Journal of Systems Architecture 160 (2025) 103368

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Lightweight batch authentication and key agreement scheme for IIoT gateways

Xiaohui Ding [a,*], Jian Wang [a], Yongxuan Zhao [b], Zhiqiang Zhang [a]

[a] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China
[b] Information Technology Research Center, China Academy of Aero-Engine Research, Beijing, 101304, China

ARTICLE INFO

Keywords:
Industrial internet of things
Batch authentication and key agreement
Gateway lightweight

ABSTRACT

Existing authentication and key agreement (AKA) schemes face two primary challenges in IIoT, where users dynamically communicate with multiple industrial devices. The first is significant computational and communication overhead, along with security vulnerabilities. Another is inability to achieve gateway lightweight solutions. To address these issues, this paper proposes a gateway lightweight batch AKA scheme based on elliptic curve cryptography for IIoT. When users access multiple industrial devices, they only need to send a batch authentication request to the gateway. Based on this request, the gateway generates a time-limited token combining Chinese Remainder Theorem (CRT), enabling users to efficiently complete AKA with multiple devices in batch manner. Furthermore, the application of the CRT allows the gateway to efficiently update the time-limited token when the user's accessed devices change. Finally, due to the use of the time-limited token, the entire scheme process requires only one round of interaction between the gateway and the user, ensuring a lightweight nature of the gateway. The security of the proposed scheme is proved through formal security proofs, heuristic analysis, and scyther tools. Performance analysis shows that, compared to the compared schemes, the proposed scheme meets all listed security requirements with the lower computational and communication overheads.

1. Introduction

In recent years, advances in computer technology and wireless sensor networks have fueled the rapid development of Internet of Things (IoT) technology. IoT is a self-organizing network of interconnected devices that can interact without human intervention [1]. IoT terminal devices generate vast amounts of valuable data in real-time, positioning IoT as the third wave of global informatization following the advent of computers and the Internet [2]. With the development of emerging communication technologies such as 5G, the demand for IoT applications continues to grow. It is estimated that by 2030, the number of IoT devices will exceed 100 billion [3]. IoT has been widely applied in smart agriculture, autonomous driving, smart healthcare, and industrial sectors, etc [4]. In the industrial field, it is referred to as the IIoT, industry 4.0, etc [5,6]. IIoT drives traditional industries toward intelligent and informatized development, enabling remote monitoring and automatic control of industrial production, which significantly enhances production efficiency [7].

Fig. 1 illustrates a typical IIoT system model, which involves three main entities: users, gateways, and industrial devices. After being authenticated by the gateway, users can remotely access industrial devices to retrieve data or directly control them. In practice, for a given industrial production task, users need to interact with multiple industrial devices, and the devices that need to be accessed or controlled will change in real-time as the task progresses. Therefore, to achieve more intelligent and efficient task completion, IIoT communication scenarios exhibit two typical characteristics: first, users need to interact with multiple industrial devices; second, the industrial devices that users need to access frequently change.

In the IIoT, users interact with industrial devices, often requiring the transmission of communication information over open channels, which introduces significant security risks. To ensure security, many researchers have proposed AKA schemes tailored for the IoT domain, aimed at authentication the legitimacy of the identities of communication entities and negotiating session keys to secure subsequent communications [8–12]. However, these schemes primarily focus on authentication and key agreement between single user and single device, resulting in one-to-one AKA schemes. If such schemes were to be applied in the IIoT, users would need to repeatedly execute the scheme to complete authentication and key agreement with multiple industrial

* Corresponding author.
E-mail address: dingxiaohui@nuaa.edu.cn (X. Ding).

https://doi.org/10.1016/j.sysarc.2025.103368
Received 9 September 2024; Received in revised form 26 December 2024; Accepted 6 February 2025
Available online 15 February 2025

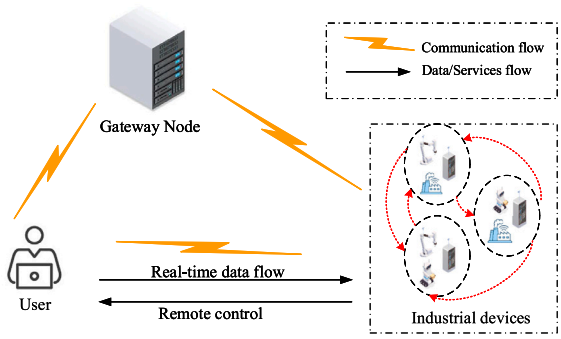


Fig. 1. IIOT system architecture.

devices. This would lead to significant computational and communication overhead, making them unsuitable for resource-constrained IIoT environments [13,14].

To make the schemes more suitable for scenarios involving communication between users and multiple devices, researchers have proposed group-based AKA schemes [15–17], batch authentication schemes [18–22], and AKA schemes designed specifically for multi-device communication [13,14,23,24]. However, group-based AKA schemes require all devices in the group to share a common group key, which makes them vulnerable to impersonation attacks by malicious devices. Moreover, when the set of industrial devices accessed by the user changes, group AKA schemes face challenges with group membership updates and group key renewal. Compared to group schemes, batch schemes offer greater flexibility, allowing users to independently select multiple devices for batch authentication. However, most existing batch schemes focus only on batch message authentication [18–20] and identity verification [21,22], without considering the simultaneous implementation of batch authentication and key agreement. In recent years, researchers have proposed several AKA schemes with batch processing attributes for multi-device communication environments [13,14,23,24]. These schemes enable users to efficiently complete authentication and key agreement with multiple terminal devices simultaneously. However, the schemes presented in the [13,23,24] exhibit notable deficiencies in resisting impersonation attacks and ensuring forward security.

Zhang et al. [14] proposes a many-to many AKA scheme for vehicular networks, allowing users to efficiently complete authentication with multiple cloud servers and negotiate different session keys for each. This scheme offers a high level of security. However, analysis reveals that the cost of implementing batch authentication and key agreement between users and cloud servers is a significant computational and communication overhead borne by the trusted center, which raises concerns about potential single points of failure. Although existing schemes consider lightweight construction to accommodate the resource-constrained IIoT environment, most of them focus primarily on minimizing the computational load for users or end devices, with little attention given to the lightweight design of the gateway itself. In an IIoT system, the gateway is connected to a large number of industrial devices and must assist users in completing authentication and key agreement with multiple devices. Therefore, the efficiency of the gateway node directly affects the overall performance of the AKA schemes, making it crucial to consider the lightweight design of the gateway [25].

Problem Statement: Existing AKA schemes are ineffective for communication scenarios in the IIoT, where users dynamically interact with multiple industrial devices. Traditional one-to-one AKA schemes face significant computational and communication overhead issues. Group-based AKA schemes have security vulnerabilities, such as being unable to prevent impersonation attacks by malicious group devices, and they also encounter challenges related to group updates and group key updates. While batch processing offers greater flexibility, most schemes primarily focus on batch identity or message authentication, failing to achieve simultaneous batch authentication and key agreement. Some AKA schemes with batch processing attributes for multi-terminal device communication face security issues and potential single points of failure in gateways. To the best of our knowledge, no existing scheme considers achieving batch authentication and key agreement between users and multiple industrial devices while ensuring the lightweight nature of the gateway.

In summary, it is necessary to design an AKA scheme that is better suited to the unique communication scenarios of the IIoT. Such a scheme should efficiently enable users to authenticate with and establish session keys for multiple industrial devices, while also accommodating minimal overhead when the industrial devices that a user wishes to access changes. Additionally, the proposed scheme should ensure the lightweight design of the gateway to prevent it from becoming a performance bottleneck for the entire system. Based on these requirements, this paper proposes a gateway lightweight batch AKA scheme for IIoT environments. The main contributions of this paper are as follows:

(1) **Batch Authentication and Key Agreement**: Based on elliptic curve cryptography combined with the Chinese Remainder Theorem and the concept of time-limited tokens, this paper presents a batch AKA scheme. This scheme allows users to independently select and authenticate multiple industrial devices in batches. Users only need to send a single batch authentication request to the gateway. In response, the gateway generates time-limited tokens using the Chinese Remainder Theorem. With the tokens, users can efficiently perform mutual authentication with multiple industrial devices and negotiate different session keys with each device. This approach effectively addresses the high computational and communication overhead associated with traditional one-to-one AKA schemes and mitigates the risk of impersonation attacks due to shared group keys in group AKA schemes.

(2) **Efficient Token Update**: Due to the use of the Chinese Remainder Theorem, the gateway can efficiently update time-limited tokens when the industrial devices that the user needs to access change, thereby avoiding the challenges of group updates and group key renewal encountered in group AKA schemes.

(3) **Gateway Lightweight**: Due to the use of time-limited tokens, in the batch authentication and key negotiation process, gateway only needs to interact with user in one round to assist user complete the authentication and key agreement with multiple industrial devices, without any direct interaction between the gateway and the industrial devices, thereby ensuring the lightweight nature of the gateway. Furthermore, the scheme does not involve computationally intensive operations such as bilinear pairings, ensuring that the computational and communication overhead for both users and industrial devices remains lightweight.

(4) **Security and Performance Analysis** : The security of the proposed scheme is demonstrated through formal security proofs, heuristic analysis, and Scyther tools. Performance analysis shows that, compared to existing schemes, the proposed scheme meets all listed security requirements with the lower computational and communication overheads and provides a significant advantage in terms of the lightweight nature of the gateway node.

The remainder of this paper is organized as follows: Section 2 reviews the related work. Section 3 presents the preliminaries and system model. Section 4 describes the detailed construction of the proposed scheme. Section 5 provides the security proof and analysis of the proposed scheme. A performance comparison between the proposed scheme and related schemes is presented in Section 6. Finally, Section 7 concludes the paper.

2. Related work

Existing AKA schemes focus on one-to-one AKA schemes for communication between users and single terminal devices, as well as group AKA schemes and batch processing schemes for communication with multiple terminal devices.

In 2009, DAS et al. [26] first proposed a lightweight two-factor authentication scheme for wireless sensor networks (WSNs, a critical component of IIoT), In their scheme, users authenticate themselves by entering a personal password and using a smart card. However, since the scheme relies solely on hash functions for security, it is unable to effectively resist various attacks, such as denial-of-service (DoS) attacks. Consequently, several authentication or key management schemes for WSN communication have been proposed [27–29]. With the development of IoT technology, and in order to balance security and lightweight requirements, several ECC-based AKA schemes for the IIoT have been proposed [10–12]. Li et al. [11] designed a privacy-preserving AKA scheme for the IIoT based on elliptic curve cryptography. Since the user and the gateway do not store the same secret value, the scheme is resistant to desynchronization attacks. However, further analysis reveals that the session key generation in this scheme does not involve long-term secret values, rendering it vulnerable to ephemeral secret leakage attacks. Similarly, the user authentication protocol proposed by Srinivas et al. [12] for the IoT-based intelligent transportation systems fails to effectively resist privileged insider attacks. In 2022, Chen et al. [10] proposed an ECC-based AKA scheme for industrial control systems, which can resist most protocol attacks. However, further analysis reveals that the scheme lacks essential properties such as malicious user traceability and terminal device update capabilities. Moreover, all of the aforementioned schemes are designed for one-to-one environments. Given the presence of a large number of industrial devices in the IIoT, deploying these schemes could result in excessive computational and communication overheads as well as single points of failure. Therefore, these schemes are not suitable for real-world IIoT communication environments.

