1}^*}. \tag{6}$$

Based on the General Forking Lemma of Bellare and Neven [34], the success probability of $\mathcal{C}$ as shown in the (7).

$$\epsilon' \geq \left(\epsilon - \frac{q_s(q_{h_1}+q_s)}{q}\right) \cdot \left(\frac{\epsilon - \frac{q_s(q_{h_1}+q_s)}{q}}{q_{h_1}} - \frac{1}{q}\right). \quad (7)$$

To solve the ECDLP, assume that the operation time of the scalar multiplication in $\mathbb{G}$ is t_{sm}, the operation time of the point addition in $\mathbb{G}$ is t_{pa}, and other operation time is negligible. The running time of $\mathcal{C}$ is twice that of $\mathcal{A}$, plus the time needed to answer the hash and signature queries. Therefore, the time $\mathcal{C}$ runs is $\varsigma' \leq 2\varsigma + q_s(2t_{sm} + t_{ap})$.

B. Security Analysis

In the proposed scheme, SD_i has two signatures, σ_{req} and σ_i, respectively. Since these two signatures are generated in the same way and to reduce the redundant description, we only analyze the message signature σ_i in this subsection.

- **Message integrity and authentication:** On the one hand, the ECDLP is difficult to solve, so no adversary can forgery a valid signature within a given polynomial time. On the other hand, if the data and the corresponding signature satisfy $\sigma_i \cdot P = V_i + f_i' \cdot APK_{i,a}^y$, then the signature is regarded as valid. Therefore, the proposed scheme can satisfy message integrity and authentication.
- **Message anonymity:** In the authentication process, the smart device SD_i's real identity is hidden in the temporary identity TI_i^x, where $TI_i^x = RID_i \oplus H_2(tsk_i^x \cdot P_{pub})$. To obtain the RID_i, the adversary should calculate $tsk_i^x \cdot P_{pub} = tsk_i^x \cdot msk \cdot P$ from $TPK_i^x = tsk_i^x \cdot P$ and $P_{pub} = msk \cdot P$. Since to the hardness of the ECDHP, the smart device's real identity is not available to entities other than the temporary identity owner and KGC.
- **Unlinkability:** SD_i generates a message signature σ_i needs a pseudonym $PID_{i,a}^y$, which is dynamically changing and not repeating. Therefore, no adversary can link two different signatures.
- **Identity traceability:** According to Section V-F, we find that ES can trace the temporary identity of SD by computing $TI_i^a = PID_{i,a}^y \oplus H_2(lsk_j \cdot APK_{i,a}^y)$. In addition, KGC can trace the real identity of SD by calculating $RID_i = TI_i^a \oplus H_2(msk \cdot TPK_i^a)$.
- **Resistance to replay attacks:** The message M_i^2 contains timestamps, and when the verifier receives the message, it immediately checks the freshness of the timestamp and discards the replayed message.
- **Resistance to impersonation attacks:** The smart device SD_i signs data d_i by calculating $\sigma_i = v_i + ask_{i,a}^y \cdot f_i$, where $f_i = H_1(d_i||t_i^2||PID_{i,a}^y||APK_{i,a}^y||V_i||LPK_j)$. Because the one-way hash function H_1 is collision-resistant and the adversary cannot obtain the anonymous secret key $ask_{i,a}^y$, the adversary cannot generate a valid message.

TABLE II
COMPARISON OF SECURITY AND FUNCTIONALITY FEATURES

	[30]	[27]	[29]	[31]	Ours
Message integrity and authentication	✓	✓	✓	✓	✓
Message anonymous	✓	✓	×	✓	✓
Unlinkability	✓	✓	×	✓	✓
Identity traceability	✓	✓	✓	✓	✓
Resist replay attacks	✓	✓	×	×	✓
Resist modification attacks	✓	✓	✓	✓	✓
Resist impersonation attacks	✓	✓	✓	✓	✓
Key derivation	×	✓	×	×	✓
Batch authentication	×	✓	×	✓	✓
Flexible revocation	×	×	×	×	✓

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

- **Resistance to modification attacks:** Once the message M_i^2 is received, the receiver determines whether the equation $\sigma_i \cdot P = V_i + f_i' \cdot APK_{i,a}^y$ holds. If it is not holds, the message will be discarded immediately. Since the modified message cannot pass authentication, this scheme can resist modification attacks.

