A Secure One-to-Many Authentication and Key Agreement Scheme for Industrial IoT

Yang Ming, *Member, IEEE*, Pengfei Yang, Hassan Mahdikhani, and Rongxing Lu, *Fellow, IEEE*

***Abstract*—Industrial Internet of Things (IIoT) is gradually changing the mode of production, in which users can directly access data from smart devices through the network instead of collecting data where the smart devices are deployed. However, the data from these smart devices is usually transmitted via an insecure channel, which raises several security concerns. To solve these security issues, many authentication and key agreement schemes have been proposed. Nevertheless, the majority of said schemes only achieve authentication between one user and one smart device. When a user wants to access multiple smart devices simultaneously, he has to initiate multiple requests of key agreement, which incurs computation and communication costs. In this article, a secure one-to-many authentication and key agreement scheme for IIoT is put forward. Specifically, three factors, namely, smart card, password, and biometrics, are used to authenticate the user in the proposed scheme. By utilizing elliptic curve cryptography and Chinese remainder theorem, the different session keys between one user and multiple smart devices are agreed upon once, meaning the user only needs to initiate one request for authentication with multiple smart devices. Furthermore, security proof and analysis demonstrate that the proposed scheme is secure under the real-or-random model and can withstand a wide range of common attacks. Lastly, performance evaluation reveals that the proposed scheme requires less computational and communication costs compared with the related schemes, which is essential for smart device networks, operating with limited resources.**

***Index Terms*—Industrial internet of things (IIoT), key agreement, session key, three factors authentication.**

I. Introduction

THE rapid development of computer technology for decades has made the Internet of Things (IoT) a major trend in today's technological development. As the basis of IoT, smart devices generate massive data which has high practical significance [1]. Therefore, IoT has become the third wave in the development of the global information industry, after computers and the internet. In addition, relevant surveys show that the global demand for IoT applications will continue to grow in the future [2]. It has been predicted by [3] that the number of smart devices connected to the network will increase to 75.44 billion in 2025.

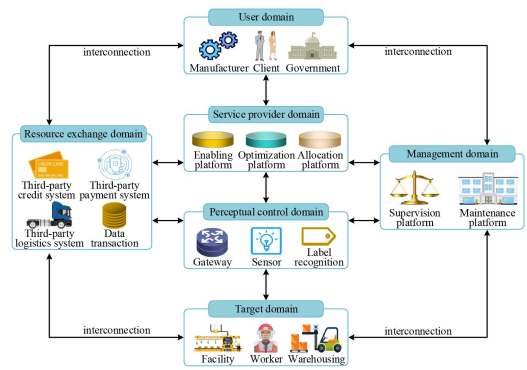

Fig. 1. Architecture of IIoT.

As a crucial component of IoT, Industrial Internet of Things (IIoT) has reached a period of rapid development. The architecture of IIoT is shown in Fig. 1, which includes user domain, service provider domain, perceptual control domain, target domain, resource exchange domain, and management domain. The user domain supports users to access IIoT, and applies to the service interface system. The service provider domain provides services, such as energy management and safety production. The data collected by the perceptual control domain are transmitted to the service provider domain through the gateway. The target domain provides information on all aspects of industrial production for the devices and tags in the perceptual control domain to perceive. The resource exchange domain realizes the exchange and the sharing of information and market resources. The management domain guarantees the stable and secure operation of other domains [4].

Currently, many countries have proposed various development strategies for IIoT [5], [6], [7]. For instance, the United States proposed the "National Strategic Plan for Advanced Manufacturing and Industrial Internet" [5], Germany put forward the "German Standardization Roadmap Industry 4.0" [6], and China has presented "Made in China 2025" [7]. The core of these developmental strategies is the deployment of smart devices into every aspect of industrial production, for the purpose of improved productivity and energy efficiency. Specifically, a large number of smart devices are deployed in industrial production. These smart devices are used for data collection, real-time monitoring, and more. Users can remotely obtain the

Manuscript received 13 January 2022; revised 3 July 2022 and 28 August 2022; accepted 22 September 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62072054, in part by the Key Research and Development Program of Shaanxi Province under Grant 2021GY-047, in part by the Fundamental Research Funds for the Central Universities, CHD under Grant 300102242201, and in part by the Project of Science and Technology of Xi'an City under Grant 2022JH-RGZN-0018. *(Corresponding Author: Yang Ming.)*

Yang Ming and Pengfei Yang are with the School of Information Engineering, Chang'an University, Xi'an 710064, China (e-mail: yangming@chd.edu.cn; 2019124036@chd.edu.cn).

Hassan Mahdikhani and Rongxing Lu are with the Canadian Institute for Cybersecurity, Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada (e-mail: hmahdikh@unb.ca; rlu1@unb.ca).

Digital Object Identifier 10.1109/JSYST.2022.3209868

data collected by smart devices. Users can utilize this data to judge the abnormality of industrial production, determining if there is loss in production, allowing them to prevent further losses. Additionally, deploying numerous smart devices can automatize production tasks, reducing personnel requirements and costs, while increasing efficiency. However, there are two basic requirements in the actual implementation process. First, the verification of data visitors is required to prevent unauthorized access. Second, because the data are transmitted on an open channel, it requires encryption to protect against malicious actors that can negatively impact production [8], [9]. At present, a large number of wireless technologies have been proposed [10], [11], [12], such as Bluetooth, WiFi, and Zigbee. However, these technologies only achieve key agreement or exchange process and lack the process of mutual authentication, which is not suitable for IIoT scenarios. In consequence, it is essential to devise a secure authentication and key agreement (AKA) scheme to ensure the legitimacy of data visitors and the security of the smart device data in IIoT.

Additionally, as the application of IIoT increases, there will be a causal increase in the number of smart devices. When users want to access data from multiple smart devices, they must negotiate a different session key with each smart device, in accordance with the existing schemes provided by [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. In conjunction with an increase in smart devices, there will be an increase in wasted communication resources and computational costs. Therefore, a scheme designed to achieve AKA between one user and multiple smart devices is desirable for efficient and scalable decision-making for IIoT.

To resolve the scalability and security issues for IIoT, we propose a secure one-to-many authentication and key agreement scheme. The main contributions of this article are listed as follows.

1) First, by applying elliptic curve cryptography (ECC) and Chinese remainder theorem (CRT), a secure one-to-many authentication and key agreement scheme is put forward. Smart cards, passwords, and biometrics are utilized to authenticate users. To accommodate smart devices with limited resources, lightweight procedures, such as symmetric cryptography, hash function, and exclusive OR (XOR) operation are implemented.
2) Second, the security of the proposed scheme is proven in the widely accepted real-or-random (ROR) model. In addition, the proposed scheme not only meets various functionality features, including user revocation and smart device join and leave but also withstands a variety of common attacks, especially to resist known session-specific temporary information attack.
3) Third, the computation and communication performance is evaluated by quantitative calculations. Compared with the related schemes, the computation and communication costs of the proposed scheme are lower when the user accesses multiple smart devices simultaneously.

The rest of this article is organized as follows. We review the related works in Section II and introduce some preliminaries in Section III. The system model, security model, and security requirements of the proposed scheme are given in Section IV. In Section V, the proposed scheme is introduced detailedly. The security analysis is shown in Section VI. Section VII offers performance evaluation. Finally, Section VIII concludes this article.

II. Related Works

As the core of IIoT technology, wireless sensor networks (WSNs) have provided great convenience to people's lives. AKA schemes proposed by [13], [14], [15], [16], [17], [18] attempt to resolve the existing security issues in WSNs. In 2009, Das et al. [13] proposed an authentication scheme for WSNs which achieves user authentication with smart card and password. Subsequently, He et al. [14] and Yeh et al. [15] found that Das et al.'s [13] proposed scheme cannot resist many attacks, such as sensor node compromised attack and denial of service attack. He et al. and Yeh et al. then proposed improved schemes. In 2017, Tai et al. [16] provided a lightweight AKA scheme in WSNs with XOR and hash function. However, in 2018, Shin et al. [17] reported that offline password guessing attack, as well as smart card stolen attack can be applied. Additionally, the anonymity of users and sensors is not considered in Tai et al.'s [16] scheme. Shin et al. then provided an improved scheme. In 2020, Zhang et al. [18] presented an authentication scheme that can resist known session-specific temporary information attack because the session key contains temporary secret values and long-term secret values.

As smart cards and passwords can be easily compromised, the two-factor schemes are limited in practical applications. Subsequently, biometric authentication was introduced into AKA schemes [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. Three-factor schemes were found to be significant improvement to two-factor schemes, due to the uniqueness of biometrics. In 2009, Fan et al. [19] put forward a truly three-factor authentication scheme with symmetric cryptography and public-key encryption for the first time. In 2016, Das et al. [20] designed an efficient three-factor authentication scheme because only lightweight operations, such as XOR and hash function are used. However, in 2018, Wu et al. [21] discovered that offline password guessing attack, as well as desynchronization attack were effective against Das et al.'s [20] scheme, and thus, devised an improved scheme. In the same year, Wazid et al. [22] supplied a three-factor AKA scheme which is very efficient, as the symmetric cryptography and hash function are used. Li et al. [23] published an authentication scheme with privacy perseveration. The scheme is secure against desynchronization attack as the user and gateway do not save the same secret value. Furthermore, the information does not need to be updated when an interaction is completed. In 2019, Jolfaei et al. [24] put forward a three-factor authentication scheme and claimed that their scheme can withstand multiple attacks. However, in 2020, Shin et al. [25] found Jolfaei et al.'s [24] scheme was vulnerable to user collusion attack, resulting in sensors being easily identified. Shin et al. then proposed an improved scheme, implementing ECC and hash function. Subsequently, Yang et al. [26] utilized XOR and hash function to achieve mutual authentication and key agreement, which achieves perfect forward

secrecy. Ali et al. [27] introduced a scheme using symmetric cryptography and hash function, which proves the security under Burrows–Abadi–Needham (BAN) logic. Additionally, they also gave the basic system models for WSNs. In 2021, Far et al. [28] created a lightweight authentication scheme suitable for IIoT, their scheme uses one-way hash chain technology to achieve less computation and communication costs and can support user revocation. Meshram et al. [29] provided an efficient, robust, and lightweight subtree-based three-factor authentication procedure, which can realize device joining. Li et al. [30] proposed a lightweight and secure authentication protocol with adaptive privacy-preserving property using XOR and hash function.