To make AKA schemes more suitable for multi-device communication scenarios, several group AKA schemes [15–17] have been proposed in recent years. Mandal et al. [15] introduced a certificateless authenticated group key agreement protocol based on elliptic curve cryptography, which ensures the non-repudiation of communication messages between senders and receivers, and establishes a group key for subsequent communication. To enhance practicality, the protocol also supports the dynamic addition and revocation of group members and considers the forward security of the session key. Xu et al. [16] designed a quantum-resistant identity-based group authentication scheme for IoT environments with concurrent access by numerous devices. The scheme is constructed using lattice-based aggregate signature algorithms and identity-based encryption algorithms, achieving quantum security while facilitating group authentication for multiple devices, and effectively addressing the issues related to certificate management. Wu et al. [17] proposed a lightweight group AKA protocol for the IIoT environment, based on symmetric bivariate polynomials, which achieves both authentication and group session key agreement. Compared to previous group AKA protocols, their scheme is more efficient. Although group AKA schemes are more suitable for multi-device communication scenarios compared to one-to-one AKA schemes, they face challenges in updating group keys when the industrial devices accessed by the user frequently change. Additionally, since all group devices share the same group key, these schemes cannot effectively prevent impersonation attacks by malicious devices.

The batch mode is more flexible than the group mode and is better suited for real-world communication scenarios in the IIoT. Pu et al. [19] proposed a lightweight message aggregation authentication protocol for drone networks, which is constructed using pairing-based cryptography and physically unclonable functions. This protocol enables secure and efficient data transmission between a base station and a group of drones. Shen et al. [21] proposed a batch authentication scheme in the vehicular network based on blockchain technology. In this scheme, a proxy vehicle selection algorithm is utilized to select proxy vehicles responsible for batch authenticating vehicles within a designated area. This effectively alleviates the authentication load when a large number of vehicles simultaneously connect to the same RSU. Additionally, the scheme employs a certificate-free mechanism and identity-based prefix encryption algorithms to achieve efficient batch authentication and protect the identity privacy of proxy vehicles. However, the aforementioned schemes, as well as most existing batch processing schemes, primarily focus on batch message authentication [18,20] or batch identity authentication [22], failing to achieve simultaneous batch authentication and key agreement.

Recently, some AKA schemes with batch processing capabilities have been proposed for multi-terminal communication scenarios [13,14,23,24], but these schemes also have limitations in terms of applicability and security. Cui et al. [23] proposed a scalable conditional privacy-preserving authentication scheme for multi-cloud environments, which is suitable for multi-terminal settings and demonstrates high efficiency. However, analysis reveals that the session key generation process in their scheme includes the identity information of the cloud server, allowing authenticated users to obtain the server's real identity. As a result, their scheme cannot effectively resist impersonation attacks or man-in-the-middle attacks. Vinoth et al. [24] utilized the Chinese Remainder Theorem and symmetric cryptography to achieve authentication and key agreement between users and multiple IIoT devices. However, in their scheme, the session keys negotiated between the user and multiple devices are identical, allowing devices to impersonate each other, which presents a significant security vulnerability. Yang et al. [13] also constructed a one-to-many AKA scheme for the IIoT based on the Chinese Remainder Theorem, addressing the issue in Vinoth et al. [24] scheme where the session keys between the user and multiple devices were identical. However, further analysis reveals that both Yang et al. [13] and Vinoth et al. [24] lack forward security. According to the work of Wang et al. [25] and Ma et al. [30], to achieve forward security, a scheme must perform at least two public key cryptographic operations on the device side. Since neither Yang et al. [13] nor Vinoth et al. [24] schemes deploy public key operations on the industrial devices, they fail to meet the forward security requirement.

Zhang et al. [14] proposed a secure and efficient many-to-many AKA scheme for vehicular networks. The scheme allows vehicle users to perform batch authentication and key agreement with multiple cloud servers, while resisting various known protocol attacks. However, further analysis reveals that the efficiency of the batch authentication and key agreement comes at the cost of significant computational and communication overhead for the trusted center (which is equivalent to the gateway in an IIoT environment). Most existing schemes, when designed, focus primarily on minimizing the computational overhead for users and end devices, with little attention given to the lightweight nature of the gateway. In 2023, Wang et al. [25] proposed a lightweight user authentication scheme for cloud-assisted IoT environments. The scheme achieves gateway lightweighting by offloading most of the computational and communication burdens from the gateway to the cloud server. However, it requires the cloud server to be fully trusted during the authentication and key agreement process, which introduces an overly strong security assumption. Moreover, the scheme does not consider adaptation to multi-device application environments, making it unsuitable for scenarios involving frequent communication between users and multiple industrial devices in IIoT environments.

In summary, existing schemes applied in IIoT environments, where users dynamically communicate with multiple industrial devices, encounter issues related to usability, security, and the lightweight nature of gateways. Regarding usability, traditional one-to-one AKA schemes suffer from excessive computational and communication overhead, while group AKA schemes face complexities related to group updates and group key updates. Moreover, batch identity authentication

and batch message authentication schemes fail to achieve simultaneous batch authentication and key agreement. In terms of security, both group AKA schemes and existing batch processing attribute AKA schemes designed for multi-terminal communication scenarios exhibit deficiencies in critical security attributes such as resistance to impersonation attacks and forward security. Furthermore, existing schemes rarely consider the lightweight requirements for gateway.

In conclusion, existing schemes fail to achieve batch authentication and key agreement between users and multiple industrial devices while ensuring the lightweight nature of the gateway. In the IIoT, ensuring secure and efficient communication between users and multiple devices, as well as avoiding single points of failure in gateway, are critical issues that require urgent solutions. Therefore, it is essential to propose a gateway lightweight batch AKA scheme suitable for the IIoT.

3. Preliminary, system model, threat model and security objectives

This section first introduces the fundamental concepts required for constructing the proposed scheme. Then, the system model and security objectives of the proposed scheme are presented.

3.1. Preliminary

Elliptic Curve Cryptosystems: elliptic curve cryptosystems were first proposed by miller [31] and koblite [32] et al. Given a large prime p and a finite field $\mathbb{F}_p$, choose a parameter $a, b \in \mathbb{F}_p$ to generate an elliptic curve $E: y^2 = x^3 + ax + b \bmod p$ based on $\mathbb{F}_p$. Let O be an infinity point on E, then O and all points on E form an additive cyclic group G of order P generating element q.

Elliptic Curve Discrete Logarithm Problem (ECDL)[14]: Given two random points $P, Q \in G$ on elliptic curve E where $Q = xP$, $x \in Z_q^*$. Then the ECDL problem refers to the difficulty of finding a positive integer x in probabilistic polynomial time (PPT) when points P and Q are known.

Elliptic Curve Computation Diffie–Hellman problem (ECCDH) [33]: Given point $P, xP, yP \in G$, where $x, y \in Z_q^*$. Then for any PPT adversary the advantage of computing $xyP \in G$ without knowing x, y is negligible.

One-Way Collision-Resistant Hash Function: One-way collision-resistant hash function is a deterministic algorithm that is irreversible and collision-resistant. It takes as input a binary string of arbitrary length and outputs a deterministic length binary string.

Chinese Remainder Theorem(CRT): The CRT [13,34] is an important theorem in number theory that has been used to solve a system of congruence equations in the modulo-invariant case, where the system of congruence equations takes the following form:

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \quad \vdots \\ x \equiv a_n \pmod{m_n} \end{cases} \tag{1}$$

Let $m_1, m_2, \ldots m_n$ be two mutually prime positive integers, and $a_1, a_2, \ldots a_n$ be any given n positive integers. Then, for a positive integer $a_i, i \in [1, n]$, the general solution of the system of congruence equations is:

$$\begin{aligned} x &= a_1 t_1 M_1 + a_2 t_2 M_2 + \cdots + a_n t_n M_n + kM \\ &= \sum_{i=1}^{n} a_i t_i M_i + kM, k \in \mathbb{Z} \end{aligned} \tag{2}$$

where $M = m_1 \times m_2 \times \cdots \times m_n = \prod_{i=1}^{n} m_i$ is the product of integers $m_1, m_2, \ldots m_n$, $M_i = M/m_i$ denotes the product of $(n-1)$ integers except m_i, and $M_i t_i \equiv 1 \pmod{m_i}$, $i \in [1, n]$. The CRT states that the system of primary congruence equations has the following unique solution in the case of mode M:

$$x = \left(\sum_{i=1}^{n} a_i t_i M_i \right) \bmod M \tag{3}$$

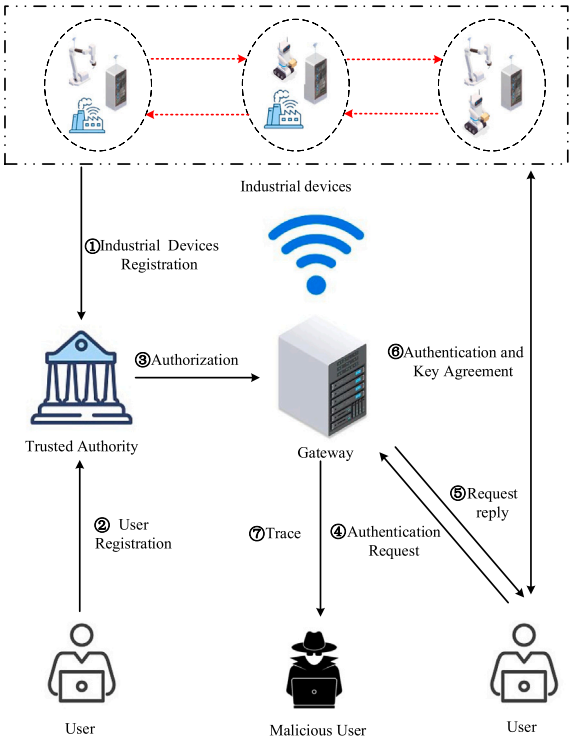


Fig. 2. system model.

3.2. System model

The system model of the proposed gateway lightweight batch AKA scheme for the IIoT is shown in Fig. 2. The system consists of four types of entities: a trusted authority, a gateway, users, and industrial devices. The detailed descriptions of each entity are as follows:

Trusted Authority(TA): TA is a fully reliable entity, typically operated by a government authority, with sufficient computational and storage capabilities. Its primary responsibilities include generating and publishing system parameters, registering users and industrial devices, and authorizing gateways. Additionally, the TA is responsible for holding malicious users accountable.

User: Users must register at TA, after which they can communicate with the gateway and industrial devices using smart mobile devices. When users wish to access industrial data collected by the devices or directly manipulate them, they need to complete mutual authentication with the industrial devices and negotiate a session key for secure subsequent communication. The user sends a batch authentication request to the gateway. Upon verifying the legitimacy of the user's identity, the gateway issues a time-limited token, enabling the user to complete authentication and key agreement with the industrial devices using the token.

Gateway: The gateway is a fully trusted entity that requires authorization from TA. It is generally considered to possess greater computational and storage capabilities than industrial devices. The gateway is responsible for issuing time-limited tokens to users and assisting them in completing batch authentication and key agreement with multiple industrial devices.

Industrial device: Industrial devices register at TA and use time-limited tokens to complete authentication and key agreement with users. Upon successful authentication and key agreement, the devices can securely transmit the collected industrial data to users after encrypting it with the session key, or they can execute corresponding industrial tasks based on user instructions.

3.3. Threat model and security objectives

- **Threat model**

This paper uses the standard Dolev–Yao (DY) model [35,36] to assess the security of the proposed AKA scheme. The DY model stipulates that an adversary $\mathcal{A}$ can control the insecure public communication channel between the parties and can read, modify, delete, forge, replay, or even inject false information into the channel. Additionally, when considering forward security, the adversary not only possesses all the capabilities defined in the DY model but can also acquire secret credentials, session states, and session keys from the communicating entities. Therefore, forward security must ensure that a compromise of the system does not affect the security of previous sessions. This paper assumes that in the IIoT environment, the gateway is a fully trusted entity, while users and industrial devices are considered untrusted participants.