C. Comparison of Security and Functionality

According to Table II, when compared with related schemes [27], [29], [30], and [31], we find that the proposed scheme achieves more merits.

VII. PERFORMANCE EVALUATION

This section presents five aspects of the experimental setting, computational overhead, storage overhead, communication overhead, and on-chain evaluation. Note that to evaluate the efficiency of authentication, we compare our proposed scheme with four related authentication schemes [27], [29], [30], and [31]. To make the comparison more fair, we introduce these schemes into our proposed system model.

A. Experimental Setting

In this subsection, we use C++ code to implement the relevant schemes [27], [29], [30], [31], and ours. The cryptographic tool library we use is Miracl Core [35], and the curve type we choose is BLS12381, which is a pairing-friendly elliptic curve and can provide a 128-bit security level. The basic equation for BLS12381 is $y^2 = x^3 + 4$.

The operations in SD are implemented on a Raspberry Pi 4 with a 1.5 GHz CPU and 4 GB memory, and running Debian GNU/Linux 11 operating system. To evaluate the performance of on-chain operations, we use Go code to implement the corresponding operations on-chain. We build a hyperledger fabric network [36] using ten PCs. The configuration of each PC is running Ubuntu 18.04.3 system, equipped with Intel Core i7-11700 CPU @2.50 GHz and 16 GB of memory. Among these

TABLE III
TIME-CONSUMING OPERATION COMPARISON

Item	Zhang *et al.* [30]	Lin *et al.* [27]	Karati *et al.* [29]	Li *et al.* [31]	Ours
Message Signing	$6GE_1 + 2GE_2 + 4BP{+}4GE_T{+}1HTP$	$3SM_1$	$2GE_2 + 1GE_T$	$2SM_1 + 2SM_2 + 1BP + 1GE_T + 2HTP$	$1SM_1$
One Message Verification	$4GE_1 + 2GE_2 + 6BP + 5GE_T$	$3SM_1$	$2GE_T + 1BP$	$3BP + 2HTP$	$2SM_1$
n-meassage Verification	$4nGE_1 + 2nGE_2 + 6nBP + 5nGE_T$	$(n + 2)SM_1$	$2nGE_T + nBP$	$(2n + 1)BP + 2nHTP + (2n + 1)SM_2$	$(n + 1)SM_1$

ten PCs, one is a client node, one is an order node, and the rest are peer nodes.

B. Computational Overhead

We evaluate the computational overhead of authentication from theoretical and simulated experimental results.

1) Theoretical Analysis: For convenience, we define the time-consuming cryptographic operations as follows.

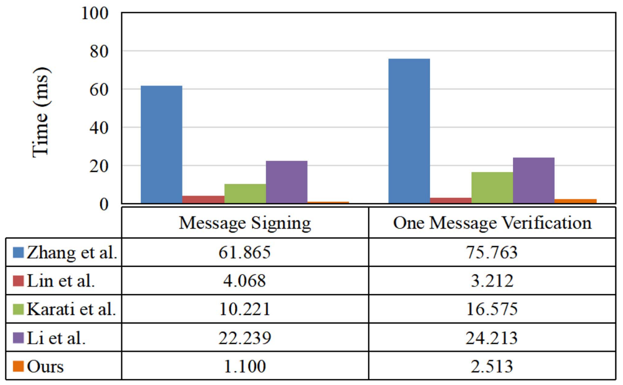
- BP: a bilinear pairing operation $e(g_1, g_2)$, where $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$.
- SM_1: a scale multiplication operation $a \cdot P_1$, where $a \in \mathbb{Z}_q^*$ and $P_1 \in \mathbb{G}_1$.
- SM_2: a scale multiplication operation $a \cdot P_2$, where $a \in \mathbb{Z}_q^*$ and $P_2 \in \mathbb{G}_2$.
- GE_1: an exponential operation P_1^r, where $r \in \mathbb{Z}_q^*$ and $P_1 \in \mathbb{G}_1$.
- GE_2 : an exponential operation P_2^r, where $r \in \mathbb{Z}_q^*$ and $P_2 \in \mathbb{G}_2$.
- GE_T: an exponential operation P_3^r, where $r \in \mathbb{Z}_q^*$ and $P_3 \in \mathbb{G}_T$.
- HTP: a hash-to-point operation $H(m) \in \mathbb{G}_1$, where $H(\cdot)$ is a secure hash function and $m \in \{0, 1\}^*$.