The above schemes have a good performance in single-smart device scenario, however, as the number of smart devices increases, huge overhead is required because the user's identity needs to be repeatedly authenticated. Therefore, one-to-many AKA schemes were put forward [31], [32]. In 2020, Cui et al. [31] presented an extensible authentication scheme with ECC and hash function. However, their scheme is insecure against known session-specific temporary information attack because the session key consists of identities of the participant and temporary information, where temporary information is secret and the identities are public for registered users. In 2021, Vinoth et al. [32] achieved one-to-many authentication with CRT and symmetric cryptography and negotiated the same session key by utilizing access control and secret sharing between the user and multiple devices. However, their scheme is vulnerable to the mutual imitation of registered devices, as the same key is computed by all registered devices. In summary, most of the existing systems either meet the security requirements for IIoT, or support one-to-many authentication.

III. Preliminaries

A. Elliptic Curve Cryptography

The concept of ECC was provided by Miller [33] and Koblite [34] for the first time. Given a large prime p, F_p is a prime finite field. The elliptic curve E on F_p is defined as the set of points satisfying $y^2 = x^3 + ax + b \bmod p$, where $a, b \in F_p$ and $4a^3 + 27b^3 \neq 0$. The infinite point O and all points on E form a cyclic additive group $\mathbb{G}$ with prime order q and generator P.

1) *Elliptic Curve Computational Diffie–Hellman (ECCDH) Problem [35]*: Given $P, aP, bP \in \mathbb{G}$, where $a, b \in \mathbb{Z}_q^*$, the ECCDH problem is to calculate $abP \in \mathbb{G}$.
2) *Elliptic Curve Computational Diffie–Hellman (ECCDH) Assumption [35]*: It is difficult for probabilistic polynomial time algorithms to solve the ECCDH problem with nonnegligible probability.

B. Fuzzy Extractor

The role of the fuzzy extractor [36] is to generate and reconstruct the biometric key, which includes two algorithms as follows.

1) $Gen(BIO_i) \rightarrow (BK_i, BP_i)$*:* Given a biometrics BIO_i as input, the probabilistic algorithm outputs a biometric key BK_i and reconstruction parameter BP_i.

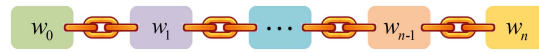

Fig. 2. Hash-chain structure.

2) $Rep(BIO_i', BP_i) \rightarrow BK_i$*:* Given a biometrics BIO_i' that is similar to BIO_i and reconstruction parameter BP_i as input, the deterministic algorithm outputs the biometric key BK_i.

C. Chinese Remainder Theorem

CRT [37] is described as follows. Let $m_1, m_2, \ldots, m_n$ be integers, which are pairwise relatively prime. Then for any integers $a_1, a_2, \cdots, a_n \in \mathbb{Z}$, the general solution of the following equation is $x = a_1t_1M_1 + a_2t_2M_2 + \cdots + a_nt_nM_n + kM = \sum_{i=1}^{n} a_it_iM_i + kM, k \in \mathbb{Z}$:

$$\begin{cases} x \equiv a_1 (\bmod\ m_1) \\ x \equiv a_2 (\bmod\ m_2) \\ \qquad \vdots \\ x \equiv a_n (\bmod\ m_n) \end{cases} \tag{1}$$

where $M = m_1 \times m_2 \times \cdots \times m_n$, $M_i = M/m_i$, $M_it_i \equiv 1 (\bmod\ m_i), i \in [1, n]$.

In the case of modulo M, the above equation has only a unique solution $x = (\sum_{i=1}^{n} a_it_iM_i) \bmod\ M$.

D. Hash-Chain

The concept of hash-chain was first proposed by Lamport [38]. Given a one way hash function $h : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$, a hash-chain is defined as a sequence $\{w_0, w_1, \ldots, w_n\}$, where w_0 is a random number and $w_i = h(w_{i-1})$, $0 < i \leq n$. Hash-chain structure is shown in Fig. 2.

IV. System Model, Threat Model, and Security Requirements

A. System Model

IIoT offers new ways of smart production such that precise control of the industry is achieved through information exchange between users and smart devices. But before information exchange, the AKA schemes needs to be used to achieve mutual authentication and agree session keys for encrypting data. Our system model of one-to-many AKA scheme for IIoT is shown in Fig. 3, which consists of four entities: 1) a key management center (KMC), 2) a gateway (GW), 3) a user (U_i), and 4) n smart devices $(SD_1, SD_2, \ldots, SD_n)$, $n < N$, where N is the maximum number of smart devices, $N \leq 20$. KMC deploys $(SD_1, SD_2, \ldots, SD_n)$, registers U_i, and authorizes GW. Then, U_i and $(SD_1, SD_2, \ldots, SD_n)$ agree the session key with the help of GW. Finally, U_i and $(SD_1, SD_2, \ldots, SD_n)$ use agreed session key for secure communication. Specifically, $(SD_1, SD_2, \ldots, SD_n)$ generate n different temporary information $(t_1, t_2, \ldots, t_n)$, which are integrated into SR by GW through CRT. Then, GW securely sends SR to U_i for generating different session key by computing $t_j = SR \bmod \text{SID}_j$, where SID_j is the identity of SD_j and $j \in \{1, 2, \ldots, n\}$. In this

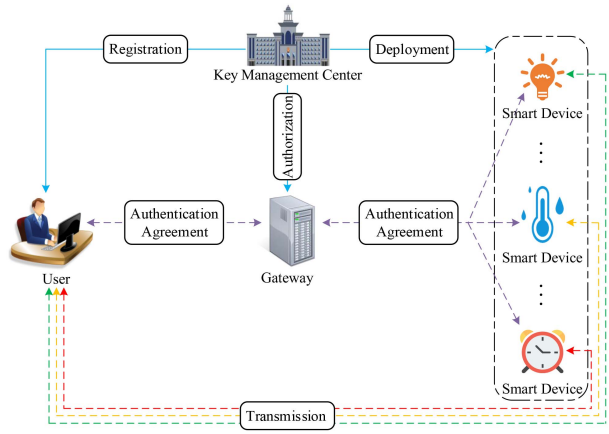

Fig. 3. System model.

way, U_i can agree n session keys with $(SD_1, SD_2, \ldots, SD_n)$ by initiating one agreement request. Comparing to the related AKA schemes [18], [23], [25], [28], in which U_i is required to initiate n agreement requests and authenticated by GW n times for agreeing session keys with $(SD_1, SD_2, \ldots, SD_n)$, the proposed scheme saves a lot of computation and communication overhead.

1) KMC: KMC is a trusted entity, whose main responsibility is to generate the system parameters, deploy $(SD_1, SD_2, \ldots, SD_n)$, register U_i, and authorize GW.
2) GW: GW is a trusted entity, who assists U_i and $(SD_1, SD_2, \ldots, SD_n)$ to run mutual authentication in key agreement phase.
3) U_i: U_i uses own identity to apply for a smart card from KMC and initiates the session key agreement request through GW using smart card.
4) SD_j: SD_j represents the jth smart device who collects data in IIoT and sends data to U_i.

B. Threat Model

Dolev–Yao (DY) threat model [39] and CK threat model [40] are used in the proposed scheme, according to the models, the adversary $\mathcal{A}$ is able to read, modify, delete, forge, and replay the information. U_i and $(SD_1, SD_2, \ldots, SD_n)$ are not regarded as trusted entities because they are easily stolen, while KMC and GW are regarded as trusted entities and cannot be compromised. In addition, $\mathcal{A}$ can obtain the ephemeral information, i.e., session state, session keys, and part of the user's secret information.

C. Security Requirements

This subsection lists the security requirements that should be achieved.

1) *Mutual Authentication:* To ensure only an authorized user can access smart device data, it is necessary that AKA schemes achieve mutual authentication between the user and smart devices.

2) *Anonymity:* To protect the identities of the user and the smart device from being leaked, the AKA schemes should provide anonymity. Even if an adversary intercepts the messages transmitted on the public channel, it should be impossible to determine which user or smart device sent it.

3) *Untraceability:* To enhance the privacy protection of the user and smart device, the AKA schemes should achieve untraceability, namely, different messages sent by the same user or smart device cannot be linked together.

4) *Perfect Forward Secrecy:* To protect the security of previously transmitted information, the AKA schemes should achieve perfect forward secrecy. Even though the adversary obtains the private key of the smart device and the latest session key, it is difficult to recover the previous session keys.

5) *Resistance to Multiple Attacks:* The AKA schemes should be able to withstand a large number of common attacks, such as smart card stolen attack, impersonation attack, replay attack, man-in-the-middle attack, denial of service attack, and known session-specific temporary information attack.

TABLE I
NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
KMC	Key management center.
GW	Gateway.
U_i	ith user.
SD_j	jth smart device.
GID	Identity of GW.
$\mathrm{UID}_i, PW_i, \mathrm{BIO}_i$	Identity, password, and biometrics of U_i.
SID_j	Identity of SD_j.
t_j	Temporary information of SD_j.
msk	Master secret key of KMC.
(s, S)	Private key and public key of GW.
$\mathrm{Gen}(\cdot), \mathrm{Rep}(\cdot)$	Generation and reproduction algorithm of fuzzy extractor.
BK_i, RP_i	Biometric key and reconstruction parameter of U_i.
TSK	Common temporary key between GW and all smart devices.
sk_j	Private key of SD_j.
$\mathrm{Enc}_K(\cdot), \mathrm{Dec}_K(\cdot)$	Symmetric encryption and decryption algorithm using key K.
TS_i	Current timestamp.
ΔTS	Maximum transmission delay.

V. Proposed Scheme

In this section, a secure one-to-many authentication and key agreement scheme is provided in detail, which includes initialization, smart device deployment, user registration, gateway authorization, authentication and key agreement, user revocation, password and biometrics update, common temporary key update, smart device join and leave phase. The specific description is as follows. Table I provides the notation utilized in this article.

A. Initialization Phase

KMC performs the system initialization by running the following steps.

1) KMC produces a group $\mathbb{G}$ of prime order q based on a nonsingular elliptic curve E defined over a finite field, where P is the generator of $\mathbb{G}$. It selects msk $\in \mathbb{Z}_q^*$ as its master secret key.
2) KMC chooses one cryptographic hash function $h : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$.
3) KMC selects the identity GID for the gateway GW, and calculates $s = h(\mathrm{GID}||\mathrm{msk})$ as the private key of GW and the corresponding public key $S = s \cdot P$.