- **Security objectives**

Based on the above threat model, the proposed scheme in this paper should meet the following security objectives:

(1) Mutual authentication and key agreement: The scheme should enable mutual authentication between the user and industrial devices, ensuring that only authenticated users can access the data collected by the industrial devices. Additionally, the scheme should facilitate the negotiation of specific session keys between the user and industrial devices for secure communication in subsequent interactions.
(2) User anonymity: To ensure the privacy of the user's identity, information transmitted over public channels should not reveal the user's true identity.
(3) Forward security: The scheme should achieve forward security, meaning that even if an adversary obtains the long-term secret values of the participants and the session state or session keys of the current session, they should not be able to compute the session keys of previous sessions.
(4) Unlinkability: The scheme should ensure unlinkability, meaning that an adversary should not be able to link two different messages transmitted over the public channel to the same user or industrial device.
(5) Resistance to Various Attacks: The scheme should be capable of withstanding common protocol attacks, such as replay attacks, spoofing attacks, privileged insider attacks, and man-in-the-middle attacks, etc.

4. Proposed scheme

The scheme consists of seven formalized algorithms, which are system establishment, industrial device registration, user registration, gateway authorization, authentication and key agreement, industrial device update, and malicious user tracking. The main symbols used in the scheme are described in Table 1.

4.1. System establishment

TA inputs the system security parameters λ and generates the system parameters accordingly. TA generates an additive cyclic group G based on non-singular elliptic curves, whose order is q and the group generator element is P. Randomly select $msk \in z_q^*$ as the system master key and compute $mpk = msk \cdot P$ as the system's master public key. Randomly select secure hash functions $h : \{0,1\}^* \rightarrow z_q^*$, $h_1 : \{0,1\}^* \rightarrow \{0,1\}^l$. Choose GID as the identity of the gateway, choose s as the gateway private key and compute $PK = s \cdot P$ as the gateway public key. Finally TA announces the system parameters $params : \{G, P, mpk, h, h_1, GID, PK\}$.

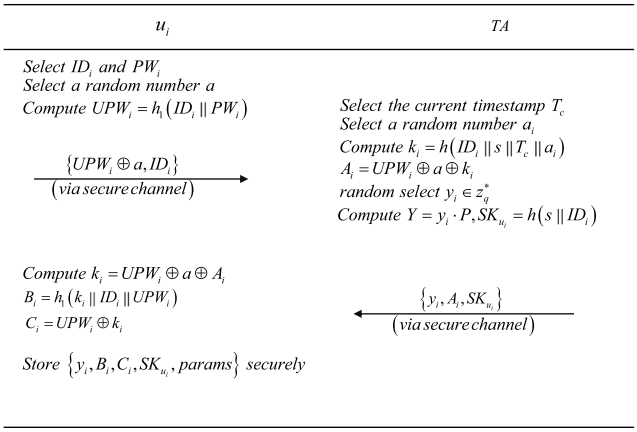


Fig. 3. User registration phase.

4.2. Industrial device registration

TA selects its identity information SID_j for industrial device SD_j, randomly chooses $x_j \in z_q^*$ as the private key of the industrial device, and calculates $SK_{SD_j} = h(s \| SID_j)$ as the long-term session key between the device and the gateway. TA sends the parameter $\{x_j, SID_j, SK_{SD_j}\}$ securely to the industrial device (e.g., by offline registration), and SD_j secretly stores the parameter $\{x_j, SID_j, SK_{SD_j}\}$ to complete the registration.

4.3. User registration

User u_i selects his identity ID_i, password PW_i and computes $UPW_i = h_1(ID_i \| PW_i)$, randomly selects $a \in z_q^*$, and securely sends the registration request parameter $\{UPW_i \oplus a, ID_i\}$ to TA.

After receiving the registration request, the TA randomly selects the current timestamp T_c and a random number $a_i \in z_q^*$, then calculates $k_i = h(ID_i \| s \| T_c \| a_i)$, $A_i = UPW_i \oplus a \oplus k_i$. Randomly select $y_i \in z_q^*$ as the user's private key, compute $Y = y \cdot P$ as the user's public key, and compute $SK_{u_i} = h(s \| ID_i)$ as the long-term session key between the user and the gateway. TA returns the parameter $\{y_i, A_i, SK_{u_i}\}$ safely to the user.

After receiving the parameters $\{y_i, A_i, SK_{u_i}\}$ returned by TA, the user calculates $k_i = UPW_i \oplus a \oplus A_i$, $B_i = h_1(k_i \| ID_i \| UPW_i)$ and $C_i = UPW_i \oplus k_i$. The user securely stores the parameter $\{y_i, B_i, C_i, SK_{u_i}, params\}$ in their mobile smart device (such as smartphone) complete the registration process (see Fig. 3).

4.4. Gateway authorization

TA authorizes the gateway, TA sends the gateway private key s and the system parameter $params$ to the gateway, and sends the industrial device and user registration parameters $\{x_j, SID_j, SK_{SD_j}\}$, $\{y_i, ID_i, SK_{u_i}\}$ to the gateway.

4.5. Authentication and key agreement phase

- **Login phase**

To communicate with industrial devices, a user must first log into their smart terminal device. User enters identity ID_i and password PW_i, the smart devices calculates $UPW_i = h_1(ID_i \| PW_i)$, $k_i = C_i \oplus UPW_i$, $B_i' = h_1(k_i \| ID_i \| UPW_i)$. Verify $B_i' \stackrel{?}{=} B_i$, If they are not equal, smart device rejects the user's login, otherwise user successfully logs into smart device (see Fig. 4).

Table 1
Notations and Definitions.

Notations	Definitions
λ	Security parameter
G	An elliptic curve cycle additive group
P	A generator of G
q	The order of G
mpk, msk	System master public–private key pair
h	Hash function
s, PK	Gateway public–private key pair
x_j	Industrial device private key
SK_{SD_j}	Long-term session key between industrial devices and the gateway
y_i, Y	User public–private key pair
SK_{u_i}	Long-term session key between user and the gateway
a, a_i, r, r_i, r_j, r_g	Random number
PID_i	User's pseudonym
T_i	Timestamp
TSK, TSK^*	Temporary secret value
SK	Session key

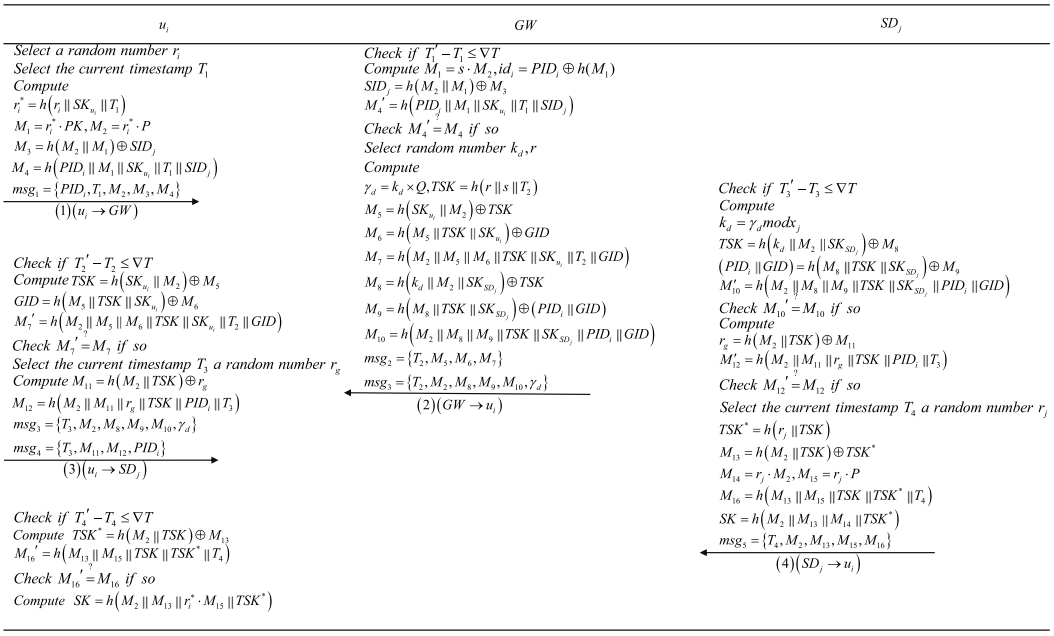


Fig. 4. Authentication and key agreement phase.

- **Authentication and key agreement**

(1) User u_i randomly selects $r_i \in z_q^*$, picks the current timestamp T_1, computes $r_i^* = h\left(r_i \| SK_{u_i} \| T_1\right)$, $M_1 = r_i^* \cdot PK$, $M_2 = r_i^* \cdot P$. u_i compute their own temporary pseudonym $PID_i = ID_i \oplus h(r_i^* \cdot PK)$. u_i communicate under a pseudonym, which enables conditional privacy protection of their identity. u_i computes $M_3 = h\left(M_2 \| M_1\right) \oplus SID_j$, where $SID_j = \left\{SID_0, \ldots, SID_n\right\}$. The user can select multiple industrial devices to access in a batch, and after the authentication and key agreement phase, negotiate distinct session keys with each device for subsequent communication. u_i computes $M_4 = h\left(PID_i \| M_1 \| SK_{u_i} \| T_1 \| SID_j\right)$. Subsequently, u_i sends $msg_1 = \left\{PID_i, T_1, M_2, M_3, M_4\right\}$to the gateway.

(2) After receiving the message sent by user, gateway first checks the validity of the timestamp by $T_1' - T_1 \le \nabla T$, where T_1' is the time gateway received msg_1. If timestamp is valid, gateway computes $M_1 = s \cdot M_2$, $ID_i = PID_i \oplus h(M_1)$, and checks if ID_i exists in revocation list. (NOTE: Gateway maintains a revocation list for storing the identity of malicious users. When a user has malicious behavior, gateway recovers its real identity based on its pseudonym and stores it in the revocation list. The tracking of malicious users will be explained later.) If ID_i is not in the revocation list, gateway computes $SID_j = h\left(M_2 \| M_1\right) \oplus M_3$, $M_4' = h\left(PID_i \| M_1 \| SK_{u_i} \| T_1 \| SID_j\right)$, verifies $M_4' \stackrel{?}{=} M_4$. If verification fails, returned error termination symbol $\perp$. Otherwise, gateway selects the current timestamp T_2, queries terminal registration tuple information based on user's identity request list SID_j, and computes $\partial g = \prod_{j=1}^{n}\left(x_j\right)$, $d_j = \partial g / x_j$, $d_j \times k_j = 1 mod x_j$, $var_j = d_j \times k_j$, $Q = \sum_{i=1}^{n} var_j$. Gateway randomly selects $k_d, r \in z_q^*$, computes $\gamma_d = k_d \times Q$, computes $TSK = h\left(r \| s \| T_2\right)$, $M_5 = h\left(SK_{u_i} \| M_2\right) \oplus TSK$, $M_6 = h\left(M_5 \| TSK \| SK_{u_i}\right) \oplus GID$, $M_7 = h\left(M_2 \| M_5 \| M_6 \| TSK \| SK_{u_i} \| T_2 \| GID\right)$, $M_8 = h\left(k_d \| M_2 \| SK_{SD_j}\right) \oplus TSK$, $M_9 = h\left(M_8 \| TSK \| SK_{SD_j}\right) \oplus \left(PID_i \| GID\right)$, $M_{10} = h\left(M_2 \| M_8 \| M_9 \| TSK \| SK_{SD_j} \| PID_i \| GID\right)$. Generates two messages $msg_2 = \left\{T_2, M_5, M_6, M_7\right\}$, $msg_3 = \left\{T_2, M_2, M_8, M_9, M_{10}, \gamma_d\right\}$. Where msg_3 is the time-limited token, and gateway sends msg_2, msg_3 to user.