The results of the theoretical analysis are shown in Table III.

For the message signing phase, we find that more time-consuming cryptographic operations are used in schemes [29], [30], and [31], which are $6GE_1 + 2GE_2 + 4BP + 4GE_T + 1HTP$, $2GE_2 + 1GE_T$, and $2SM_1 + 2SM_2 + 1BP + 1GE_T + 2HTP$, respectively. And the fewer time-consuming cryptographic operations are used in scheme [27] and our proposed scheme, which are $3SM_1$ and $1SM_1$ respectively.

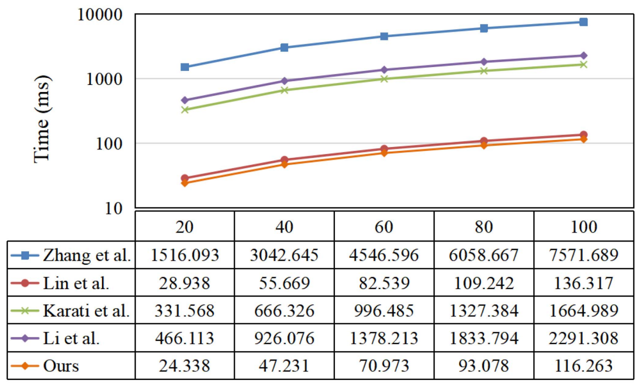
For the message verification phase, we can see that more time-consuming cryptographic operations are used in schemes [29], [30], and [31]. When the number of messages to be verified is 1, the time-consuming cryptographic operations required by scheme [30], scheme [27], scheme [29], scheme [31], and our proposed scheme are $4GE_1 + 2GE_2 + 6BP + 5GE_T$, $3SM_1$, $2GE_T + 1BP$, $3BP + 2HTP$, and $2SM_1$, respectively.

Scheme [30] and scheme [29] do not support batch verification. Therefore, when the number of messages to be verified is n, the time-consuming cryptographic operations required for these schemes are n times that required for verifying a single message. Specifically, the time-consuming cryptographic operations required are $n(4GE_1 + 2GE_2 + 6BP + 5GE_T)$ for scheme [30] and $n(2GE_T + BP)$ for scheme [29]. Conversely,



	Message Signing	One Message Verification
🟦 Zhang et al.	61.865	75.763
🟥 Lin et al.	4.068	3.212
🟩 Karati et al.	10.221	16.575
🟪 Li et al.	22.239	24.213
🟧 Ours	1.100	2.513

Fig. 5. Comparison result of different phases.



	20	40	60	80	100
■ Zhang et al.	1516.093	3042.645	4546.596	6058.667	7571.689
● Lin et al.	28.938	55.669	82.539	109.242	136.317
× Karati et al.	331.568	666.326	996.485	1327.384	1664.989
◆ Li et al.	466.113	926.076	1378.213	1833.794	2291.308
◆ Ours	24.338	47.231	70.973	93.078	116.263

Fig. 6. Comparison of n-message verification efficiency.

scheme [27], scheme [31], and our proposed scheme support batch verification. Therefore, the time-consuming cryptographic operations required to verify n messages are less than n times those required for verifying a single message. Specifically, the operations are $(n + 2)SM_1$ for scheme [27], $(2n + 1)BP + 2nHTP + (2n + 1)SM_2$ for scheme [31], and $(n + 1)SM_1$ for our proposed scheme.

2) Simulation Experimental Results: To show more visually that our proposed scheme is lightweight, we implement three other related schemes and our proposed scheme according to the experimental setup. The experimental results are shown in Figs. 5 and 6.