4) KMC publishes the system parameters $\text{params} = \{P, \mathbb{G}, q, \text{GID}, S, h\}$.

B. Smart Device Deployment Phase

KMC generates the identity and keys for $SD_j (j \in [1, n])$ before deployment [41], [42]. The specific process is mathematically depicted as follows.

1) KMC chooses a large prime number p satisfying $p > q$, where p is used for defining a multiplicative group $\mathbb{Z}_p^*$. KMC selects $\text{SID}_j \in \mathbb{Z}_p^*$, $j \in [1, N]$ in advance as the identities of the smart devices and computes $\partial = \text{SID}_1 \times \text{SID}_2 \times \cdots \times \text{SID}_N$, $x_j = \partial / \text{SID}_j$, $y_j \equiv 1/x_j \bmod \text{SID}_j$ and $x_j y_j$. KMC saves SID_j and $x_j y_j$ in the local database.
2) KMC chooses SID_j from $(\text{SID}_1, \text{SID}_2, \ldots, \text{SID}_N)$ as the identity of SD_j and computes $sk_j = h(\text{SID}_j || s)$ as the private key of SD_j.
3) KMC chooses a common temporary key $\text{TSK} \in \mathbb{Z}_q^*$ for all smart devices and stores $\{\text{SID}_j, sk_j, \text{TSK}\}$ in the memory of SD_j.

Note that KMC prepares N tuples $\{\text{SID}_j, x_j, y_j, x_j y_j\}$ for n smart devices in this phase, which means that $N - n$ tuples are free, the purpose is to allow new smart devices to join after the system is deployed. In addition, when n reaches to N, KMC reperforms the smart device deployment phase to achieve system scalability.

C. User Registration Phase

U_i registers with KMC via a reliable channel. The specific process is mathematically depicted as follows.

1) U_i randomly chooses a unique identity UID_i and submits it to KMC via a secure channel.
2) Upon receiving UID_i, KMC randomly chooses $r_i \in \mathbb{Z}_q^*$ and computes $a = h(\text{UID}_i || s || r_i)$. Then KMC stores $\{\text{UID}_i, r_i\}$ in its database and sends a smart card contained $\{a, P, \text{Gen}(\cdot), \text{Rep}(\cdot), h(\cdot)\}$ to U_i securely (such as hand delivery, registered post).
3) Given the smart card, U_i enters his identity UID_i, password PW_i, and biometrics BIO_i, then, the smart card SC_i of U_i extracts biometrics key BK_i and reconstruction parameter RP_i with fuzzy extractor $\text{Gen}(\text{BIO}_i) \rightarrow (BK_i, RP_i)$. Next, SC_i computes $A = a \oplus h(\text{UID}_i \oplus PW_i \oplus BK_i)$, $B = h(\text{UID}_i || PW_i || BK_i) \bmod l_0$, where l_0 is the integer chosen by the smart card and decides the capability of obstructing online guessing attack using fuzzy verifier [43].
4) U_i replaces a with A and stores $\{B, RP_i, l_0\}$ in the smart card.

D. Gateway Authorization Phase

KMC authorizes the gateway GW to assist the user and all smart devices in making mutual authentication and key agreement. The specific process is mathematically depicted as follows.

1) KMC sends s to GW securely.
2) KMC transmits $\{\text{SID}_j, x_j y_j, \text{TSK}\}$, $j \in [1, N]$ and $\{\text{UID}_i, r_i\}$ to GW via a secure channel.

E. Authentication and Key Agreement Phase

U_i and all smart devices $\{SD_1, SD_2, \ldots, SD_n\}$ make mutual authentication and agree to different session keys with the help of GW, where U_i only needs to initiate a request of key agreement. The specific process is mathematically depicted as follows.

1) U_i first inserts his smart card into the card reader and inputs his identity UID_i, password PW_i, and biometrics BIO_i. Then, the smart card SC_i of U_i reconstructs biometric key BK_i with fuzzy extractor $\text{Rep}(\text{BIO}_i, RP_i) \rightarrow BK_i$, computes $B' = h(\text{UID}_i || PW_i || BK_i) \bmod l_0$ and checks whether $B' = B$. If it does not hold, the login request is terminated. Otherwise, the authentication for U_i is successful. Next, U_i randomly chooses $d \in \mathbb{Z}_q^*$ and the current timestamp TS_1, computes $a = A \oplus h(\text{UID}_i \oplus PW_i \oplus BK_i)$, $M_1 = (a + d) \cdot P$, $K = h((a + d) \cdot S)$, $e = h(\text{UID}_i || \text{GID} || a || TS_1)$, $M_2 = \text{Enc}_K(\text{UID}_i, e)$ and $M_3 = h(M_1 || M_2 || \text{UID}_i || e || TS_1)$. Finally, the message $\text{msg}_1 = \{M_1, M_2, M_3, TS_1\}$ is sent to GW by U_i via an open channel.
2) Upon receiving msg_1 from U_i, GW checks whether $|TS_1 - TS_1'| \leq \Delta TS$, where TS_1' is the time that GW received msg_1. Then, GW computes $K = h(sM_1)$, $(\text{UID}_i, e) = Dec_K(M_2)$, and finds r_i in the local database through UID_i, computes $a' = h(\text{UID}_i || s || r_i)$ and $e' = h(\text{UID}_i || \text{GID} || a' || TS_1)$. GW checks whether $e' = e$. If it holds, GW computes $M_3' = h(M_1 || M_2 || UID_i || e' || TS_1)$ and checks whether $M_3' = M_3$. If it holds, GW computes $\text{TSK}' = h(\text{TSK})$ and updates TSK with the new value TSK'. Next, GW chooses the current timestamp TS_2 and computes $M_4 = \text{Enc}_{\text{TSK}}(\text{UID}_i, \text{GID}, e)$, $M_5 = h(M_4 || \text{UID}_i || \text{GID} || e || TS_2)$. Finally, GW broadcasts the message $\text{msg}_2 = \{M_4, M_5, TS_2\}$ to all smart devices $\{SD_1, SD_2, \ldots, SD_n\}$.
3) Upon receiving msg_2 from GW, each smart device $SD_j (j \in [1, n])$ checks whether $|TS_2 - TS_2^{j'}| \leq \Delta TS$, where $TS_2^{j'}$ is the time that SD_j received msg_2. Then, SD_j computes $\text{TSK}' = h(\text{TSK})$, $(\text{UID}_i, \text{GID}, e) = \text{Dec}_{\text{TSK}'}(M_4)$, $M_5' = h(M_4 || \text{UID}_i || \text{GID} || e || TS_2)$ and checks whether $M_5' = M_5$. If it does not hold, SD_j discards the message. Otherwise, SD_j updates TSK with TSK' and chooses the current timestamp TS_3^j. Next, SD_j randomly chooses $t_j \in \mathbb{Z}_q^*$ and computes $c_j = sk_j \oplus t_j$, $M_6^j = \text{Enc}_{\text{TSK}}(\text{SID}_j, c_j)$ and $M_7^j = h(M_6^j || \text{SID}_j || t_j || TS_3^j)$. Finally, each smart device $SD_j (j \in [1, n])$ returns the message $msg_3^j = \{M_6^j, M_7^j, TS_3^j\}$ to GW publicly.
4) Upon receiving msg_3^j from all smart devices $\{SD_1, SD_2, \ldots, SD_n\}$, for every message msg_3^j $(j \in [1, n])$, GW checks whether $|TS_3^j - TS_3^{j'}| \leq \Delta TS$, where $TS_3^{j'}$ is the time that GW received msg_3^j. Then, GW computes $(\text{SID}_j, c_j) = \text{Dec}_{\text{TSK}}(M_6^j)$, $sk_j = h(\text{SID}_j || s)$, $t_j = c_j \oplus sk_j$, $M_7^{j'} = h(M_6^j || \text{SID}_j || t_j || TS_3^j)$ and

checks whether $M_7^{j'} = M_7^j$. If it holds, GW chooses the current timestamps TS_4, TS_5 and computes $SR = t_1x_1y_1 + \cdots + t_nx_ny_n \bmod \partial$, $U = \sum_{j=1}^n sk_j$, $M_8 = h(U||TSK||TS_4)$, $M_9 = M_8 \oplus \text{TSK}$, $M_{10} = h(M_9||M_8||TS_4)$, $M_{11} = \text{Enc}_K(M_8, SR)$ and $M_{12} = h(M_{11}||M_8||SR||TS_5)$. Finally, GW broadcasts the message $\text{msg}_4 = \{M_9, M_{10}, TS_4\}$ to all smart devices $\{SD_1, SD_2, \ldots, SD_n\}$ and sends the message $\text{msg}_5 = \{M_{11}, M_{12}, TS_5\}$ to U_i.

5) Upon receiving msg_4 from GW, each smart device $SD_j (j \in [1, n])$ checks whether $|TS_4 - TS_4^{j'}| \leq \Delta TS$, where $TS_4^{j'}$ is the time that SD_j received msg_4. Then, SD_j computes $M_8 = M_9 \oplus \text{TSK}$, $M'_{10} = h(M_9||M_8||TS_4)$ and checks whether $M'_{10} = M_{10}$. Finally, SD_j computes the session key $SK_j = h(h(\text{UID}_i||\text{GID}||e||M_8) \cdot t_j)$.
6) Upon receiving msg_5 from GW, U_i checks whether $|TS_5 - TS'_5| \leq \Delta TS$, where TS'_5 is the time that U_i received msg_5. Then, U_i computes $(M_8, SR) = \text{Dec}_K(M_{11})$, $M'_{12} = h(M_{11}||M_8||SR||TS_5)$ and checks whether $M'_{12} = M_{12}$. Finally, U_i computes the master session key $SK = h(\text{UID}_i||\text{GID}||e||M_8) \cdot SR$.

If U_i wishes to communicate with SD_j, the session key SK_j would be computed from the master session key SK by the following way:

$$\begin{aligned} SK_j &= h(SK \bmod \text{SID}_j) \\ &= h((h(\text{UID}_i||\text{GID}||e||M_8) \cdot SR) \bmod \text{SID}_j) \\ &= h(h(\text{UID}_i||\text{GID}||e||M_8) \cdot t_j). \end{aligned} \tag{2}$$

F. User Revocation Phase

If U_i loses the smart card or wishes to replace it with a new smart card, the old smart card's authentication should be revoked. The specific process is mathematically depicted as follows.