(3) After receiving the message, user first opens the message msg_2 and checks the validity of timestamp by $T_2' - T_2 \leq \nabla T$, where T_2' is the time when the user receives msg_2, msg_3. If timestamp is valid, users computes $TSK = h\left(SK_{u_i} \parallel M_2\right) \oplus M_5$, $GID = h\left(M_5 \parallel TSK \parallel SK_{u_i}\right) \oplus M_6$, $M_7' = h\left(M_2 \parallel M_5 \parallel M_6 \parallel TSK \parallel SK_{u_i} \parallel T_2 \parallel GID\right)$. and verify $M_7' \stackrel{?}{=} M_7$. If verification fails, returned the error termination symbol $\perp$. Otherwise, user selects the current timestamp T_3, randomly selects $r_g \in z_q^*$, computes $M_{11} = h\left(M_2 \parallel TSK\right) \oplus r_g$, $M_{12} = h\left(M_2 \parallel M_{11} \parallel r_g \parallel TSK \parallel PID_i \parallel T_3\right)$, and generates the message $msg_4 = \left\{T_3, M_{11}, M_{12}, PID_i\right\}$. User broadcasts the received time-limited token $msg_3 = \left\{T_3, M_2, M_8, M_9, M_{10}, \gamma_d\right\}$ from the gateway and the generated message $msg_4 = \left\{T_3, M_{11}, M_{12}, PID_i\right\}$ to the industrial devices in the area.

(4) After industrial device in the region receives the message, it first opens the message msg_4 and checks the validity of timestamp by $T_3' - T_3 \leq \nabla T$, where T_3' is the time when industrial device receives msg_3, msg_4. If timestamp is valid, industrial device meets the authentication conditions opens the time-limited token message msg_3 and uses its own private key to obtain the secret value k_d by calculating $k_d = \gamma_d mod x_j$. Next, compute $TSK = h\left(k_d \parallel M_2 \parallel SK_{SD_j}\right) \oplus M_8$, $\left(PID_i \parallel GID\right) = h\left(M_8 \parallel TSK \parallel SK_{SD_j}\right) \oplus M_9$, and $M_{10}' = h\left(M_2 \parallel M_8 \parallel M_9 \parallel TSK \parallel SK_{SD_j} \parallel PID_i \parallel GID\right)$ and verify $M_{10}' \stackrel{?}{=} M_{10}$. If verification fails, returned error termination symbol $\perp$. Otherwise, compute $r_g = h\left(M_2 \parallel TSK\right) \oplus M_{11}$, $M_{12}' = h\left(M_2 \parallel M_{11} \parallel r_g \parallel TSK \parallel PID_i \parallel T_3\right)$ and verify $M_{12}' \stackrel{?}{=} M_{12}$. If verification passes, the industrial device authenticates both the user and the gateway. Industrial device picks the current timestamp T_4, randomly selects $r_j \in z_q^*$, computes $TSK^* = h\left(r_j \parallel TSK\right)$, $M_{13} = h\left(M_2 \parallel TSK\right) \oplus TSK^*$, $M_{14} = r_j \cdot M_2$, $M_{15} = r_j \cdot P$, $M_{16} = h\left(M_{13} \parallel M_{15} \parallel TSK \parallel TSK^* \parallel T_4\right)$. Computes the session key $SK = h\left(M_2 \parallel M_{13} \parallel M_{14} \parallel TSK^*\right)$ with the user u_i. Industrial device generates message $msg_5 = \left\{T_4, M_2, M_{13}, M_{15}, M_{16}\right\}$ and sends message msg_5 to the user u_i.

(5) After receiving the message, user opens the message msg_5 and checks the validity of timestamp by $T_4' - T_4 \leq \nabla T$. If timestamp is valid, computes $TSK^* = h\left(M_2 \parallel TSK\right) \oplus M_{13}$, $M_{16}' = h\left(M_{13} \parallel M_{15} \parallel TSK \parallel TSK^* \parallel T_4\right)$, and verify $M_{16}' \stackrel{?}{=} M_{16}$. If verification fails, returned the error termination symbol $\perp$. Otherwise, user computes the session key $SK = h\left(M_2 \parallel M_{13} \parallel r_i^* \cdot M_{15} \parallel TSK^*\right)$. At this point, user and industrial device have completed mutual authentication and agreement a session key for subsequent communication.

4.6. Time-limited token update

As the production tasks progress, the industrial devices that the user needs to access may change in real-time. Compared to the current list of accessed devices, the user may need to access new devices or no longer need access to certain devices. In this case, the user sends a new batch authentication request, which includes the identity list of the newly requested industrial devices, denoted as $SID_j' = \left\{SID_0, \ldots, SID_n\right\}$, to the gateway. If the list contains new industrial device identities, the gateway computes $Q' = Q + var_j$. If certain devices are not included in the new identity request list, the gateway computes $Q' = Q - var_j$. Then gateway randomly selects a new secret value $k_d' \in z_q^*$ and computes $\gamma_d' = k_d' \times Q'$ to complete the update of the time-limited token. After the update completed, the deleted industrial device will not be able to recover the secret value k_d' by modulo operation because its private key no longer exists in Q'. Similarly, the new industrial devices added to the list can use their private keys to recover the new secret value k_d' through a modulo operation, and then complete the subsequent authentication and key agreement process.

(Note: var_j represents multiple industrial devices. For example, when the identity list includes newly added industrial devices SID_3, SID_5, SID_7, then $Q' = Q + \left(var_3 + var_5 + var_7\right)$. If devices SID_4, SID_8 are not in the new identity request list, then $Q' = Q - \left(var_4 + var_8\right)$.)

4.7. Malicious user tracking

When gateway detects the malicious behavior of user PID_i, gateway can recover its real identity ID_i by compute $ID_i = PID_i \oplus h(s \cdot M_2)$, then add its real identity to the revocation list, and submit the real identity ID_i of the malicious user to TA.

5. Security analysis

This section provides a security proof and analysis of the proposed batch authentication and key agreement scheme. First, the security of the scheme is formally proven using the Real-Or-Random (ROR) model [37]. Next, heuristic analysis is employed to demonstrate the scheme's resilience against various protocol attacks. Finally, the advanced protocol verification tool Scyther is used to validate the security of the proposed scheme.

The ROR model is widely used in the formal security proofs of AKA schemes. Formal security proofs can characterize the capabilities of adversaries in both passive and active attacks, demonstrating that the scheme can provide secure authentication and semantic security. However, formal security proofs cannot fully capture the attack capabilities of adversaries in real-world environments.

Heuristic security analysis can adequately consider the attack capabilities of adversaries in real-world environments, as well as the security requirements of the scheme. Therefore, heuristic analysis is often used in conjunction with formal security proofs to jointly assess the security of the scheme. However, heuristic analysis heavily relies on the experience of the analyst, which introduces the risk of human oversight in the analysis.

The Scyther tool is widely used for the analysis of authentication schemes, providing a range of statements to test the security properties of the schemes. Secret statements are used to assess key security, while authentication statements primarily evaluate the scheme's resistance to various attacks, such as replay attacks, impersonation attacks, and man-in-the-middle attacks. However, similar to formal security proofs, the Scyther tool cannot fully capture the attack capabilities of adversaries in real-world environments.

In summary, the three analysis methods each have their own advantages and disadvantages. Security proofs and the Scyther tool represent formal analysis approaches, which effectively mitigate the analytical errors introduced by human factors in heuristic analysis. However, formal methods cannot fully capture the capabilities of attackers and the security properties that the scheme must satisfy, whereas heuristic analysis can effectively address this limitation. It is well known that designing a secure AKA scheme and proving its security is a complex task. Therefore, we employ these three mainstream approaches to analyze and prove the security of the scheme proposed in this paper, aiming to complement each method's strengths and weaknesses to minimize security oversights.

5.1. Formal security proof

- **Security model**

Before proving the security of the scheme in this paper, the definition of each basic primitive in the ROR model is first given [37]:

(1) **Participants**: In the scheme of this paper, there are three participants, namely user, gateway, and industrial device. During the protocol execution, they are instantiated as U_i, SD_j, and GW respectively. Let U_i^a denote the instance a of user U_i, SD_j^b denote the instance b of industrial device SD_j, GW^c denote the instance c of gateway GW.
(2) **partnering**: Let sid denote the session identifier, if there is a partnership between instance U_i^a and instance SD_j^b, then they satisfy the following three conditions: they are both in the accepted state; they share the same session identifier sid; they are partners with each other.
(3) **Freshness**: Freshness is a fundamental concept that defines protocol security. Freshness means that instances U_i^a and SD_j^b are Freshness if a session key SK has been agreement between user U_i and industrial device SD_j and SK has not been compromised to an adversary.

The DY model defines that an adversary can take full control of the open channel and eavesdrop to obtain public parameters on the open channel. In addition, the adversary can modify or replay messages exchanged in the open channel and forge new messages to spoof other instances. Adversary $\mathcal{A}$ can perform the following queries:

(1) $Hash(\cdot)$: When $\mathcal{A}$ performs a hash query, it returns a random value of fixed length.
(2) $Execute\left(U_i^a, SD_j^b, GW^c\right)$: The query simulates eavesdropping attack. $\mathcal{A}$ can obtain all the messages transmitted by U_i, SD_j, GW on the open channel by monitoring.
(3) $Send\left(U_i^a/SD_j^b/GW^c, m\right)$: The query simulates an active attack. $\mathcal{A}$ sends message m to instance $U_i^a/SD_j^b/GW^c$. If m is valid, the instance responds and replies to the message; otherwise, the instance ignores this query.
(4) $Reveal\left(U_i^a, SD_j^b\right)$: This query simulates the disclosure of session key. When $\mathcal{A}$ executes this query, the session key SK established between instances U_i^a, SD_j^b will revealed to the adversary.
(5) $Corrupt\left(U_i^a, SD_j^b\right)$: This query simulates the ability of an adversary to corrupt an instance. When $\mathcal{A}$ executes this query, $\mathcal{A}$ has access to all the secret parameters of the participating instances.
(6) $Test\left(U_i^a, SD_j^b\right)$: This query simulates the semantic security of the session key SK between instances U_i^a, SD_j^b. When $\mathcal{A}$ executes this query, the simulator flips a random coin $b \in \{0,1\}$. If $b == 1$, the simulator returns to $\mathcal{A}$ the session key; if $b == 0$, it returns a random string of the same length as the session key.