From Fig. 5, we can observe that our proposed scheme has the lowest computational overhead for signing a single message, requiring only 1.100 ms. This performance results from the efficient design of our proposed scheme where the time-consuming cryptographic operation required for signing a message is only $1SM_1$. The schemes with higher computational overhead are scheme [30], scheme [29], and scheme [31], with the time required to sign a single message being approximately 61.865 ms, 10.221 ms, and 22.239 ms respectively. This is because in these schemes, signing a message involves multiple of the most time-consuming bilinear pairing operations. Through calculation, it is evident that the computational overhead advantage of our designed scheme is significant when signing a single message. Specifically, the computational overhead of our designed scheme is $1.100/61.865 \approx 1.78\%$ of scheme [30], $1.100/4.068 \approx 27.04\%$ of scheme [27], $1.100/10.221 \approx 10.76\%$ of scheme [29], and $1.100/22.239 \approx 4.95\%$ of scheme [31]. Similarly, we can observe that our proposed scheme has the lowest computational overhead for verifying a single message, requiring only 2.513 ms. This is because the time-consuming cryptographic operations required to verify a single message in our proposed scheme are only $2SM_1$. Specifically, when verifying a single message, the computational overhead of our designed scheme is $2.513/75.763 \approx 3.32\%$ of scheme [30], $2.513/3.212 \approx 78.24\%$ of scheme [27], $2.513/16.575 \approx 15.16\%$ of scheme [29], and $2.513/24.213 \approx 10.38\%$ of scheme [31]. This reason is that schemes [29], [30], and [31] involve many time-consuming pairing operations. Although [27] does not have time-consuming pairing operations, it requires more scalar multiplications than our proposed scheme.

As can be seen in Fig. 6, the time consumed for verification in our proposed scheme is always at the lowest level as the number of messages to be verified increases. The reason is that when verifying a single message, our designed scheme has the minimal computational overhead. As the number of messages to verify increases, the growth rate of computational overhead in our designed scheme remains low. This is because our scheme supports batch verification, and for each additional message, only one additional SM_1 cryptographic operation is required. Although scheme [27] requires only one additional time-consuming cryptographic operation SM_1 for each message added, it still incurs greater computational overhead than our designed scheme during batch verification because its initial computational overhead are higher than those of our designed scheme. In addition, schemes [29], [30], and [31] require an increasing number of the most time-consuming pairing operations as the number of messages increases. Therefore, these three schemes have higher computational overheads. When the number of messages is 100, the computational overhead of our proposed scheme is approximately 116.263 ms, which is about $116.263/7571.689 \approx 1.54\%$ of scheme [30], about $116.263/136.317 \approx 85.29\%$ of scheme [27], about $116.263/1664.989 \approx 6.98\%$ of scheme [29], and about $116.263/2291.308 \approx 5.07\%$ of scheme [31].

C. Communication Overhead

In the cryptographic library we use, the elements in $\mathbb{Z}_q^*$ are 48 bytes, the elements in $\mathbb{G}_1$ are 97 bytes, the elements in $\mathbb{G}_2$ are 193 bytes, and the elements in $\mathbb{G}_T$ are 576 bytes. We set the length of the data to 20 bytes, the length of the service to 20 bytes, and the length of the timestamp to 16 bytes.

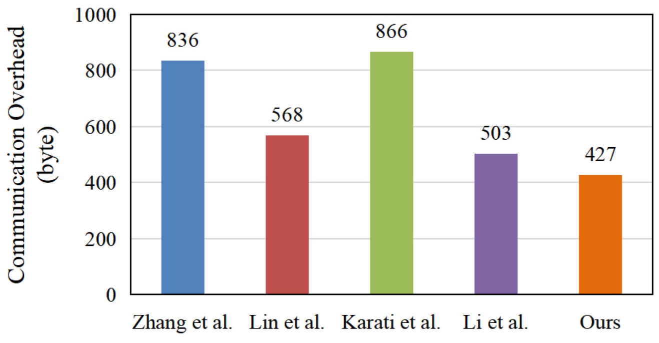


Fig. 7. Comparison of communication overhead.