1) U_i sends his identity UID_i to KMC securely.
2) KMC checks UID_i, if it exists in the local database, KMC randomly chooses $r_i^{\text{new}} \in \mathbb{Z}_q^*$ and replaces r_i in the local database with r_i^{new}.
3) KMC computes $a^{\text{new}} = h(\text{UID}_i||s||r_i^{\text{new}})$ and stores $\{a^{\text{new}}, P, \text{Gen}(\cdot), \text{Rep}(\cdot), h(\cdot)\}$ in a new smart card. Finally, the new smart card is sent to U_i by KMC via a reliable way (such as hand delivery, registered post).
4) KMC securely sends $\{\text{UID}_i, r_i^{\text{new}}\}$ to GW for updating the database of GW.

G. Password and Biometrics Update Phase

The password and biometrics of legal users can be updated locally without the help of KMC. The specific process is mathematically depicted as follows.

1) U_i enters UID_i, PW_i, and BIO_i, the smart card SC_i of U_i computes $\text{Rep}(\text{BIO}_i, RP_i) \rightarrow BK_i$, $B' = h(\text{UID}_i||PW_i||BK_i) \bmod l_0$ and checks whether $B' = B$. If it does not hold, SC_i rejects update request. Otherwise, SC_i computes $a = A \oplus h(\text{UID}_i \oplus PW_i \oplus BK_i)$.
2) U_i inputs new password PW_i^{new} and new biometrics $\text{BIO}_i^{\text{new}}$, SC_i computes $\text{Gen}(\text{BIO}_i^{\text{new}}) \rightarrow (BK_i^{\text{new}}, RP_i^{\text{new}})$, $A^{\text{new}} = a \oplus h(\text{UID}_i \oplus PW_i^{\text{new}} \oplus BK_i^{\text{new}})$ and $B^{\text{new}} = h(\text{UID}_i||PW_i^{\text{new}}||BK_i^{new}) \bmod l_0$. At last, SC_i replaces $\{A, B, RP_i\}$ with $\{A^{\text{new}}, B^{\text{new}}, RP_i^{\text{new}}\}$.

H. Common Temporary Key Update Phase

When the common temporary key TSK is leaked, KMC can update TSK to ensure the security of the system. The specific process is mathematically depicted as follows.

1) KMC randomly chooses a new common temporary key $\text{TSK}^{\text{new}} \in \mathbb{Z}_q^*$, the current timestamp TS^{new}, and computes $\text{TKG}_j^{\text{new}} = sk_j \oplus h(TS^{\text{new}})$ $(j \in [1, n])$, $f(x) = \prod_{j=1}^n (x - \text{TKG}_j^{\text{new}}) + \text{TSK}^{\text{new}} = x^n + g_{n-1}^{\text{new}} x^{n-1} + g_{n-2}^{\text{new}} x^{n-2} + \cdots + g_0^{\text{new}}$ and $Z = h(\text{TSK}^{\text{new}}||TS^{\text{new}})$. Finally, KMC broadcasts $\{g_{n-1}^{\text{new}}, g_{n-2}^{\text{new}}, \ldots, g_0^{\text{new}}, Z, TS^{\text{new}}\}$ to all smart devices $\{SD_1, SD_2, \ldots, SD_n\}$ and sends TSK^{new} to GW securely.
2) Each smart device $SD_j (j \in [1, n])$ checks the freshness of TS^{new}. If it holds, SD_j computes $\text{TKG}_j^{\text{new}} = sk_j \oplus h(TS^{\text{new}})$, $f(\text{TKG}_j^{\text{new}}) = (\text{TKG}_j^{\text{new}})^n + g_{n-1}^{\text{new}} (\text{TKG}_j^{\text{new}})^{n-1} + g_{n-2}^{\text{new}} (\text{TKG}_j^{\text{new}})^{n-2} + \cdots + g_0^{\text{new}} = \text{TSK}^{\text{new}}$, $Z' = h(\text{TSK}^{\text{new}}||TS^{\text{new}})$ and checks whether $Z' = Z$. If it does not hold, SD_j discards the message. Otherwise SD_j replaces TSK in its memory with TSK^{new}.
3) GW replaces TSK in its database with TSK^{new}.

I. Smart Device Join Phase

As a business expands its operations, there will be a need to deploy additional smart devices in their ecosystem. The new smart device SD_{n+1} needs to register with KMC. KMC will then select a new common temporary key to ensure forward security. The specific process is mathematically depicted as follows.

1) KMC chooses the identity SID_{n+1} and computes the private key sk_{n+1} in the same way as smart device deployment phase.
2) KMC randomly chooses a new common temporary key $\text{TSK}^{\text{new}} \in \mathbb{Z}_q^*$ and distributes it to other smart devices by common temporary key update phase.

J. Smart Device Leave Phase

Some smart devices may leave due to malfunction after the system is deployed. Suppose that the smart device SD_i leaves the system, KMC needs to select a new common temporary key for other smart devices to ensure backward security. The specific process is mathematically depicted as follows.

1) KMC randomly chooses a new common temporary key $\text{TSK}^{\text{new}} \in \mathbb{Z}_q^*$, the current timestamp TS^{new}, and computes $\text{TKG}_j^{\text{new}} = sk_j \oplus h(TS^{\text{new}})$ $(j \in [1, n], j \neq i)$, $f(x) = \prod_{j=1, j \neq i}^n (x - \text{TKG}_j^{\text{new}}) + \text{TSK}^{\text{new}} = x^{n-1} + g_{n-2}^{\text{new}} x^{n-2} + g_{n-3}^{\text{new}} x^{n-3} + \cdots + g_0^{\text{new}}$, $Z = h(\text{TSK}^{\text{new}}||TS^{\text{new}})$. Finally, KMC broadcasts $\{g_{n-2}^{\text{new}}, g_{n-3}^{\text{new}}, \ldots, g_0^{\text{new}}, Z, TS^{\text{new}}\}$ to all smart devices

$\{SD_1, \ldots, SD_{j\neq i}, \ldots, SD_n\}$ and sends TSK^{new} to GW securely.

2) $SD_j (j \in [1, n], j \neq i)$ checks the freshness of TS^{new}. If it holds, SD_j computes $\text{TKG}_j^{\text{new}} = sk_j \oplus h(TS^{\text{new}})$, $f(\text{TKG}_j^{\text{new}}) = (\text{TKG}_j^{\text{new}})^{n-1} + g_{n-2}^{\text{new}}(\text{TKG}_j^{\text{new}})^{n-2} + g_{n-3}^{\text{new}}(\text{TKG}_j^{\text{new}})^{n-3} + \cdots + g_0^{\text{new}} = \text{TSK}^{\text{new}}$, $Z' = h(\text{TSK}^{\text{new}}||TS^{\text{new}})$ and checks whether $Z' = Z$. If it does not hold, SD_j discards the message. Otherwise SD_j replaces TSK in its memory with TSK^{new}.
3) GW replaces TSK in its database with TSK^{new}.

VI. Security Analysis

In this section, the security of the proposed scheme is evaluated by security proof and analysis.

A. Security Model

The widespread ROR model [44] is adopted in the security model. The primitives of the ROR model are introduced as follows.

1) *Participants:* There are multiple parties in the proposed scheme, user U_i, gateway GW, and smart device $SD_j (j \in [1, n])$. Denote the instances α, β, and γ of U_i, GW, and SD_j by $\prod_{U_i}^{\alpha}$, $\prod_{GW}^{\beta}$ and $\prod_{SD_j}^{\gamma}$. These instances are simulated as oracles.
2) *Partnership:* If $\prod_{U_i}^{\alpha}$ and $\prod_{SD_j}^{\gamma}$ can exchange information directly, share the same session key, and do not form a session key with other instances, they are regarded as partners.
3) *Freshness:* If a session key SK has been established between $\prod_{U_i}^{\alpha}$ and $\prod_{SD_j}^{\gamma}$, and has not revealed to the adversary, $\prod_{U_i}^{\alpha}$ and $\prod_{SD_j}^{\gamma}$ are regarded as fresh.

The adversary $\mathcal{A}$ can obtain system parameters and intercept messages transmitted on the public channel. Moreover, $\mathcal{A}$ can modify the messages or forge new messages to deceive other instances. $\mathcal{A}$ can make the following queries.

1) $Hash(\cdot)$*:* Upon receiving $\mathcal{A}$'s query, a random value is returned.
2) $Execute(\prod_{U_i}^{\alpha}, \prod_{GW}^{\beta}, \prod_{SD_j}^{\gamma})$*:* This query is simulated as an eavesdropping attack. Upon receiving $\mathcal{A}$'s query, the messages transmitted among $\prod_{U_i}^{\alpha}$, $\prod_{GW}^{\beta}$, and $\prod_{SD_j}^{\gamma}$ on the public channel are returned.
3) $Send(\prod_{U_i}^{\alpha} / \prod_{GW}^{\beta} / \prod_{SD_j}^{\gamma}, m)$*:* This query is simulated as an active attack. Upon receiving $\mathcal{A}$'s query on the message m, a response message is returned.
4) $Reveal(\prod_{U_i}^{\alpha}, \prod_{SD_j}^{\gamma})$*:* Upon receiving $\mathcal{A}$'s query, if the session key SK has been built between instances $\prod_{U_i}^{\alpha}$ and $\prod_{SD_j}^{\gamma}$, SK is returned.
5) $Corrupt(\prod_{U_i}^{\alpha}, v)$*:* This query simulates the security of three-factor information. Upon receiving $\mathcal{A}$'s query, the related information is returned.

a) $v = 0$*:* The password PW_i is returned to $\mathcal{A}$.
b) $v = 1$*:* The data in the smart card is returned to $\mathcal{A}$.
c) $v = 2$*:* The biometrics BIO_i is returned to $\mathcal{A}$.

6) $Test(\prod_{U_i}^{\alpha}, \prod_{SD_j}^{\gamma})$*:* This query is simulated as the semantic security of the session key SK^* between U_i and SD_j. Upon receiving $\mathcal{A}$'s query, a bit b is randomly selected. If $b = 1$, the session key SK^* is returned, if $b = 0$, a string of the same length as SK^* is selected and returned.

Semantic security of session key: In the ROR model, the goal of $\mathcal{A}$ is to distinguish the real session key of the instance from a random string by the way of games. $\mathcal{A}$ can make a number of $Execute$, $Send$, $Reveal$, $Corrupt$, and $Test$ queries to $\prod_{U_i}$ or its partner. As soon as the game is over, $\mathcal{A}$ guesses a bit b' and wins the game if $b = b'$.