Semantic security[38]: In the ROR model, the goal of the adversary $\mathcal{A}$ is to distinguish whether a real session key or a random number is returned by the $Test$ query. $\mathcal{A}$ can query the instance U_i^a, SD_j^b with the PPT number of $Execute$, $Send$, $Reveal$, $Corrupt$, $Test$, when the query is finished, $\mathcal{A}$ outputs a bit b', and only when $b' = b$, $\mathcal{A}$ wins this game. Let $Succ$ denote that $\mathcal{A}$ wins the game, and let $\mathcal{P}$ denote the AKA scheme constructed in this paper, then the advantage of $\mathcal{A}$ in breaking the semantic security of $\mathcal{P}$ is:

$$Adv_{\mathcal{P}}^{aka}(\mathcal{A}) = 2Pr[Succ\mathcal{A}] - 1. \tag{4}$$

- **Security proof**

Theorem 1. *Let $\mathcal{A}$ denote the adversary that breaks the scheme $\mathcal{P}$ in PPT time t and D be a cipher space that obeys the distribution of Zipf's law [39]. q_h, q_s, q_e denote the number of $Hash$ queries, $Send$ queries, $Execute$ queries respectively. $|Hash|$ and l represent the output space of the hash function $h(\cdot)$ and the output length of the random prediction machine. $Adv_{\mathcal{A}}^{ECCDH}(t)$ denotes the advantage of adversary $\mathcal{A}$ solving $ECCDH$ difficult problem in PPT time. Then the advantage of adversary $\mathcal{A}$ breaking the security of the scheme $\mathcal{P}$ session key in PPT time is:*

$$Adv_{\mathcal{A}}^{\mathcal{P}}(t) \le \frac{(q_s+q_e)^2}{2^P} + \frac{q_h^2+2(2q_h+q_s)^2+3(q_h+q_s)^2}{2^l} + q_h\left((q_s+q_e)^2+1\right)\cdot Adv_{\mathcal{A}}^{ECCDH}(t) \tag{5}$$

Define six different games to prove the security of the scheme, denoted $G_0 - G_5$. The games start at G_0 and end at G_5. In these games, the adversary's advantage is gradually reduced to zero. $Succ_i$ and $Pr\left[Succ_i\right]$ respectively denote the event and probability that $\mathcal{A}$ makes a successful guess in game $G_i, i \in [0,5]$.

Game G_0: Game G_0 simulates the real attack of adversary $\mathcal{A}$ on the proposed scheme $\mathcal{P}$ under the ROR model, which can be obtained according to the definition of semantic security:

$$Adv_{\mathcal{A}}^{\mathcal{P}}(t) = 2Pr\left[Succ_0\right] - 1. \tag{6}$$

Game G_1: Game G_1 simulates eavesdropping attacks. Compared with game G_0, $Execute\left(U_i^a, SD_j^b, GW^c\right)$ query is added to G_1. $\mathcal{A}$ monitoring the communication information $msg_1 = \{PID_i, T_1, M_2, M_3, M_4\}$, $msg_2 = \{T_2, M_5, M_6, M_7\}$, $msg_3 = \{T_2, M_2, M_8, M_8, M_{10}, \gamma_d\}$, $msg_4 = \{T_3, M_{11}, M_{12}, PID_i\}$, $msg_5 = \{T_4, M_2, M_{13}, M_{15}, M_{16}\}$ between the three participants through $Execute\left(U_i^a, SD_j^b, GW^c\right)$ query, and finally determines whether the value of the $Test$ query output is a real session key or a random string. In the scheme of this paper, the process of computing the session key $SK = h\left(M_2 \parallel M_{13} \parallel M_{14} \parallel TSK^*\right)$ contains the secret values r_j and TSK^*. Therefore, it is obvious that $\mathcal{A}$ cannot compute SK between user and industrial device by monitoring to the message. Compared with G_0, monitoring message cannot increase the probability of $\mathcal{A}$ winning the game G_1, which can be obtained:

$$\left|Pr\left[Succ_1\right] - Pr\left[Succ_0\right]\right| = 0. \tag{7}$$

Game G_2: Game G_2 describes the ability of adversary $\mathcal{A}$ to attack actively. Compared with G_1, adversary $\mathcal{A}$ in G_2 will actively join the session by executing $Send$ query and $Hash$ query, and try to forge legitimate messages to deceive the scheme participating entities. $\mathcal{A}$ has the possibility to construct a valid message only when a collision is detected, which in turn destroys the semantic security of $\mathcal{P}$. The scheme in this paper has two types of collisions in the phase of authentication and key agreement:

(1) The hash function h collides on output, and its maximum probability is: $\frac{q_h^2}{2^l}$.
(2) The random number in message $\left(msg_1, msg_2, msg_3, msg_4, msg_5\right)$ experiences a collision, and its maximum probability is: $\frac{(q_s+q_e)^2}{2p}$.

Therefore, unless a collision occurs, G_2 and G_1 are indistinguishable. According to the birthday paradox, we have:

$$\left|Pr\left[Succ_2\right] - Pr\left[Succ_1\right]\right| \le \frac{q_h^2}{2^l} + \frac{\left(q_s+q_e\right)^2}{2p} \tag{8}$$

Game G_3: In Game G3, $\mathcal{A}$ tries to forge a valid message that can be verified by guessing the secret parameter. Specifically, $\mathcal{A}$ tries to forge the following message:

(1) The adversary successfully forged the message msg_1. In this case, $\mathcal{A}$ needs to make $Hash$ query to compute msg_1. Therefore, $\mathcal{A}$ make the following query: $\{(M_2 \parallel * \parallel SID_j), (PID_i \parallel * \parallel * \parallel T_1 \parallel SID_j), M_4\}$. And the probability of success in this event is denoted as: $\frac{(q_h+q_s)^2}{2^l}$.
(2) The adversary successfully forged the message msg_2. Similar to above, $\mathcal{A}$ needs to make $Hash$ query to compute msg_2. $\mathcal{A}$ make the following query: $\left\{\begin{array}{l}(* \parallel M_2 \parallel *), (M_5 \parallel * \parallel * \parallel GID), \\ (M_2 \parallel M_5 \parallel M_6 \parallel * \parallel * \parallel T_2 \parallel GID), M_7\end{array}\right\}$. And the probability of success in this event is denoted as: $\frac{(2q_h+q_s)^2}{2^l}$.

(3) The adversary successfully forged the message msg_3. $\mathcal{A}$ needs to make the following $Hash$ query:
$\left\{ \begin{array}{l} (*\| M_2 \|*\|*), (M_8 \|*\|*\| PID_i \| GID), \\ (M_2 \| M_8 \| M_9 \|*\|*\| PID_i \| GID), M_{10} \end{array} \right\}$. And the probability of success in this event is denoted as: $\frac{(2q_h+q_s)^2}{2^l}$.

(4) The adversary successfully forged the message msg_4. $\mathcal{A}$ needs to make the following $Hash$ query:
$\{(M_2 \|*\|*), (M_2 \| M_{11} \|*\|*\| PID_i \| T_3), M_{12}\}$. And the probability of success in this event is denoted as: $\frac{(q_h+q_s)^2}{2^l}$.

(5) The adversary successfully forged the message msg_5. $\mathcal{A}$ needs to make the following $Hash$ query:
$\{(M_2 \|*\|*), (M_2 \| M_{13} \| M_{15} \|*\|*\| T_4), M_{16}\}$. And the probability of success in this event is denoted as: $\frac{(q_h+q_s)^2}{2^l}$.

Thus, unless $\mathcal{A}$ successfully forges all of the above messages, G_3 is indistinguishable from G_2, we have:

$$\left|Pr\left[Succ_3\right] - Pr\left[Succ_2\right]\right| \leq \frac{2\left(2q_h + q_s\right)^2 + 3\left(q_h + q_s\right)^2}{2^l} \tag{9}$$

Game G_4: In game G_4, $\mathcal{A}$ tries to compute the session key SK. Since the session key is constructed based on the ECCDH problem, the difficulty for $\mathcal{A}$ to compute the session key in PPT time is equivalent to solving the ECCDH problem in PPT time. $\mathcal{A}$ chooses the ECCDH tuple $(r_i P, r_j P)$ with probability $\frac{1}{q_h}$, thus we have:

$$\left|Pr\left[Succ_4\right] - Pr\left[Succ_3\right]\right| \leq q_h \cdot Adv_{\mathcal{A}}^{ECCDH}(t) \tag{10}$$

Game G_5: The game G_5 considers the forward security of scheme $\mathcal{P}$. In this game, $\mathcal{A}$ can execute $Send$ queries as well as $Corrupt$ queries to obtain the long-term secret values stored by the user and the industrial device. The probability that tuple $(r_i P, r_j P)$ in a session is $\frac{1}{(q_s+q_e)^2}$, thus we have:

$$\left|Pr\left[Succ_5\right] - Pr\left[Succ_4\right]\right| \leq q_h\left(q_s + q_e\right)^2 \cdot Adv_{\mathcal{A}}^{ECCDH}(t) \tag{11}$$

Based on Eqs. (6)–(11), we obtained the result:

$$\begin{aligned} Adv_{\mathcal{A}}^{\mathcal{P}}(t) \leq & \frac{(q_s+q_e)^2}{2^P} + \frac{q_h^2+2(2q_h+q_s)^2+3(q_h+q_s)^2}{2^l} \\ & + q_h\left(\left(q_s + q_e\right)^2 + 1\right) \cdot Adv_{\mathcal{A}}^{ECCDH}(t) \end{aligned} \tag{12}$$

The above proof procedure implies that after all the prediction machines have been simulated, $\mathcal{A}$ does not gain any additional advantage to win the game. Therefore, the scheme proposed in this paper is safe under the ROR model.

5.2. Heuristic security analysis

(1) Mutual authentication: In the scheme proposed in this paper, all participating entities have completed mutual authentication. The details are analyzed as follows:
Authentication between the user and the gateway: the gateway accomplishes the direct authentication of the user through M_4 in message msg_1. Because message M_4 contains the session key SK_{u_i} between the gateway and user and (M_1, M_2) is a pair of plain ciphertexts constructed by the public key algorithm, other users are unable to forge message M_4. Similarly, user accomplishes direct authentication to the gateway via M_7 in message msg_2, since message M_7 also contains the session key SK_{u_i} and the secret value TSK cryptographically protected by SK_{u_i}.
Authentication between the gateway and the industrial device: the industrial device authenticates the gateway directly by means of M_{10} in message msg_3, since message M_{10} contains the session key SK_{SD_j} between the gateway and the industrial device. The gateway indirectly authenticates the industrial device through γ_d in the message msg_3. This is because based on the CRT, only the industrial device that meets the authentication conditions can recover the secret value k_d based on its own private key as well as γ_d to complete the subsequent authentication.
Authentication between the user and the industrial device: the industrial device directly authenticates the user via M_{12} in message msg_4, because message M_{12} contains the secret value TSK. similarly, the user directly authenticates the industrial device via M_{16} in message msg_5.

(2) Session key agreement: Session key $SK = h(M_2 \| M_{13} \| M_{14} \| TSK^*)$ is agreement between the user and the industrial device, which contains the secret values TSK^* and M_{14}. Except for both parties of the session, no third party can obtain the session key.

(3) User anonymity: Users use pseudonym $PID_i = ID_i \oplus h(r_i^* \cdot PK)$ to communicate, effectively protect their identity ID_i, realize user anonymity. At the same time, when the user has violated the law, the gateway can recover the user's real identity ID_i through the $ID_i = PID_i \oplus h(s \cdot M_2)$ to complete the tracking. Therefore, the scheme in this paper guarantees the anonymity of the user while realizing the conditional privacy protection of the user.

(4) Forward security: forward security means that the compromise of the current system does not affect the security of previous sessions. Assuming that all users' long-term secret values are compromised, the attacker obtains the message M_2, M_{13} through passive attack listening, and the session key is computed as $SK = h(M_2 \| M_{13} \| M_{14} \| TSK^*)$. Therefore, if the adversary wants to calculate the session key SK, he still needs to know the secret value M_{14}, TSK^*, which is never transmitted in the open channel. TSK^* only both sides of the communication know that the adversary needs to solve the ECCDH problem if he wants to calculate M_{14} through M_2, M_{13}, but the ECCDH problem is unsolvable in PPT time. Therefore, the proposed scheme in this paper satisfies forward security.