In the proposed scheme, the SD_i sends the message $M_i^2 = \{\sigma_i, d_i, t_i^2, Serv_s, PID_{i,a}^y, APK_{i,a}^y, V_i, LPK_j\}$ to the corresponding message receiver, where the size of this message is $(48 + 20 + 16 + 20 + 32 + 97 + 97 + 97) = 427$ bytes. We utilize the same method to calculate the communication overhead of the other four related schemes. Through Fig. 7, we can see the communication overhead of scheme [30], scheme [27], scheme [29] and scheme [31] are 836 bytes, 568 bytes, 866 bytes, and 503 bytes respectively. Specifically, we find that the communication overhead of our proposed scheme is the lowest, about $427/836 \approx 51.08\%$ of [30], about $427/568 \approx 75.18\%$ of [27], about $427/866 \approx 49.31\%$ of [29], and about $427/503 \approx 84.89\%$ of [31]. The reason is that compared to other relevant schemes, the signature form of our proposed scheme is simpler and more lightweight.

D. Storage Overhead

This subsection will thoroughly evaluate the storage overhead of smart devices. It aims to demonstrate that our proposed scheme needs lower storage costs and is suitable for smart devices with limited storage capacity.

In our proposed scheme, the data to be stored by the smart device is $\{lsk_i, n_i, \omega_i, \pi_i, RID_i, P_{pub}, P\}$, so its storage overhead is $(48 + 48 + 48 + 48 + 32 + 97 + 97) = 418$ bytes. Since the key derivation algorithm is used in our proposed scheme, which allows the smart device to derivates pseudonyms and the corresponding anonymous secret keys independently, the storage overhead of the smart device is constant. Otherwise, if the key derivation algorithm is not used, the smart device needs to request multiple pseudonyms and corresponding anonymous secret keys from the key generation center and store them in advance, with a storage overhead of $418 + (32 + 48)l = 418 + 80l$ bytes (assuming the number of pseudonyms stored in advance is l). We use the same approach to calculate the storage overhead of the other schemes and the results are shown in Fig. 8.

From the Fig. 8, we can see that the storage overhead is about 1095 bytes in scheme [30], which is $1095 - 418 = 677$ bytes more than our proposed scheme. This is because the pairing-based group signature used in this scheme is more

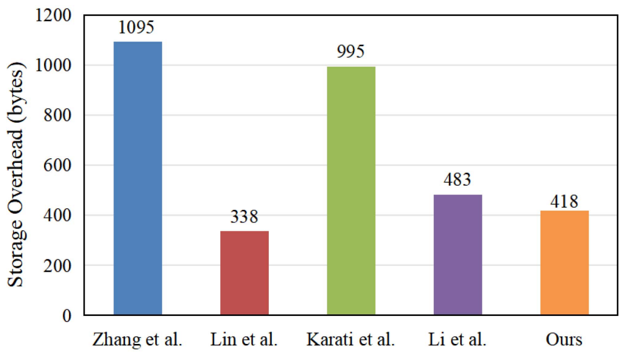


Fig. 8. Comparison of storage overhead.

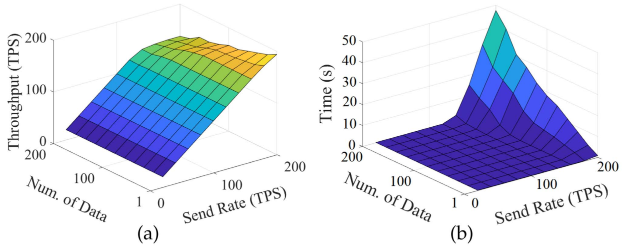


Fig. 9. Write operation on blockchain.

complex and requires more parameters. The storage overhead of scheme [30] is about 338 bytes, which is $418 - 338 = 80$ bytes less than our proposed scheme. This is because this scheme also supports key derivation and requires only a private seed to derive pseudonymous and anonymous private keys. However, the computational overhead in our proposed scheme is lower than the scheme [30]. The storage overhead of scheme [29] is about 995 bytes, which is $995 - 418 = 577$ bytes more than our proposed scheme. This is because there are more elements in $\mathbb{G}_1$ and $\mathbb{G}_T$ stored by the smart device in this scheme. The storage overhead of scheme [31] is about 483 bytes, which is $483 - 418 = 65$ bytes more than our proposed scheme. Similar to the scheme [29], the scheme [31] needs to store more elements in $\mathbb{G}_1$ and $\mathbb{G}_2$.