The advantage of $\mathcal{A}$ in breaking the semantic security of proposed scheme Σ is defined as $Adv_{\mathcal{A}}^{\Sigma}(t) = |2\Pr[b' = b] - 1|$.

If $Adv_{\mathcal{A}}^{\Sigma}(t)$ is negligible for any probabilistic polynomial time adversary $\mathcal{A}$, the proposed scheme Σ is secure.

B. Formal Security Proof

Theorem 1: Assuming that $\mathcal{A}$ is the adversary running in polynomial time t against the proposed scheme Σ. Denote D and N be uniformly distributed dictionaries of password and biometrics. The advantage of $\mathcal{A}$ in breaking session key security of the proposed scheme is

$$\text{Adv}_{\mathcal{A}}^{\Sigma}(t) \leq \frac{q_h^2}{|\text{Hash}|} + 2 \cdot \text{Adv}_{\mathcal{A}}^{\text{ECCDH}}(t') + 2 \cdot \max\left(\frac{q_s}{|N|}, \frac{q_s}{|D|}, \delta q_s\right)$$

where $|D|$, $|N|$, $|Hash|$, $Adv_{\mathcal{A}}^{ECCDH}(t')$, δ, q_e, q_s, and q_h represent the size of D and N, the scope space of hash function $h(\cdot)$, the advantage of $\mathcal{A}$ in breaking the ECCDH problem in polynomial time t', the probability for the case "false positive," the number of $Execute$ query, $Send$ query, and $Hash$ query, respectively.

Proof: Five games are set up to prove the security of the proposed scheme, which is denoted as G_k, $k \in [0, 4]$. Suc_k and $\Pr[\text{Suc}_k]$ represent the event and the probability that $\mathcal{A}$ successfully guesses the correct bit b in the game G_k, respectively. ■

Game G_0: The simulation of G_0 is identical to the real attack in the ROR model without any queries. According to the definition of semantic security, it follows that

$$\text{Adv}_{\mathcal{A}}^{\Sigma}(t) = |2\Pr[\text{Suc}_0] - 1|. \tag{3}$$

Game G_1: This game models the eavesdropping attack launched by $\mathcal{A}$. During the authentication and key agreement phase, $\mathcal{A}$ can obtain the messages $\text{msg}_1 = \{M_1, M_2, M_3, TS_1\}$, $\text{msg}_2 = \{M_4, M_5, TS_2\}$, $\text{msg}_3^j = \{M_6^j, M_7^j, TS_3^j\}$, $\text{msg}_4 = \{M_9, M_{10}, TS_4\}$, and $\text{msg}_5 = \{M_{11}, M_{12}, TS_5\}$ transmitted on the open channel by making $Execute(\prod_{U_i}^{\alpha}, \prod_{GW}^{\beta}, \prod_{SD_j}^{\gamma})$ query. After that, $\mathcal{A}$ makes $Test(\prod_{U_i}^{\alpha}, \prod_{SD_j}^{\gamma})$ query and determines whether the output of $Test(\prod_{U_i}^{\alpha}, \prod_{SD_j}^{\gamma})$ query is the real session key SK_j^* or a random string. However, according to the proposed scheme, the session key $SK_j^* = h(h(UID_i||GID||e||M_8) \cdot t_j)$ contains the secret information UID_i, e, M_8 and t_j which cannot be obtained by $\mathcal{A}$ from the eavesdropping messages msg_1, msg_2, msg_3^j, msg_4, and msg_5. By $Execute(\prod_{U_i}^{\alpha}, \prod_{GW}^{\beta}, \prod_{SD_j}^{\gamma})$ query, the probability that $\mathcal{A}$ correctly guesses the bit b does not increase. Therefore, it follows

that

$$\Pr[\text{Suc}_1] = \Pr[\text{Suc}_0]. \tag{4}$$

Game G_2: The game models an active attack by adding $Send$ query and $Hash$ query compared to G_1. $\mathcal{A}$ fools the participant $\prod_{U_i}^{\alpha} / \prod_{GW}^{\beta} / \prod_{SD_j}^{\gamma}$ to convince the forged message in the game G_2. $\mathcal{A}$ can make many $Hash$ queries to discover the collisions of the secret key. After that, $\mathcal{A}$ makes $Test(\prod_{U_i}^{\alpha}, \prod_{SD_j}^{\gamma})$ query and determines whether the output of $Test(\prod_{U_i}^{\alpha}, \prod_{SD_j}^{\gamma})$ query is the real session key SK_j^* or a random string. However, it can be seen from the proposed scheme, all messages contain the timestamps, the random numbers and the common temporary key to ensure randomness. Thus, when $\mathcal{A}$ makes $Send(\prod_{U_i}^{\alpha} / \prod_{GW}^{\beta} / \prod_{SD_j}^{\gamma}, m)$ query, the probability of message collision is negligible. Therefore, according to the birthday paradox, it follows that

$$|\Pr[Suc_1] - \Pr[Suc_2]| \leq \frac{q_h^2}{2|Hash|}. \tag{5}$$

Game G_3: In this game, $\mathcal{A}$ can try to compute the session key SK_j^* through the attained sensitive information. Specifically, $\mathcal{A}$ can first obtain $M_1 = (a+d) \cdot P$, $M_2 = \text{Enc}_K(\text{UID}_i, e)$, $M_{11} = \text{Enc}_K(M_8, SR), \text{GID}$ and S by eavesdropping. Then, $\mathcal{A}$ computes $K = h((a+d) \cdot S) = h(sM_1)$ and obtains the secret value UID_i, e, M_8 and SR by decrypting the messages M_2 and M_{11} using K. Finally, $\mathcal{A}$ computes the session key $SK_j^* = h((h(\text{UID}_i||\text{GID}||e||M_8) \cdot SR) \bmod \text{SID}_j)$. After that, $\mathcal{A}$ makes $Test(\prod_{U_i}^{\alpha}, \prod_{SD_j}^{\gamma})$ query and determines whether the output of $Test(\prod_{U_i}^{\alpha}, \prod_{SD_j}^{\gamma})$ query is the real session key SK_j^* or a random string. However, $\mathcal{A}$ has to know the private key s of GW or $(a+d)$ to gain $K = h((a+d) \cdot S) = h(sM_1)$, which is equivalent to solving the ECCDH problem in polynomial time t'. Therefore, it follows that

$$|\Pr[\text{Suc}_2] - \Pr[\text{Suc}_3]| \leq \text{Adv}_{\mathcal{A}}^{\text{ECCDH}}(t') \tag{6}$$

Game G_4: Compared with G_3, $Corrupt(\prod_{U_i}^{\alpha}, v)$ query is added in G_4 to simulate the security of three-factor information. By this query, $\mathcal{A}$ can interact with GW by impersonating U_i to obtain the session key SK_j^*. In G_4, suppose that $\mathcal{A}$ can get two factors at most because the worst case is considered. Therefore, it can be divided into the following three cases.

Case 1: $\mathcal{A}$ gets PW_i and the data $\{A, B, P, RP_i, l_0, \text{Gen}(\cdot), \text{Rep}(\cdot), h(\cdot)\}$ in the smart card by making $Corrupt(\prod_{U_i}^{\alpha}, 0)$ and $Corrupt(\prod_{U_i}^{\alpha}, 1)$ queries. Next, the biometrics BIO_i needs to be guessed by $\mathcal{A}$. Since $\mathcal{A}$ can make q_s queries, so the probability which $\mathcal{A}$ imitates U_i successfully is $\frac{q_s}{|N|}$.

Case 2: $\mathcal{A}$ obtains PW_i and BIO_i by making $Corrupt(\prod_{U_i}^{\alpha}, 0)$ and $Corrupt(\prod_{U_i}^{\alpha}, 2)$ queries. Next, a needs to be computed by $\mathcal{A}$. Since $\mathcal{A}$ does not have the data in the smart card and the information about a, so the probability which $\mathcal{A}$ imitates U_i successfully is negligible.

Case 3: $\mathcal{A}$ acquires the data $\{A, B, P, RP_i, l_0, \text{Gen}(\cdot), \text{Rep}(\cdot), h(\cdot)\}$ in the smart card and BIO_i by making $Corrupt(\prod_{U_i}^{\alpha}, 1)$ and $Corrupt(\prod_{U_i}^{\alpha}, 2)$ queries. Next, the password PW_i needs to be guessed by $\mathcal{A}$. Since $\mathcal{A}$ can make q_s queries, so the probability which $\mathcal{A}$ imitates U_i successfully is $\frac{q_s}{|D|}$.

In addition, "false positive" may occur because the fuzzy extractor is used. Suppose $\mathcal{A}$ inputs the forged biometrics and the probability of deceiving reproduction algorithm $\text{Rep}(\cdot)$ of the fuzzy extractor is δ. Since $\mathcal{A}$ can make q_s queries, so the probability that $\mathcal{A}$ imitates U_i successfully is δq_s.

As soon as $Corrupt(\prod_{U_i}^{\alpha}, v)$ query is over, $\mathcal{A}$ makes $Test(\prod_{U_i}^{\alpha}, \prod_{SD_j}^{\gamma})$ query and determines whether the output of $Test(\prod_{U_i}^{\alpha}, \prod_{SD_j}^{\gamma})$ query is the real session key SK_j^* or a random string. However, the above cases cannot exist simultaneously. Therefore, it follows that

$$|\Pr[\text{Suc}_3] - \Pr[\text{Suc}_4]| \leq \max(\frac{q_s}{|N|}, \frac{q_s}{|D|}, \delta q_s). \tag{7}$$

At last, all the oracles have been modeled in the last game. $\mathcal{A}$ will win the game if $\mathcal{A}$ guess the bit b successfully. Since $\mathcal{A}$ has no other knowledge of the bit b, it follows that $\Pr[Suc_4] = \frac{1}{2}$. From (3)–(7), the following results can be obtained

$$\begin{aligned}
\text{Adv}_{\mathcal{A}}^{\Sigma}(t) &= 2 \cdot |\Pr[\text{Suc}_0] - \frac{1}{2}| = 2 \cdot |\Pr[\text{Suc}_1] - \Pr[\text{Suc}_4]| \\
&\leq 2\left(\frac{q_h^2}{2|\text{Hash}|} + \text{Adv}_{\mathcal{A}}^{\text{ECCDH}}(t') \right. \\
&\qquad \left. + \max\left(\frac{q_s}{|N|}, \frac{q_s}{|D|}, \delta q_s\right)\right) \\
&= \frac{q_h^2}{|\text{Hash}|} + 2 \cdot \text{Adv}_{\mathcal{A}}^{\text{ECCDH}}(t') \\
&\qquad + 2 \cdot \max\left(\frac{q_s}{|N|}, \frac{q_s}{|D|}, \delta q_s\right).
\end{aligned} \tag{8}$$

C. Security Analysis

The security requirements are analyzed detailedly in this subsection.