(5) Resistance to replay attacks : In the scheme of this paper, timestamps and random numbers are used to resist replay attacks. Even if an adversary can intercept the communication messages in the open channel and replay them, the replayed messages cannot be verified due to the presence of timestamps and random numbers.

(6) Resistant to impersonation attack:
Resistance to user impersonation attack: To successfully impersonation as a user, adversary needs to construct an authenticated message $msg_1 = \{PID_i, T_1, M_2, M_3, M_4\}$. The construction of authentication message M_4 requires a long-term session key SK_{u_i} between the user and the gateway, which is unavailable to the adversary, and thus the adversary is not able to forge an authenticated message, so the proposed scheme is resistant to user impersonation attack.
Resistance to gateway impersonation attack: To successfully impersonation as gateway, adversary needs to construct authentication messages $msg_2 = \{T_2, M_5, M_6, M_7\}$, $msg_3 = \{T_2, M_2, M_8, M_8, M_{10}, \gamma_d\}$. Similar to the above, constructing authentication messages M_7, M_{10} requires a long term session key SK_{u_i}, SK_{SD_j}, so the adversary is unable to construct valid authentication messages, and the proposed scheme is resistant to gateway impersonation attack.
Resistance to industrial device impersonation attack: To successfully impersonation as an industrial device, adversary needs to construct an authentication message $msg_5 = \{T_4, M_2, M_{13}, M_{15}, M_{16}\}$, where the construction of the authentication message M_{16} requires the secret values TSK and TSK^*. TSK^* is computed from TSK, which requires secret values k_d, SK_{SD_j}. Therefore, the adversary cannot construct a valid authentication message, and the proposed scheme is resistant to industrial device impersonation attacks.

(7) Resisting privileged internal attacks: During the user registration process, the user sends the registration request parameters $\{UPW_i \oplus a, ID_i\}$ to the TA, where UPW_i, a_i, ID_i is the pseudo-password, the random number, and the user's identity, respectively. Due to the randomness of the random number and the unidirectionality of the hash function, it is difficult for the privileged adversary inside the TA to recover the user's real password PW_i based on the registration parameters, and thus the proposed scheme in this paper can resist the privileged internal attack.

(8) Resistance to man-in-the-middle attack: adversary can monitor to obtain messages $msg_1, msg_2, msg_3, msg_4, msg_5$ transmitted in the open channel and try to spoof U_i, GW, SD_j by modifying these messages. However, for an adversary to generate a legitimate message msg_1, it needs to obtain a random secret value r_i^* and a long-term secret value SK_{u_i}. Therefore, the adversary cannot generate a legitimate message msg_1. Similarly, an adversary cannot generate a legitimate message $msg_2, msg_3, msg_4, msg_5$. Therefore, the scheme proposed in this paper is resistant to man-in-the-middle attacks.

(9) Unlinkability: In the scheme proposed in this paper, the user communicates using a temporary pseudonym, and the identity information of the industrial devices is not transmitted over the public channel. All messages transmitted over the public channel are encrypted using random numbers, timestamps, or secret values. Due to the randomness of the random numbers and timestamps, an adversary cannot distinguish whether two different messages originate from the same entity. Therefore, the proposed scheme ensures unlinkability.

5.3. Verification based on scyther tool

This section uses the protocol verification tool Scyther [40] to validate the security of the proposed scheme. Scyther is widely used for the security verification and analysis of protocols. It employs a black-box approach, allowing users to evaluate whether the protocol meets the declared security goals and properties from their perspective [41]. Scyther models the roles in a protocol and their message sending and receiving behaviors using the SPDL language. Scyther supports nine common adversary models, including DY, CK, and eCK, and verifies the security of the protocol based on these models, analyzing whether the protocol has any security vulnerabilities.

Scyther proposed a set of statements to test the security properties of a protocol, including the secret statement $Secret$, and several verification statements $Alive$, $Weakagree$, $Niagree$, $Nisynch$[42]. Secret statements are mainly used to test the confidentiality of an identity or keys. Authentication statements are used to check for the presence of various attacks, such as replay attacks, impersonation attacks, and man-in-the-middle attacks. This section analyzes the security of the scheme in this paper using the standard DY model, which defines that an adversary can monitor, steal, replay or even modify the information transmitted in the open channel.

The results of this paper scheme verified using scyther tool are shown in Fig. 5. For the authentication and key agreement phase of this paper's scheme the tripartite participants user, gateway, and industrial device are defined as roles UI, GW, and SDJ respectively. The information sent and received by each role during the authentication and key agreement phases is modeled using the SPDL language, and the security and authentication statements for each role are verified. For example, for the role UI, there are four secret statements and four authentication statements. Where Key represents the session key between the UI and the SDJ. $sk(UI)$ represents the private key of the UI. $k(UI, GW)$ represents the long-term session key between the UI and the GW. The authentication statement, on the other hand, is to verify the security features that the scheme has. According to the simulation results Fig. 5 shows that the scheme proposed in this paper satisfies all the above declared security features. scyther tool does not find any attack on this paper's scheme under DY model.

Fig. 5. Formal verification results under the tset of scyther tool.

6. Performance analysis

This section provides a comparative analysis of the proposed scheme with existing scheme [13,14,23–25], in terms of security and functional features, computational overhead, and communication overhead. The compared schemes are all recently proposed AKA schemes for the IIoT or the Vehicular Networks (a specific IoT application). Among them, the schemes proposed in [13,14,23,24] are designed for multi-devices communication scenarios with batch processing capabilities, while the scheme in [25] considers the issue of gateway lightweighting in IoT environments. In the comparison of security and functional features, the ability of each scheme to resist various protocol attacks is evaluated, including unlinkability, forward security, and resistance to replay attacks. Additionally, the functional features met by each scheme are compared, such as user anonymity, suitability for multi-device communication scenarios, and gateway lightweight. The computational and communication overhead section compares the computational and communication costs of each scheme in the context of multi-device communication. These factors are essential criteria for assessing whether a scheme can be safely and efficiently applied in real-world IIoT environments.

6.1. Comparison of security and functional features

Firstly, we compare the security and functional features of the schemes, with the results shown in Table 2. Upon analysis, only the proposed scheme in this paper meets all 13 security and functional requirements. Although Wang et al. [25] scheme addresses the lightweight

Table 2
Comparison of security and functional features.

	Scheme [24]	Scheme [23]	Scheme [13]	Scheme [25]	Scheme [14]	Our scheme
SG_1	✓	✓	✓	✓	✓	✓
SG_2	✓	✓	✓	✓	✓	✓
SG_3	✓	✓	✓	✓	✓	✓
SG_4	N/A	✓	N/A	N/A	✓	✓
SG_5	✓	✓	✓	✓	✓	✓
SG_6	✗	✓	✗	✓	✓	✓
SG_7	✓	✓	✓	✓	✓	✓
SG_8	✓	✓	✓	✓	✓	✓
SG_9	✓	✓	✗	✓	✓	✓
SG_{10}	✓	✗	✓	✓	✓	✓
SG_{11}	✓	N/A	✓	✓	N/A	✓
SG_{12}	✓	✓	✓	N/A	✓	✓
SG_{13}	N/A	N/A	N/A	✓	N/A	✓

SG_1: Mutual authentication. SG_2: Key agreement. SG_3: User anonymity. SG_4: Malicious user tracking. SG_5: Unlinkability. SG_6: Forward security. SG_7: Resistant to replay attacks. SG_8: Resistant to impersonation attack. SG_9: Resistant privileged internal attack. SG_{10}: Resistance to man-in-the-middle attack. SG_{11}: Terminal device update. SG_{12}: Suitable for Multi-Device Scenarios. SG_{13}: Gateway Lightweighting. N/A Means not consider the functional feature.

Table 3
Computation time for cryptographic operations (Milliseconds).

Operations	T_{ecm}	T_{eca}	T_m	T_{se}	T_{sd}	T_h
Computation time	0.7587	0.0048	0.0072	0.0114	0.0122	0.0015

nature of the gateway, it does not consider its application in multi-devices communication scenarios and is therefore unsuitable for the IIoT environment. The other schemes [13,14,23,24], while considering multi-devices communication scenarios, still present certain security and usability issues. The schemes proposed by Vinoth et al. [24] and Yang et al. [13] lack forward security and do not consider the functional feature of malicious user tracking; additionally, Yang et al. [13] scheme is vulnerable to privileged insider attacks. Cui et al. [23] scheme fails to resist man-in-the-middle attacks and does not account for the functional feature of terminal device updates. The scheme by Zhang et al. [14] offers high security but does not consider the terminal device update feature, making it ineffective in scenarios where the user's accessed devices frequently change. Moreover, none of the aforementioned AKA schemes for multi-terminal devices [13,14,23,24] take the lightweight nature of the gateway into account. In summary, only the proposed scheme in this paper satisfies all 13 security and functional requirements, making it more suitable for the IIoT environment where users frequently communicate with multiple industrial devices.

6.2. Comparison of computation overhead

This section compares the computational overhead of the proposed scheme with the comparison schemes [13,14,23–25]. Since the registration or authorization login phase is performed only once throughout the entire process, this subsection focuses solely on the computational overhead during the authentication and key agreement phase. Additionally, considering that users in the IIoT frequently communicate with multiple industrial devices, the comparison here will emphasize the computational overhead in multi-device communication scenarios to better reflect real-world IIoT environments.

To achieve a 128-bit security level, construct an additive cyclic group G generated by an elliptic curve $E : y^2 = x^3 + ax + b \bmod p$, where the order of the group is p and the generator is q. Here, p and q are 256-bit prime numbers. Experiments were conducted on a personal computer to measure the computational overhead of cryptographic operations based on the MIRACL library [43]. The experimental environment was configured with a 12th Gen Intel(R) Core(TM) i5-1235U @1.30 GHz processor, 16 GB of RAM, and the Ubuntu 22.04 operating system. Each cryptographic operation was executed 1,000 times using the MIRACL library to obtain the average computation time, thereby reducing measurement errors. The average computation times for various cryptographic operations are presented in Table 3. Where, T_{ecm}, T_{eca}, T_m, T_{se}, T_{sd}, T_h represent the computation times for various operations: point multiplication in group G, point addition in group G, multiplication in group Z_q^*, symmetric encryption (AES-CBC), symmetric decryption (AES-CBC), and hash function operations, respectively. As the computational overhead of the XOR operation is negligible, it is not considered when comparing computational costs. In addition, according to the work of Wang et al. [25], the calculation time of fuzzy biometric extraction is $T_b \approx T_{ecm}$.