E. On-Chain Evaluation

In our proposed scheme, the blockchain performs write operations for certificate storage and revocation, while the querying of certificates during authentication involves read operations of the blockchain. Therefore, we test the performance of write and read operations in the blockchain according to the experimental setup. The experimental results are shown in Figs. 9 and 10.

As shown in Fig. 9(a), when the number of data written in a batch is constant, the system throughput increases as the sending rate increases. When the send rate reaches 160 TPS, the system throughput gradually reaches saturation. In addition, we find that when the number of data written in a batch increases, its system throughput reaches saturation sooner. And when the sending rate is consistent and exceeds 160 TPS, the system throughput decreases as the number of data written in a batch increases. For example, under the send rate as 180 TPS, the system throughput is 178.7 TPS, 163.8 TPS, and 137.5 TPS when the number of data is 1, 100, and 200, respectively. As shown in Fig. 9(b), when the sending rate is lower than 140 TPS and consistent, the average latency increases slowly as the number of data written in a batch increases. For example, at a send rate of 20 TPS, the average write latency is 0.08 s when the number of data written in a batch is 1, and only 0.13 s when the number of data written in a batch is 200. When the sending rate exceeds 160 TPS, the average latency increases significantly with the number of data written in a batch. The reason is that PCs are limited in performance. When the sending rate surpasses 160 TPS, PCs struggle to process requests promptly, leading to a notable increase in average latency. Using higher performance servers will effectively solve this issue.

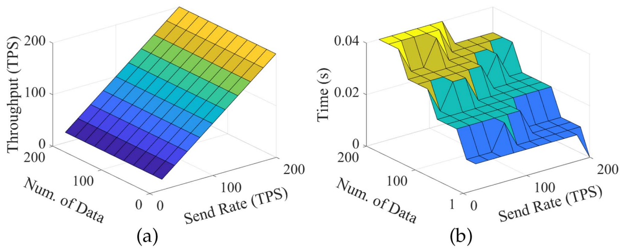


Fig. 10. Read operation on blockchain.

As shown in Fig. 10(a), when the number of data read in a batch is constant, the system throughput increases as the send rate increases. From Fig. 10(b), we can see that when the send rate is constant, although the average latency of the read operation tends to increase with the number of data read in a batch, the latency remains at a low level. For example, when the sending rate is 200 TPS and the number of data is 200, the query latency is 0.03 s. In our proposed scheme, the authentication process requires only read operations and the read latency is acceptable.

Insight: When the amount of data is within a certain range and constant, the latency of batch writes/reads is lower than that of one-by-one writes/reads. Considering the performance bottleneck of the blockchain and the maximum acceptable latency of each service, we can design a practical caching mechanism. The system can periodically perform batch write/read operations using cached data. In addition, we can choose a blockchain with better performance and faster processing speed to accelerate message verification in IIoT.

VIII. Conclusion

This paper proposes a blockchain-assisted lightweight authentication scheme, which achieves an effective tradeoff between data anonymity, data traceability, pseudonym management, and smart devices' flexible revocation in IIoT. Firstly, we use Okamoto's protocol and elliptic curve cryptography to design a lightweight authentication algorithm that achieves data anonymity and traceability. Secondly, we design a two-level key derivation algorithm and use blockchain technology to achieve efficient management of certificates and flexible revocation of smart devices. The security proof and analysis show that the proposed scheme meets the requirements in terms of security.

Finally, the performance evaluation results show that our proposed scheme has low computational and communication overhead, which is suitable for IIoT systems with high requirements for real-time and flexibility. In future work, we will study the relationship between performance bottlenecks of blockchain and latency requirements of IIoT services, then design a practical data storage and query mechanism.

Acknowledgment

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

References

[1] Y. Wu, H.-N. Dai, and H. Wang, “Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0,” *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2300–2317, Feb. 2021.