Mutual Authentication: The message $M_2 = \text{Enc}_K(\text{UID}_i, e)$ sent from U_i to GW is encrypted by employing the symmetric key $K = h((a+d) \cdot S)$, and only the person who has the private key s of GW can compute the same symmetric key $K = h(sM_1)$, which achieves the authentication of U_i to GW. GW computes $a' = h(\text{UID}_i||s||r_i)$, $e' = h(\text{UID}_i||\text{GID}||a'||TS_1)$ with UID_i and checks whether $e' = e$ to achieve authentication of GW to U_i. GW and $\{SD_1, SD_2, \ldots, SD_n\}$ achieve mutual authentication through TSK and sk_j. U_i and $\{SD_1, SD_2, \ldots, SD_n\}$ can authenticate each other with the help of GW. As a result, the mutual authentication among U_i, GW, and $\{SD_1, SD_2, \ldots, SD_n\}$ are achieved in the proposed scheme.

Anonymity: The real identity UID_i of U_i is hidden in the ciphertext $M_2 = \text{Enc}_K(\text{UID}_i, e)$, where $K = h((a+d) \cdot S)$, $M_1 = (a+d) \cdot P$, a is the long-term secret of U_i and d is a random number. If $\mathcal{A}$ wants to obtain UID_i, he needs to know $(a+d)$ or s and compute $K = h((a+d) \cdot S) = h(sM_1)$. However, due to the difficulty of the ECCDH problem, $\mathcal{A}$ cannot compute K and obtain UID_i. Similarly, the identity SID_j of the smart device SD_j is hidden in the ciphertext $M_6^j = \text{Enc}_{\text{TSK}}(\text{SID}_j, c_j)$.

If $\mathcal{A}$ wants to obtain SID_j, he needs to know the common temporary key TSK. However, TSK is only owned by GW and the deployed smart devices $\{SD_1, SD_2, \ldots, SD_n\}$. It is obvious that $\mathcal{A}$ is unable to know TSK. Consequently, the proposed scheme guarantees user and smart device anonymity.

Untraceability: The message $\{M_1, M_2, M_3, TS_1\}$ is different in each session because random number d and timestamp TS_1 are used. Therefore, $\mathcal{A}$ is unable to trace the behavior of U_i. Analogously, in each session, SD_j chooses different random number t_j and timestamp TS_3^j to compute the message $\{M_6^j, M_7^j, TS_3^j\}$. Thus, it is impossible for $\mathcal{A}$ to trace the behavior of SD_j and the proposed scheme provides user and smart device untraceability.

Perfect Forward Secrecy: Supposing $\mathcal{A}$ obtains U_i's long-term secret a and the latest session key SK, but the information constituting SK is encrypted by the symmetric key composed of different random numbers in each session, so previous session keys cannot be obtained. Similarly, assuming $\mathcal{A}$ gets the private key sk_j of SD_j, the latest common temporary key TSK and previous messages, however, the secret information with regard to the session key in each session is encrypted by different TSK. Due to the preimage-resistance of the hash function, $\mathcal{A}$ cannot recover the previous TSK and get secret information about the session key. Hence, the proposed scheme ensures perfect forward secrecy.

Resist Smart Card Stolen Attack: If $\mathcal{A}$ obtains the smart card of a registered user U_i, he is capable of extracting the data $\{A, B, P, RP_i, l_0, \text{Gen}(\cdot), \text{Rep}(\cdot), h(\cdot)\}$ using side-channel attack [45], where $A = a \oplus h(\text{UID}_i \oplus PW_i \oplus BK_i)$, $B = h(\text{UID}_i||PW_i||BK_i) \bmod l_0$. However, $\mathcal{A}$ does not know UID_i, PW_i, and BK_i which are secret information of U_i. Hence, $\mathcal{A}$ cannot obtain the long-term secret a to impersonate U_i. In this way, the proposed scheme can resist the smart card stolen attack.

Resist Impersonation Attack:

1) *Resist the User Impersonation Attack:* If $\mathcal{A}$ wants to impersonate U_i to initiate key agreement request, he must know the long-term secret a of U_i. However, a is protected by UID_i, PW_i, and BK_i which are only known by U_i. Accordingly, the user impersonation attack can be withstood.
2) *Resist the Gateway Impersonation Attack:* GW computes $M_{11} = \text{Enc}_K(M_8, SR)$ and $t_j = c_j \oplus sk_j$ during the authentication and key agreement phase, where $K = h(sM_1)$, $sk_j = h(SID_j||s)$. If $\mathcal{A}$ tries to create valid messages on behalf of GW, he must obtain the private key s of GW. Clearly, $\mathcal{A}$ is unable to get s. Hence, the proposed scheme is secure against the gateway impersonation attack.
3) *Resist the Smart Device Impersonation Attack:* If $\mathcal{A}$ wishes to imitate SD_j, he has to know the common temporary key TSK and the private key sk_j of SD_j. Obviously, $\mathcal{A}$ cannot get this information. Thus, the smart device impersonation attack can be resisted.

Resist Replay Attack: The current timestamp is included in all messages. If $\mathcal{A}$ tries to send the previous messages, which can check out the replay attack because the maximum delay is exceeded. That is, the proposed scheme is capable of protecting the replay attack.

Resist Man-in-the-Middle Attack: The goal of $\mathcal{A}$ is to modify the messages to cheat U_i, GW, and $\{SD_1, SD_2, \ldots, SD_n\}$ in this attack. When $\mathcal{A}$ wants to modify msg_1, where $M_2 = \text{Enc}_K(\text{UID}_i, e)$, $e = h(\text{UID}_i||\text{GID}||a||TS_1)$, and $K = h((a + d) \cdot S)$, he must know the long-term secret a and the random number d which are only known to U_i. Obviously, $\mathcal{A}$ cannot obtain this information. Analogously, $\mathcal{A}$ cannot modify legal messages msg_2, msg_3^j, msg_4, and msg_5. Therefore, the man-in-the-middle attack is able to be withstood.

Resist Denial-of-Service Attack: In the proposed scheme, the three-factor authentication is used. U_i can initiate authentication and key agreement request only if the correct UID_i, PW_i, and BIO_i are entered. Meanwhile, all messages contain timestamps, only fresh messages will be accepted. Consequently, the denial-of-service attack would be resisted.

Resist Known Session-Specific Temporary Information Attack: The master session key $SK = h(\text{UID}_i||\text{GID}||e||M_8) \cdot SR$ contains the secret information UID_i, e, M_8, and SR, which are encrypted by $K = h((a + d) \cdot S)$. Even if the random number d is exposed, $\mathcal{A}$ cannot compute K without the long-term secret a. Similarly, even if a is exposed, $\mathcal{A}$ cannot compute K without d. As a result, $\mathcal{A}$ cannot compute SK. Hence, the proposed scheme can prevent the known session-specific temporary information attack.

VII. Performance Evaluation

This section compares the proposed scheme with the related schemes [18], [23], [25], [28], [31], [32] in terms of security and functionality features, computation and communication costs.

A. Security and Functionality Features

Comparison between our proposed scheme and the related schemes [18], [23], [25], [28], [31], [32] regarding security and functionality features is provided in Table II. From Table II, the schemes proposed by [18], [23], [25], [28] are both one user and one smart device AKA schemes. In addition, Zhang et al.'s scheme [18], Li et al.'s scheme [23], and Shin et al.'s scheme [25] do not consider smart device join/leave. This is crucial, as the number of smart devices will change in accordance with the demand. Zhang et al.'s scheme [18] and Li et al.'s scheme [23] do not consider user revocation, which is a basic feature since smart cards are easily lost or stolen in real world scenarios. Li et al.'s scheme [23] and Shin et al.'s scheme [25] are insecure against known session-specific temporary information attack. A key flaw in Zhang et al.'s scheme [18] is that it does not support password and biometrics update. This is a crucial feature that must be considered, as passwords and biometrics should be regularly changed to ensure security. Li et al.'s scheme [23] cannot resist denial-of-service attack. Although Cui et al.'s scheme [31] and Vinoth et al.'s scheme [32] are AKA schemes for one user and multiple smart devices, their schemes are unable to withstand known session-specific temporary information attack, and user revocation cannot be realized.

TABLE II
COMPARISON OF SECURITY AND FUNCTIONALITY FEATURES

Schemes	Zhang *et al.*'s scheme [18]	Li *et al.*'s scheme [23]	Shin *et al.*'s scheme [25]	Far *et al.*'s scheme [28]	Cui *et al.*'s scheme [31]	Vinoth *et al.*'s scheme [32]	The proposed scheme
SF_1	✓	✓	✓	✓	✓	✓	✓
SF_2	✓	✓	✓	✓	✓	✓	✓
SF_3	✓	✓	✓	✓	✓	✓	✓
SF_4	✓	✓	✓	✓	✓	✗	✓
SF_5	✓	✓	✓	✓	✓	✓	✓
SF_6	✓	✓	✓	✓	✓	✓	✓
SF_7	✓	✓	✓	✓	✓	✓	✓
SF_8	✓	✓	✓	✓	✓	✓	✓
SF_9	✓	✓	✓	✓	✓	✓	✓
SF_{10}	✓	✓	✓	✓	✗	✓	✓
SF_{11}	✓	✗	✓	✓	✓	✓	✓
SF_{12}	✓	✗	✗	✓	✗	✗	✓
SF_{13}	N/A	N/A	N/A	✓	N/A	✓	✓
SF_{14}	N/A	✓	✓	✓	✓	✓	✓
SF_{15}	N/A	N/A	✓	✓	N/A	N/A	✓
SF_{16}	✓	✓	✓	✓	✓	✗	✓
SF_{17}	✗	✗	✗	✗	✓	✓	✓

Note: SF_1: Mutual authentication. SF_2: Anonymity. SF_3: Untraceability. SF_4: Perfect forward secrecy. SF_5: Smart card stolen attack. SF_6: User impersonation attack. SF_7: Gateway impersonation attack. SF_8: Smart device impersonation attack. SF_9: Replay attack. SF_{10}: Man-in-the-middle attack. SF_{11}: Denial of service attack. SF_{12}: Known session-specific temporary information attack. SF_{13}: Smart device join/leave. SF_{14}: Password and biometrics update. SF_{15}: User revocation. SF_{16}: Different session keys. SF_{17}: One-to-many scheme. "✓" means the scheme satisfies the functionality/security feature. "✗" means the scheme does not satisfy the functionality/security feature. "N/A" means not considered.