- **Computational Overhead in Multi-Device Communication Scenarios**

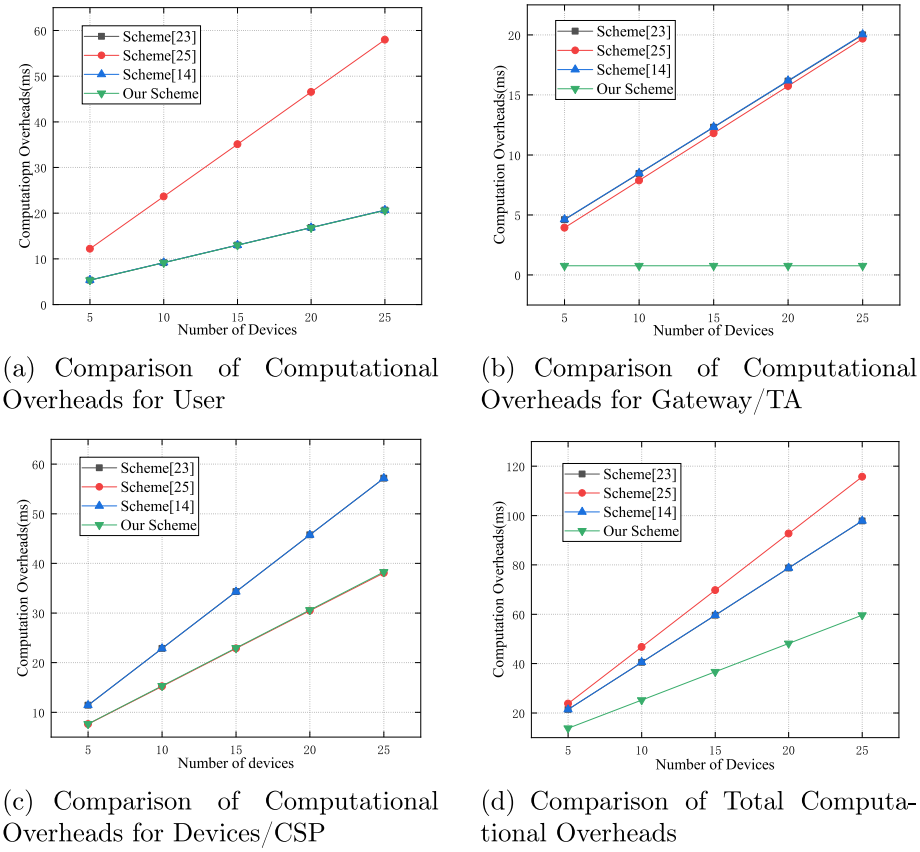
In the proposed scheme, three main entities are involved during the authentication and key agreement phase: the user, the gateway, and the industrial devices. During this phase, when user intends to authenticate and negotiate keys with n industrial devices, they first send a batch authentication request message msg_1 to the gateway. Upon receiving the gateway's response msg_2, msg_3, the user must perform the necessary computations and broadcast message msg_3, msg_4 to the n industrial devices. At this point, the computational overhead for the user is denoted as $8T_h + 2T_{ecm}$. Finally, in the key agreement phase, the user needs to process the responses msg_5 from the n industrial devices simultaneously to compute different session keys SK. Therefore, the computational cost for the user in the key agreement phase is $3nT_h + nT_{ecm}$. The total computational overhead for the user during the entire authentication and key agreement process in the proposed scheme is $(3n + 8)T_h + (n + 2)T_{ecm}$. In the proposed scheme, due to the application of the Chinese Remainder Theorem and time-limited tokens, the gateway only needs to handle the batch authentication request message from the user without interacting directly with the industrial devices. Consequently, the total computational overhead for the gateway is $10T_h + T_{ecm}$. Each industrial device, however, must process the authentication message from the user and compute the session key independently. Therefore, in a multi-device scenario, the computational overhead for n industrial devices is $(9T_h + 2T_{ecm})n$. The total computational overhead of the proposed scheme during the authentication and key agreement phase in a multi-device communication scenario is $(12n + 18)T_h + (3n + 3)T_{ecm}$. The computational overheads for the authentication and key agreement phase of other schemes in multi-terminal device communication scenarios are presented in Table 4, with the analysis method being the same as that of the proposed scheme, and thus not further elaborated here.

As shown in Table 4, Vinoth et al. [24] scheme, which is based on symmetric cryptography, and Yang et al. [13] scheme, which does not

Table 4
Computational overhead for each scheme in multi-device communication scenarios.

scheme	User/Vehicle	Gateway/TA	n Industrial device/Smart device/CSP	Total computation overhead
[23]	$(5n+3)T_h+(n+2)T_{ecm}$	$(7n+3)T_h+(n+1)T_{ecm}$	$(7T_h+3T_{ecm})n$	$(19n+6)T_h+(5n+3)T_{ecm}$
[24]	$9T_h+T_{sd}$	$6T_h+(2n+2)T_m$ $+2T_{se}+nT_{sd}$	$(4T_h+T_{se}+T_{sd})n$	$(4n+15)T_h+(2n+2)T_m$ $+(n+2)T_{se}+(2n+1)T_{sd}$
[13]	$(7+n)T_h+2T_{ecm}$ $+T_m+T_{se}+T_{sd}$	$(2n+9)T_h+T_{ecm}$ $+nT_m+2T_{se}+(n+1)T_{sd}$	$(6T_h+T_m+T_{se}+T_{sd})n$	$(9n+16)T_h+3T_{ecm}+(2n+1)T_m$ $+(n+3)T_{se}+(2n+2)T_{sd}$
[25]	$8nT_h+3nT_{ecm}+T_b$	$19nT_h+nT_{ecm}$	$4nT_h+2nT_{ecm}$	$31nT_h+6nT_{ecm}+T_b$
[14]	$(5n+4)T_h$ $+(n+2)T_{ecm}+T_{se}$	$(2+8n)T_h$ $+(1+n)T_{ecm}+T_{se}$	$7nT_h+3nT_{ecm}$	$(20n+6)T_h+(5n+3)T_{ecm}$ $+2T_{se}$
Our scheme	$(3n+8)T_h$ $+(n+2)T_{ecm}$	$10T_h+T_{ecm}$	$(9T_h+2T_{ecm})n$	$(12n+18)T_h$ $+(3n+3)T_{ecm}$

User/Vehicle denotes Uesr, Vehicle user in Vehicular Networks.
Gateway/TA denotes trusted entity.
Industrial Device/Smart Device/CSP denotes Industrial device, Smart Device in IOT, Cloud server in Vehicular Networks.



(a) Comparison of Computational Overheads for User

(b) Comparison of Computational Overheads for Gateway/TA

(c) Comparison of Computational Overheads for Devices/CSP

(d) Comparison of Total Computational Overheads

Fig. 6. The Comparisons of Computational overhead.

deploy public-key cryptographic operations on industrial devices, have lower computational overhead compared to the proposed scheme and other schemes based on public-key cryptography [13,14,25]. However, their schemes suffer from significant security deficiencies. It is well known that schemes solely based on symmetric cryptography cannot effectively ensure a high level of security. According to the work of Wang et al. [25], since these schemes do not deploy public-key operations on industrial devices, they fail to provide forward security.

To more clearly demonstrate the computational cost comparison between the proposed scheme and the other public-key cryptography-based schemes [14,23,25] in a multi-device communication scenario during the authentication and key agreement phase, we have plotted a graph (as shown in Fig. 6) illustrating the computational overheads of each entity and the total computational overheads as the number of devices increases. The results show that the total computational overhead of the proposed scheme in a multi-device communication scenario is lower than that of other compared schemes. The computational overhead at the user is close to the schemes proposed in [14,23], and better than the scheme in [25]. The computational overhead at the industrial device is close to the scheme proposed in [25], and better than the schemes in [14,23]. This is primarily because, to ensure forward security, the scheme requires at least two public key operations to be

Table 5
Comparison of communication overheads for each scheme.

Scheme	User/Vehicle	Gateway/TA	Industrial device/Smart device/CSP	Total communication overhead	Multi-device scenarios total communication overhead
[24]	100*byte*	216*byte*	52*byte*	368*byte*	$(184+184n)\,byte$
[23]	168*byte*	200*byte*	300*byte*	668*byte*	$(136+532n)\,byte$
[13]	116*byte*	172*byte*	52*byte*	340*byte*	$(168+172n)\,byte$
[25]	160*byte*	480*byte*	96*byte*	736*byte*	$(736n)byte$
[14]	112*byte*	164*byte*	268*byte*	544*byte*	$(112+432n)\,byte$
Our scheme	444*byte*	280*byte*	164*byte*	888*byte*	$(724+164n)\,byte$

deployed at the industrial device side. Both the proposed scheme and the scheme in [25] deploy two ECC point multiplications at the device side, while the schemes in [14,23] deploy three point multiplications. As the computational overhead of the scheme is mainly influenced by the number of point multiplications, the computational overhead at the industrial device in the proposed scheme is close to the scheme in [25]. Similarly, since point multiplication operations are deployed at the industrial device side to compute the session key, in order to ensure the secure negotiation of the session key and achieve a balance between security and efficiency, the proposed scheme deploys a certain amount of point multiplication operations at the user side. This results in the computational overhead at the user being similar to that of the schemes proposed in [14,23]. However, overall, the computational overhead at both the user and industrial device in the proposed scheme still meets the lightweight requirements. Furthermore, due to the use of the Chinese Remainder Theorem and time-limited tokens in the proposed scheme, the computational overhead at the gateway node remains constant regardless of the number of industrial devices accessed by the user in a multi-device communication scenario. Therefore, compared to other schemes, the proposed scheme has a significant advantage in gateway lightweighting, effectively avoiding the issue of gateway single-point failure, and is more suitable for IIoT environments where users frequently communicate with multiple devices.

Further, in practical applications, the computational overhead of hash operations is closely related to the byte length of the input data, and different hash operations in the scheme have different input data lengths. For instance, when calculating $r_i^* = h\left(r_i \parallel SK_{u_i} \parallel T_1\right)$, the input length of this hash operation is 68 bytes. This is because, to achieve 128-bit security, the elliptic curve parameter q has a length of 32 bytes, the hash function's output length is 32 bytes, and the length of the timestamp is 4 bytes. The analysis of the input data byte length for the other hash operations follows the same logic, which will not be reiterated here.

To more accurately and clearly evaluate the computational overhead of the proposed scheme, we fully implemented it using the Miracl library. The experimental platform used is the same as that employed for measuring the time of various cryptographic operations as described earlier. We set n=10, meaning that we assessed the computational overhead incurred by the user, gateway, and each industrial device during batch authentication and key agreement when the user communicates with 10 industrial devices. According to the experiment, the computational overhead at the user side during the batch authentication and key agreement phase is 8.5487 ms, the gateway's computational overhead is 0.7433 ms, and the computational overhead for each industrial device is 1.4625 ms. The experimental results show that when the user performs batch authentication and key agreement with multiple industrial devices, the computational overhead on the industrial devices and the gateway is lightweight. On the other hand, since the user needs to negotiate different session keys with each industrial device, the computational overhead on the user side is higher than that of the gateway and industrial devices. Overall, the computational overhead of the proposed scheme is acceptable for all communication entities in the IIoT environment.

6.3. Comparison of communication overhead

This section compares the communication overhead of the proposed scheme with the comparison schemes [13,14,23–25] during the authentication and key agreement phase. To achieve 128-bit security, the elliptic curve parameter q is chosen with a length of 32 bytes, making the elements in the group G 64 bytes long. It is assumed that the output length of the hash function, the length of the timestamp, the length of ciphertext for symmetric encryption/decryption, and the length of random numbers are 32 bytes, 4 bytes, 16 bytes, and 16 bytes, respectively.

The proposed scheme involves four rounds of communication during the authentication and key agreement phase, with the communication messages for each round as follows: msg_1, (msg_2, msg_3), (msg_3, msg_4), msg_5. $msg_1 = \{PID_i, T_1, M_2, M_3, M_4\}$, which PID_i, M_3, M_4 is the output of hash function, T_1 is timestamp, and M_2 belongs to group G. Therefore, the communication overhead of message msg_1 is $|msg_1| = (32+4+64+32+32) = 164bytes$. Similarly, the communication overheads of $msg_2, msg_3, msg_4, msg_5$ are $100bytes$, $180bytes$, $100bytes$, and $164bytes$, respectively.