[2] L. D. Xu, W. He, and S. Li, “Internet of Things in industries: A survey,” *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[3] Q. Zhang, C. Zhu, L. T. Yang, Z. Chen, L. Zhao, and P. Li, “An incremental CFS algorithm for clustering large data in Industrial Internet of Things,” *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1193–1201, Jun. 2017.

[4] M. H. ur Rehman, I. Yaqoob, K. Salah, M. Imran, P. P. Jayaraman, and C. Perera, “The role of Big Data analytics in Industrial Internet of Things,” *Future Gener. Comput. Syst.*, vol. 99, pp. 247–259, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18313645

[5] T. Wang, P. Wang, S. Cai, Y. Ma, A. Liu, and M. Xie, “A unified trustworthy environment establishment based on edge computing in industrial IoT,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6083–6091, Sep. 2020.

[6] M. Younan, E. H. Houssein, M. Elhoseny, and A. A. Ali, “Challenges and recommended technologies for the Industrial Internet of Things: A comprehensive review,” *Measurement*, vol. 151, 2020, Art. no. 107198. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224119310644

[7] T. Qiu, Y. Zhang, D. Qiao, X. Zhang, M. L. Wymore, and A. K. Sangaiah, “A robust time synchronization scheme for Industrial Internet of Things,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3570–3580, Aug. 2018.

[8] J. Lu, J. Shen, P. Vijayakumar, and B. B. Gupta, “Blockchain-based secure data storage protocol for sensors in the Industrial Internet of Things,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 8, pp. 5422–5431, Aug. 2022.

[9] G. Rathee, F. Ahmad, R. Iqbal, and M. Mukherjee, “Cognitive automation for smart decision-making in Industrial Internet of Things,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 2152–2159, Mar. 2021.

[10] L. Bu, Y. Zhang, H. Liu, X. Yuan, J. Guo, and S. Han, “An IIoT-driven and ai-enabled framework for smart manufacturing system based on three-terminal collaborative platform,” *Adv. Eng. Informat.*, vol. 50, 2021, Art. no. 101370. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034621001233

[11] J. Cui, F. Wang, Q. Zhang, Y. Xu, and H. Zhong, “Anonymous message authentication scheme for semitrusted edge-enabled IIoT,” *IEEE Trans. Ind. Electron.*, vol. 68, no. 12, pp. 12921–12929, Dec. 2021.

[12] H. Liao et al., “Learning-based context-aware resource allocation for edge-computing-empowered industrial IoT,” *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4260–4277, May 2020.

[13] M. Tanveer, A. Alkhayyat, A. U. Khan, N. Kumar, and A. G. Alharbi, “Reap-IIoT: Resource-efficient authentication protocol for the Industrial Internet of Things,” *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24453–24465, Dec. 2022.

[14] 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./Apr. 2023.

[15] C. Feng, B. Liu, Z. Guo, K. Yu, Z. Qin, and K.-K. R. Choo, “Blockchain-based cross-domain authentication for intelligent 5G-enabled Internet of Drones,” *IEEE Internet Things J.*, vol. 9, no. 8, pp. 6224–6238, Apr. 2022.

[16] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, and C. Su, “Blockchain-based reliable and efficient certificateless signature for IIoT devices,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 7059–7067, Oct. 2022.

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

[18] J. Sengupta, S. Ruj, and S. D. Bit, “A secure fog-based architecture for Industrial Internet of Things and industry 4.0,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2316–2324, Apr. 2021.

[19] Y. Zhang, H. Huang, L.-X. Yang, Y. Xiang, and M. Li, “Serious challenges and potential solutions for the Industrial Internet of Things with edge intelligence,” *IEEE Netw.*, vol. 33, no. 5, pp. 41–45, Sep./Oct. 2019.

[20] 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. Forensics Secur.*, vol. 17, pp. 3789–3800, 2022.

[21] H. Yang et al., “Blockchain-enabled tripartite anonymous identification trusted service provisioning in industrial IoT,” *IEEE Internet Things J.*, vol. 9, no. 3, pp. 2419–2431, Feb. 2022.

[22] X. Zhang, H. Zhong, C. Fan, I. Bolodurina, and J. Cui, “CBACS: A privacy-preserving and efficient cache-based access control scheme for software defined vehicular networks,” *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 1930–1945, 2022.