TABLE III
RUNTIME OF CRYPTOGRAPHIC OPERATIONS (MILLISECOND)

Notations	Descriptions	Runtime
T_h	General hash operation	0.0013
$T_{m-\mathbb{G}}$	Scalar multiplication operation in $\mathbb{G}$	0.3851
T_m	multiplication operation in $\mathbb{Z}_q^*$	0.0044
T_{se}	Symmetric encryption operation (AES-128)	0.0024
T_{sd}	Symmetric decryption operation (AES-128)	0.0028

Moreover, Cui et al.'s scheme [31] lacks a way to prevent man-in-the-middle attack and does not consider smart device join/leave. Vinoth et al.'s scheme [32] does not provide perfect forward secrecy and the agreed session keys are the same. In comparison, all security and functionality features can be met in the proposed scheme.

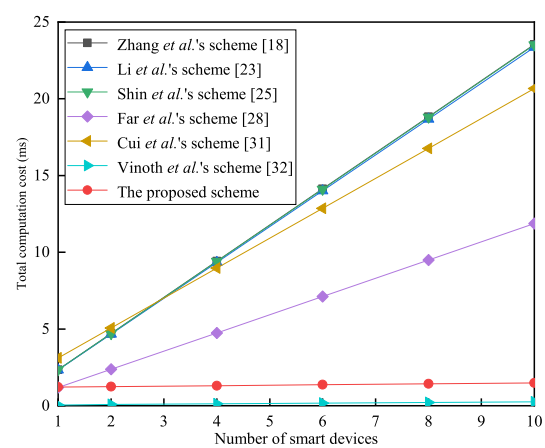

Fig. 4. Total computation cost versus Number of smart devices.

B. Computation Cost

The computation cost will be analyzed and compared between the proposed scheme and the related schemes [18], [23], [25], [28], [31], [32] in this subsection. To achieve 80-bit security level, an additive group $\mathbb{G}$ with prime order q is chosen by non-singular elliptic curve $E : y^2 = x^3 + ax + b \bmod p$, in which p, q are both 160-bit prime numbers and $a = -3$, b is a random 160-bit prime number. T_h, $T_{m-\mathbb{G}}$, T_m, T_{se}, and T_{sd} are used to denote the runtime of general hash operation, scale multiplication operation in $\mathbb{G}$, multiplication operation in $\mathbb{Z}_q^*$, symmetric encryption and decryption operations (AES-128), respectively. Addition and XOR operations are ignored because they are lightweight compared with other operations. The experimental environment for evaluating cryptography operations is 64-bit Windows 10 operating system with 2.53 GHz using MIRACL Crypto SDK [46], in which CPU is i5 and memory is 4 GB. Table III manifests the average runtime of related operations running 10 000 times.

Table IV exhibits the comparison result in the field of computation cost between the proposed scheme and the related schemes [18], [23], [25], [28], [31], [32]. The total computation cost of accessing one smart device for the related schemes [18], [23], [25], [28], [31], [32] and the proposed scheme are 2.3548, 2.3353, 2.3509, 1.1865, 3.1133, 0.0579, and 1.2218 ms, respectively. Since less scalar multiplication operation in $\mathbb{G}$ and symmetric encryption/decryption operation (AES-128) are adopted, the computational cost of the proposed scheme is smaller than that of the related schemes [18], [23], [25], [31]. Although the computational cost of the related schemes [28], [32] is smaller than that of the proposed scheme, however, their schemes cannot resist some common attacks and support the most basic functions. The total computation cost of accessing n smart devices for the related schemes [18], [23], [25], [28], [31], [32] and the proposed scheme are $2.3548n$, $2.3353n$, $2.3509n$, $1.1865n$, $1.9502n + 1.1631$, $0.0216n + 0.0363$, and $0.0285n + 1.1933$ ms, individually.

Fig. 4 shows the relation between the total computation cost and the number of smart devices. It can be seen from Fig. 4 that in the multismart device scenario, the computation cost of the related schemes [18], [23], [25], [28], [31] are greater than that

TABLE IV
COMPARISON OF COMPUTATION COST

Schemes	Computation cost (ms)			Total computation cost (ms)	
	U_i	GW	SD_j	*one* smart device	n smart devices
Zhang *et al.*'s scheme [18]	$7T_h+3T_{m-\mathbb{G}}+T_{se}=1.1668$	$9T_h+T_{m-\mathbb{G}}+T_{se}+2T_{sd}=0.4048$	$6T_h+2T_{m-\mathbb{G}}+T_{se}+T_{sd}=0.7832$	$22T_h+6T_{m-\mathbb{G}}+3T_{se}+3T_{sd}=2.3548$	$(22T_h+6T_{m-\mathbb{G}}+3T_{se}+3T_{sd})n=2.3548n$
Li *et al.*'s scheme [23]	$8T_h+3T_{m-\mathbb{G}}=1.1657$	$7T_h+T_{m-\mathbb{G}}=0.3942$	$4T_h+2T_{m-\mathbb{G}}=0.7754$	$19T_h+6T_{m-\mathbb{G}}=2.3353$	$(19T_h+6T_{m-\mathbb{G}})n=2.3353n$
Shin *et al.*'s scheme [25]	$14T_h+3T_{m-\mathbb{G}}=1.1735$	$12T_h+T_{m-\mathbb{G}}=0.4007$	$5T_h+2T_{m-\mathbb{G}}=0.7767$	$31T_h+6T_{m-\mathbb{G}}=2.3509$	$(31T_h+6T_{m-\mathbb{G}})n=2.3509n$
Far *et al.*'s scheme [28]	$9T_h+2T_{m-\mathbb{G}}=0.7819$	$10T_h+T_{m-\mathbb{G}}=0.3981$	$5T_h=0.0065$	$24T_h+3T_{m-\mathbb{G}}=1.1865$	$(24T_h+3T_{m-\mathbb{G}})n=1.1865n$
Cui *et al.*'s scheme [31]	$8T_h+3T_{m-\mathbb{G}}=1.1657$	$10T_h+2T_{m-\mathbb{G}}=0.7832$	$7T_h+3T_{m-\mathbb{G}}=1.1644$	$25T_h+8T_{m-\mathbb{G}}=3.1133$	$(19T_h+5T_{m-\mathbb{G}})n+6T_h+3T_{m-\mathbb{G}}=1.9502n+1.1631$
Vinoth *et al.*'s scheme [32]	$9T_h+T_{sd}=0.0145$	$6T_h+4T_m+2T_{se}+T_{sd}=0.033$	$4T_h+T_{se}+T_{sd}=0.0104$	$19T_h+4T_m+3T_{se}+3T_{sd}=0.0579$	$(4T_h+2T_m+2T_{se}+T_{sd})n+15T_h+2T_m+T_{se}+2T_{sd}=0.0216n+0.0363$
The proposed scheme	$8T_h+2T_{m-\mathbb{G}}+T_m+T_{se}+T_{sd}=0.7902$	$11T_h+T_{m-\mathbb{G}}+T_m+2T_{se}+2T_{sd}=0.4142$	$6T_h+T_m+T_{se}+T_{sd}=0.0174$	$25T_h+3T_{m-\mathbb{G}}+3T_m+4T_{se}+4T_{sd}=1.2218$	$(9T_h+2T_m+T_{se}+2T_{sd})n+16T_h+3T_{m-\mathbb{G}}+T_m+3T_{se}+2T_{sd}=0.0285n+1.1933$

TABLE V
COMPARISON OF COMMUNICATION COST

Schemes	Communication cost (bits)			Total communication cost (bits)	
	U_i	GW	SD_j	*one* smart device	n smart devices
Zhang *et al.*'s scheme [18]	864	512	1600	2976	$2976n$
Li *et al.*'s scheme [23]	640	960	480	2080	$2080n$
Shin *et al.*'s scheme [25]	992	1184	512	2688	$2688n$
Far *et al.*'s scheme [28]	960	1440	480	2880	$2880n$
Cui *et al.*'s scheme [31]	672	832	1184	2688	$2176n+512$
Vinoth *et al.*'s scheme [32]	512	2144	672	3328	$320n^2+1664n+1344$
The proposed scheme	672	1536	512	2720	$1696n+1024$

of the proposed scheme. Specifically, the proposed scheme is reduced by 93.7%, 93.7%, 93.7%, 87.5%, and 92.8% compared with the related schemes [18], [23], [25], [28], [31], respectively. The scheme [32] of Vinoth et al. is slightly better than our work in the computation cost. The degradation is regarded as reasonable since the proposed scheme can resist known session-specific temporary information attack, and achieves perfect forward secrecy and user revocation as well.

C. Communication Cost

The communication cost of the proposed scheme and related schemes [18], [23], [25], [28], [31], [32] will be analyzed and compared in this subsection. As previously mentioned, the length of elements in $\mathbb{G}$, the length of elements in $\mathbb{Z}_q^*$, the ciphertext length of the symmetric encryption, the identity, the output of hash function, and the timestamp are 160, 160, 128, 160, 160, and 32 bits, respectively.

Table V indicates the comparison result of the proposed scheme and the related schemes [18], [23], [25], [28], [31], [32] in relation to communication cost. The total communication cost of accessing one smart device for the related schemes [18], [23], [25], [28], [31], [32] and the proposed scheme are 2976, 2080, 2688, 2880, 2688, 3328, and 2720 bits, respectively. The total communication cost of accessing n smart devices for the related schemes [18], [23], [25], [28], [31], [32] and the proposed scheme are $2976n$, $2080n$, $2688n$, $2880n$, $2176n+512$, $320n^2+1664n+1344$, and $1696n+1024$ bits, individually.

Fig. 5 depicts the relationship between total communication cost and the number of smart devices. According to Fig. 5, as the number of smart devices increases, the difference in communication cost between related schemes [18], [23], [25], [28], [31], [32] and our proposed scheme also increases. Specifically, the proposed scheme is reduced by 39.6%, 13.5%, 33.1%, 37.6%, 19.3%, and 64.0% compared with the related schemes [18], [23], [25], [28], [31], [32], respectively, which shows that the proposed scheme is a better choice for multismart devices scenarios.