In the multi-device communication scenario, due to the use of the Chinese Remainder Theorem and time-limited tokens, a user only needs to send three messages msg_1, msg_3, and msg_4 to access n industrial devices. Similarly, the gateway only needs to communicate with the user by sending two messages msg_2, and msg_3. However, since each of the n industrial devices needs to complete mutual authentication with the user and negotiate a distinct session key, the n devices must collectively send n messages msg_5. In the multi-device communication scenario, the total communication overhead of the proposed scheme during the authentication and key agreement phase is $(724+164n)\,byte$. The communication overhead of the other schemes [13,14,23–25] is shown in Table 5, with the analysis method being the same as that used for the proposed scheme and thus not elaborated further here. To provide a clear comparison of the communication overheads of each scheme in a multi-device scenario, we select $n = 25$. The results show that, when $n = 25$, the communication overheads for the respective schemes are 35.94kb, 26.11kb, 9.12kb, 8.56kb, and 21.20kb. In comparison, the communication overhead of the proposed scheme in this scenario is 8.71kb. Thus, the proposed scheme demonstrates a relatively low communication overhead compared to the other schemes, making it suitable for real-world IIoT environments.

7. Conclusion

This paper proposes a batch AKA scheme for the IIoT environment, designed based on elliptic curve cryptography combined with the Chinese Remainder Theorem and the concept of time-limited tokens. The scheme enables batch authentication between a user and multiple industrial devices and establishes distinct session keys for secure subsequent communications. It satisfies the lightweight requirements for the gateway and all entities, making it suitable for resource-constrained IIoT environments. The security of the proposed scheme is demonstrated through formal proofs, heuristic analysis, and verification using

the Scyther tool. Performance analysis indicates that, compared to existing schemes, the proposed scheme meets all specified security requirements with lower computational and communication overheads and shows a significant advantage in lightweight operation at the gateway node.

CRediT authorship contribution statement

Xiaohui Ding: Writing – review & editing, Writing – original draft, Formal analysis. **Jian Wang:** Writing – review & editing, Formal analysis. **Yongxuan Zhao:** Writing – review & editing. **Zhiqiang Zhang:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] S. Li, L.D. Xu, S. Zhao, The Internet of Things: a survey, Inf. Syst. Front. 17 (2015) 243–259.

[2] I. Zhou, I. Makhdoom, N. Shariati, M.A. Raza, R. Keshavarz, J. Lipman, M. Abolhasan, A. Jamalipour, Internet of Things 2.0: Concepts, applications, and future directions, IEEE Access 9 (2021) 70961–71012.

[3] S.H. Shah, I. Yaqoob, A survey: Internet of Things (IOT) technologies, applications and challenges, in: 2016 IEEE Smart Energy Grid Engineering, SEGE, IEEE, 2016, pp. 381–385.

[4] M.S. Azhdari, A. Barati, H. Barati, A cluster-based routing method with authentication capability in vehicular Ad Hoc networks (VANETs), J. Parallel Distrib. Comput. 169 (2022) 1–23.

[5] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, M. Gidlund, Industrial Internet of Things: Challenges, opportunities, and directions, IEEE Trans. Ind. Inform. 14 (11) (2018) 4724–4734.

[6] P.K. Malik, R. Sharma, R. Singh, A. Gehlot, S.C. Satapathy, W.S. Alnumay, D. Pelusi, U. Ghosh, J. Nayak, Industrial Internet of Things and its applications in industry 4.0: State of the art, Comput. Commun. 166 (2021) 125–139.

[7] W.Z. Khan, M. Rehman, H.M. Zangoti, M.K. Afzal, N. Armi, K. Salah, Industrial Internet of Things: Recent advances, enabling technologies and open challenges, Comput. Electr. Eng. 81 (2020) 106522.

[8] A.G. Mirsaraei, A. Barati, H. Barati, A secure three-factor authentication scheme for IoT environments, J. Parallel Distrib. Comput. 169 (2022) 87–105.

[9] L. Khajehzadeh, H. Barati, A. Barati, A lightweight authentication and authorization method in IoT-based medical care, Multimedia Tools Appl. (2024) 1–40.

[10] Y. Chen, F. Yin, S. Hu, L. Sun, Y. Li, B. Xing, L. Chen, B. Guo, ECC-based authenticated key agreement protocol for industrial control system, IEEE Internet Things J. 10 (6) (2022) 4688–4697.

[11] X. Li, J. Niu, M.Z.A. Bhuiyan, F. Wu, M. Karuppiah, S. Kumari, A robust ECC-based provable secure authentication protocol with privacy preserving for industrial Internet of Things, IEEE Trans. Ind. Inform. 14 (8) (2017) 3599–3609.

[12] J. Srinivas, A.K. Das, M. Wazid, A.V. Vasilakos, Designing secure user authentication protocol for big data collection in IoT-based intelligent transportation system, IEEE Internet Things J. 8 (9) (2020) 7727–7744.

[13] Y. Ming, P. Yang, H. Mahdikhani, R. Lu, A secure one-to-many authentication and key agreement scheme for industrial IoT, IEEE Syst. J. (2022).

[14] J. Zhang, H. Zhong, J. Cui, Y. Xu, L. Liu, SMAKA: Secure many-to-many authentication and key agreement scheme for vehicular networks, IEEE Trans. Inf. Forensics Secur. 16 (2020) 1810–1824.

[15] S. Mandal, S. Mohanty, B. Majhi, CL-AGKA: Certificateless authenticated group key agreement protocol for mobile networks, Wirel. Netw. 26 (4) (2020) 3011–3031.

[16] P. Xu, H. Wu, X. Tao, C. Wang, D. Chen, G. Nan, Anti-quantum certificateless group authentication for massive accessing IoT devices, IEEE Internet Things J. (2024).

[17] S. Wu, C. Hsu, Z. Xia, J. Zhang, D. Wu, Symmetric-bivariate-polynomial-based lightweight authenticated group key agreement for industrial Internet of Things, J. Internet Technol. 21 (7) (2020) 1969–1979.

[18] M. Zhang, J. Zhou, G. Zhang, M. Zou, M. Chen, EC-BAAS: Elliptic curve-based batch anonymous authentication scheme for Internet of Vehicles, J. Syst. Archit. 117 (2021) 102161.

[19] C. Pu, K.-K.R. Choo, A lightweight aggregate authentication protocol for Internet of Drones, in: 2024 IEEE 21st Consumer Communications & Networking Conference, CCNC, IEEE, 2024, pp. 143–151.

[20] W. Mao, P. Jiang, L. Zhu, Locally verifiable batch authentication in IoMT, IEEE Trans. Inf. Forensics Secur. (2023).

[21] H. Shen, T. Wang, J. Chen, Y. Tao, F. Chen, Blockchain-based batch authentication scheme for Internet of Vehicles, IEEE Trans. Veh. Technol. (2024).

[22] C. Maurya, V.K. Chaurasiya, Efficient anonymous batch authentication scheme with conditional privacy in the Internet of Vehicles (IoV) applications, IEEE Trans. Intell. Transp. Syst. 24 (9) (2023) 9670–9683.

[23] J. Cui, X. Zhang, H. Zhong, J. Zhang, L. Liu, Extensible conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment, IEEE Trans. Inf. Forensics Secur. 15 (2019) 1654–1667.

[24] R. Vinoth, L.J. Deborah, P. Vijayakumar, N. Kumar, Secure multifactor authenticated key agreement scheme for industrial IoT, IEEE Internet Things J. 8 (5) (2020) 3801–3811.

[25] C. Wang, D. Wang, Y. Duan, X. Tao, Secure and lightweight user authentication scheme for cloud-assisted Internet of Things, IEEE Trans. Inf. Forensics Secur. (2023).

[26] M.L. Das, Two-factor user authentication in wireless sensor networks, IEEE Trans. Wirel. Commun. 8 (3) (2009) 1086–1090.

[27] A. Barati, A. Movaghar, M. Sabaei, RDTP: Reliable data transport protocol in wireless sensor networks, Telecommun. Syst. 62 (2016) 611–623.

[28] P. Alimoradi, A. Barati, H. Barati, A hierarchical key management and authentication method for wireless sensor networks, Int. J. Commun. Syst. 35 (6) (2022) e5076.

[29] S.A. Khah, A. Barati, H. Barati, A dynamic and multi-level key management method in wireless sensor networks (WSNs), Comput. Netw. 236 (2023) 109997.

[30] C.-G. Ma, D. Wang, S.-D. Zhao, Security flaws in two improved remote user authentication schemes using smart cards, Int. J. Commun. Syst. 27 (10) (2014) 2215–2227.

[31] V.S. Miller, Use of elliptic curves in cryptography, in: Conference on the Theory and Application of Cryptographic Techniques, Springer, 1985, pp. 417–426.

[32] N. Koblitz, Elliptic curve cryptosystems, Math. Comp. 48 (177) (1987) 203–209.

[33] W. Diffie, M.E. Hellman, New directions in cryptography, in: Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman, 2022, pp. 365–390.

[34] J. Zhang, J. Cui, H. Zhong, Z. Chen, L. Liu, PA-CRT: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular Ad-Hoc networks, IEEE Trans. Dependable Secur. Comput. 18 (2) (2019) 722–735.

[35] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inform. Theory 29 (2) (1983) 198–208.

[36] B. Authentication, EAP-DDBA: Efficient anonymity proximity device discovery and batch authentication mechanism for massive D2D communication devices in 3GPP 5G HetNet, 2020.

[37] M. Abdalla, P.-A. Fouque, D. Pointcheval, Password-based authenticated key exchange in the three-party setting, in: Public Key Cryptography-PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, January 23-26, 2005. Proceedings 8, Springer, 2005, pp. 65–84.

[38] C.-C. Chang, H.-D. Le, A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks, IEEE Trans. Wirel. Commun. 15 (1) (2015) 357–366.

[39] D. Wang, H. Cheng, P. Wang, X. Huang, G. Jian, Zipf's law in passwords, IEEE Trans. Inf. Forensics Secur. 12 (11) (2017) 2776–2791.

[40] C. Cremers, The Scyther Tool, University of Oxford, Department of Computer Science, 2024, http://www.cs.ox.ac.uk/people/cas.cremers/scyther. (Accessed 08 Sep 2024).

[41] J. Cao, M. Ma, Y. Fu, H. Li, Y. Zhang, CPPHA: Capability-based privacy-protection handover authentication mechanism for SDN-based 5G HetNets, IEEE Trans. Dependable Secur. Comput. 18 (3) (2019) 1182–1195.

[42] C. Lai, Y. Ma, R. Lu, Y. Zhang, D. Zheng, A novel authentication scheme supporting multiple user access for 5G and beyond, IEEE Trans. Dependable Secur. Comput. (2022).

[43] Miracl, MIRACL core, 2024, https://github.com/miracl/core. (Accessed: 08 Sep 2024).

Xiaohui Ding is currently working toward the Ph.D. degree at the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include applied cryptography, IIoT security, and authentication and key agreement protocols.

Jian Wang received his M.S. degree in engineering from Southeast University, Nanjing, China in 1992. and received the Ph.D. degree s in Nanjing University in 1998. He ever is a postdoc at Tokyo University from 2000 to 2002. He is currently a Professor at the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. His research interests include applied cryptography, cryptographic protocol and malicious tracking. He has published more than 60 papers in international journals and conferences.

Yongxuan Zhao received his Master's degree in Management from Beijing Institute of Technology, Beijing, China in 2013. He is currently a researcher and director of the Information Technology Research Center of China Academy of Aero-Engine Research. His research interests include information technology, industrial digital transformation and IIoT security.

Zhiqiang Zhang is currently working toward the Ph.D. degree at the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include public key cryptography and privacy-preserving protocols.