[23] L. Wei, J. Cui, Y. Xu, J. Cheng, and H. Zhong, “Secure and lightweight conditional privacy-preserving authentication for securing traffic emergency messages in VANETs,” *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 1681–1695, 2021.

[24] M. Zhang, J. Zhou, G. Zhang, M. Zou, and M. Chen, “EC-BAAS: Elliptic curve-based batch anonymous authentication scheme for Internet of Vehicles,” *J. Syst. Archit.*, vol. 117, 2021, Art. no. 102161. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762121001156

[25] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, “PA-CRT: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks,” *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 2, pp. 722–735, Mar./Apr. 2021.

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

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

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

[29] A. Karati, S. H. Islam, and M. Karuppiah, “Provably secure and lightweight certificateless signature scheme for IIoT environments,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3701–3711, Aug. 2018.

[30] 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 Secur.*, vol. 18, pp. 233–247, 2023.

[31] T. Li, H. Wang, D. He, and J. Yu, “Designated-verifier aggregate signature scheme with sensitive data privacy protection for permissioned blockchain-assisted IIoT,” *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 4640–4651, 2023.

[32] T. P. Pedersen, “Non-interactive and information-theoretic secure verifiable secret sharing,” in *Proc. Conf. Adv. Cryptology — CRYPTO*, J. Feigenbaum, Ed., 1992, pp. 129–140.

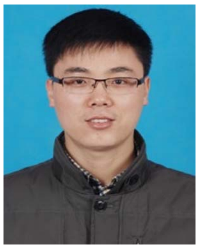
[33] H. Xiong, Y. Wu, C. Su, and K. H. Yeh, “A secure and efficient certificateless batch verification scheme with invalid signature identification for the Internet of Things,” *J. Inf. Secur. Appl.*, vol. 53, 2020, Art. no. 102507. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212619307999

[34] 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.*, 2006, pp. 390–399, doi: 10.1145/1180405.1180453.

[35] “Miracl core,” 2020. [Online]. Available: https://github.com/miracl/core

[36] “Hyperledger,” 2020. [Online]. Available: https://github.com/hyperledger/fabric

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



Jie Cui (Senior Member, IEEE) was born in Henan Province, China, in 1980. He received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2012. He is currently a Professor and Ph.D. supervisor with the School of Computer Science and Technology, Anhui University, Hefei. He has more than 150 scientific publications in reputable journals, such as IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE TRANSACTIONS ON CLOUD COMPUTING and IEEE TRANSACTIONS ON MULTIMEDIA, academic books and international conferences. His research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security and software-defined networking (SDN).

Qingyang Zhang (Member, IEEE) was born in Anhui Province, China, in 1992. He received the B.Eng. degree and Ph.D. degree in computer science from Anhui University, Hefei, China, in 2021. He is currently a Lecture with the School of Computer Science and Technology, Anhui University. His research interests include edge computing, computer systems, and security.

Debiao He (Member, IEEE) received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, Wuhan, China in 2009. He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University, and Shanghai Key Laboratory of Privacy Preserving Computation, MatrixElements Technologies, Shanghai, China. He has authored or coauthored more than 100 research papers in refereed international journals and conferences, such as IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION SECURITY AND FORENSIC, and Usenix Security Symposium. His main research interests include cryptography and information security, in particular, cryptographic protocols. He was the recipient of the 2018 IEEE Sysems Journal Best Paper Award and 2019 IET Information Security Best Paper Award. His work has been cited more than 10000 times at Google Scholar. He is in the Editorial Board of several international journals, such as *Journal of Information Security and Applications, Frontiers of Computer Science*, and *Human-centric Computing & Information Sciences*.

Hong Zhong (Member, IEEE) was born in Anhui Province, China, in 1965. She received the Ph.D. degree in computer science from the University of Science and Technology of China, Hefei, China, in 2005. She is currently a Professor and Ph.D. supervisor with the School of Computer Science and Technology, Anhui University, Hefei. She has more than 200 scientific publications in reputable journals, such as 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, IoT security, vehicular ad hoc network, cloud computing security and software-defined networking (SDN).