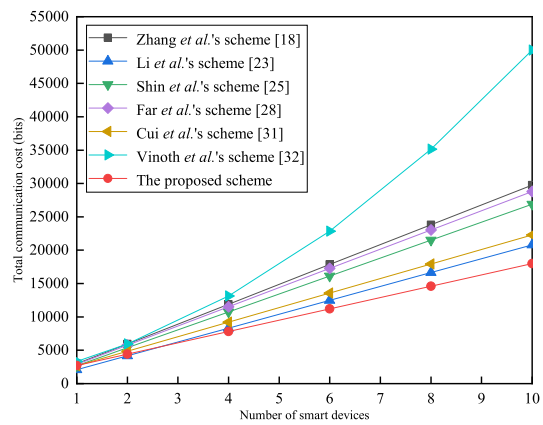

Fig. 5. Total communication cost versus Number of smart devices.

VIII. CONCLUSION

In this article, we have proposed a secure three-factor authentication and key agreement scheme which is suitable for multismart devices scenarios in IIoT by utilizing ECC and CRT. The proposed scheme has achieved mutual authentication and agreed to different session keys between one user and multiple smart devices simultaneously. Even if the secret information of the smart device and the latest session key are leaked, the previous session keys cannot be recovered due to the hash chain technology, which protects the security of the previously transmitted data. Additionally, the proposed scheme also accommodates password and biometrics update, smart device join and leave, and user revocation. The security of the proposed scheme is proven in the ROR model. Finally, the performance evaluations have demonstrated that the proposed scheme is significantly superior to related schemes in terms of computation and communication costs. In the future, we plan to expand the user side of the proposed scheme and design a secure many-to-many authentication and key agreement scheme for Industrial IoT, which can achieve secure communication between multiple users and multiple smart devices.

References

[1] S. Rakas, V. Timčenko, M. Kabović, and A. Kabović, "Industrial Internet: Architecture, characteristics and implementation challenges," in *Proc. IEEE Int. Symp.*, 2021, pp. 1-4.

[2] I. Zhou et al., "Internet of things 2.0: Concepts, applications, and future directions," *IEEE Access*, vol. 9, pp. 70961–71012, 2020.

[3] L. Xia and S. Liu, "Intelligent IoT-based cross-border e-commerce supply chain performance optimization," *Wireless Commun. Mobile Comput.*, vol. 2021, no. 121, Jun. 2021, Art. no. 9961925.

[4] CESI, "Industrial Internet of Things white paper," Beijing, China, Sep. 2017, Accessed: Jun. 28, 2021. [Online]. Available: http://www.cesi.cn/images/editor/20170913/20170913114540317.pdf

[5] J. Holdren, T. Power, G. Tassey, A. Ratcliff, and L. Christodoulou, "A national strategic plan for advanced manufacturing," pp. 8–20, 2012.

[6] DIN, "German standardization roadmap industry 4.0 (version 4)," Berlin, Germany, Mar. 2020, Accessed: Jun. 28, 2021. [Online]. Available: http://www.din.de/blob/65354/f5252239daa596d8c4d1f24b40e4486d/roadmap-i4-0-e-data.pdf

[7] F. Tao and Q. Qi, "New IT driven service-oriented smart manufacturing: Framework and characteristics," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 49, no. 1, pp. 81–91, Jan. 2019.

[8] L. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Trans. Ind. Inform.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[9] I. Makhdoom, M. Abolhasan, J. Lipman, R. Liu, and W. Ni, "Anatomy of threats to the Internet of Things," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 2, pp. 1636–1675, Secondquarter 2019.

[10] G. Xu and B. Yu, "Security enhanced design of the bluetooth simple pairing protocol," in *Proc. IEEE Int. Conf. Comput. Sci. Netw. Technol.*, 2011, pp. 292–296.

[11] Y. Zhang, X. Huang, X. Chen, L. Zhang, J. Zhang, and Y. Xiang, "A hybrid key agreement scheme for smart homes using the Merkle puzzle," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1061–1071, Feb. 2020.

[12] H. Song, B. Wei, Q. Yu, X. Xiao, and T. Kikkawa, "WiEps: Measurement of dielectric property with commodity WiFi device-an application to ethanol/water mixture," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11667–11677, Dec. 2020.

[13] M. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1086–1090, Mar. 2009.

[14] D. He, Y. Gao, S. Chan, C. Chen, and J. Bu, "An enhanced two-factor user authentication scheme in wireless sensor networks," *Ad Hoc Sens. Wirl. Netw.*, vol. 10, no. 4, pp. 361–371, Feb. 2010.

[15] H. Yeh, T. Chen, P. Liu, T. Kim, and H. Wei, "A secured authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 11, no. 5, pp. 4767–4779, May 2011.

[16] W. Tai, Y. Chang, and W. Li, "An IoT notion-based authentication and key agreement scheme ensuring user anonymity for heterogeneous ad hoc wireless sensor networks," *J. Inf. Secur. Appl.*, vol. 34, no. 2, pp. 133–141, Jun. 2017.

[17] S. Shin and T. Kwon, "Two-factor authenticated key agreement supporting unlinkability in 5G-integrated wireless sensor networks," *IEEE Access*, vol. 6, pp. 11229–11241, 2018.

[18] J. Zhang, J. Cui, H. Zhong, I. Bolodurina, and L. Liu, "Intelligent drone-assisted anonymous authentication and key agreement for 5 G/B5G vehicular ad-hoc networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 4, pp. 2982–2994, Oct./Dec. 2021.

[19] C. Fan and Y. Lin, "Provably secure remote truly three-factor authentication scheme with privacy protection on biometrics," *IEEE Trans. Inf. Forensics Secur.*, vol. 4, no. 4, pp. 933–945, Dec. 2009.

[20] A. Das, "A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks," *Peer-to-Peer Netw Appl.*, vol. 9, no. 1, pp. 223–244, Jan. 2016.

[21] F. Wu, L. Xu, S. Kumari, and X. Li, "An improved and provably secure three-factor user authentication scheme for wireless sensor networks," *Peer-to-Peer Netw Appl.*, vol. 11, no. 5, pp. 1–20, Jan. 2018.

[22] M. Wazid, A. Das, V. Odelu, N. Kumar, M. Conti, and M. Jo, "Design of secure user authenticated key management protocol for generic IoT networks," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 269–282, Feb. 2018.

[23] X. Li, J. Niu, M. Bhuiyan, F. Wu, M. Karuppiah, and S. Kumari, "A robust ECC-based provable secure authentication protocol with privacy preserving for industrial Internet of Things," *IEEE Trans. Ind. Inform.*, vol. 14, no. 8, pp. 3599–3609, Aug. 2018.

[24] A. Jolfaei, M. Talouki, and S. Aghili, "Lightweight and anonymous three-factor authentication and access control scheme for real-time applications in wireless sensor networks," *Peer-to-Peer Netw Appl.*, vol. 12, no. 1, pp. 43–59, Jan. 2019.

[25] S. Shin and T. Kwon, "A privacy-preserving authentication, authorization, and key agreement scheme for wireless sensor networks in 5G-integrated Internet of Things," *IEEE Access*, vol. 8, pp. 67555–67571, 2020.

[26] Z. Yang, J. He, Y. Tian, and J. Zhou, "Faster authenticated key agreement with perfect forward secrecy for industrial Internet-of-Things," *IEEE Trans. Ind. Inform.*, vol. 16, no. 10, pp. 6584–6596, Oct. 2020.

[27] R. Ali, P. Chandrakar, M. Obaidat, K. Hsiao, A. Pal, and S. Islam, "A secure authentication mechanism for wireless sensor networks," in *Proc. IEEE Int. Conf. Comput., Inf. Telecommun. Syst.*, 2020, pp. 1–8.

[28] H. Far, M. Bayat, A. Das, M. Fotouhi, S. Pournaghi, and M. Doostari, "LAPTAS: Lightweight anonymous privacy-preserving three-factor authentication scheme for WSN-based IIoT," *Wireless Netw.*, vol. 27, pp. 1389–1412, Feb. 2021.

[29] C. Meshram, M. Obaidat, C. Lee, and S. Meshram, "An efficient, robust, and lightweight subtree-based three-factor authentication procedure for large-scale DWSN in random oracle," *IEEE Syst. J.*, vol. 15, no. 4, pp. 4927–4938, Dec. 2021.

[30] Y. Li and Y. Tian, "A lightweight and secure three-factor authentication protocol with adaptive privacy-preserving property for wireless sensor networks," *IEEE Syst. J.*, to be published, doi: 10.1109/JSYST.2022.3152561.

[31] J. Cui, X. Zhang, H. Zhong, J. Zhang, and L. Liu, "Extensible conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 1654–1667, 2020.

[32] R. Vinoth, L. Deborah, P. Vijayakumar, and N. Kumar, "Secure multifactor authenticated key agreement scheme for industrial IoT," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3801–3811, Mar. 2021.

[33] V. Miller, "Use of Elliptic Curves in Cryptography," in *Proc. Adv. Cryptology*, 1985, pp. 417–426.

[34] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, Jan. 1987.

[35] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[36] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. Adv. Cryptology*, 2004, pp. 523–540.

[37] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. London, U.K.: Chapman and Hall/CRC Cryptography and Network Security, 2007.

[38] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, Nov. 1981.

[39] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[40] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Adv. Cryptology*, 2002, pp. 337–351.

[41] A. Irshad et al., "An anonymous and efficient multiserver authenticated key agreement with offline registration centre," *IEEE Syst. J.*, vol. 13, no. 1, pp. 436–446, Mar. 2019.

[42] D. Yang and B. Yang, "A biometric password-based multi-server authentication scheme with smart card," in *Proc. IEEE Int. Conf. Comput. Des. Appl.*, 2010, pp. 554–559.

[43] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 708–722, Jul.-Aug. 2018.

[44] M. Abdalla, P. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. Int. Workshop Public Key Cryptography*, Springer, 2005, pp. 65–84.

[45] T. Messerges, E. Dabbish, and R. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[46] Shamus SoftwareDublin, Ireland, "Multi precision integer and rational arithmetic cryptographic library (MIRACL)," *Accessed*: Jun. 10, 2021. [Online]. Available: http://www.certivox.com/miracl/