electronics

Article

Secure Authentication and Key Agreement Protocol for Cloud-Assisted Industrial Internet of Things

Huanhuan Hu [1], Longxia Liao [1] and Junhui Zhao [1,2,*]

[1] School of Information Engineering, East China Jiaotong University, Nanchang 330013, China; huanhuan_hu@hotmail.com (H.H.); liaolxcl@163.com (L.L.)
[2] School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
* Correspondence: junhuizhao@hotmail.com

Abstract: With the expansion of the Industrial Internet of Things (IIoT), real-time data collected by smart sensors deployed in factories are shared over open channels , which may cause unauthorized access of transmitted messages by adversaries, thus causing the problem of privacy leakage. User authentication is the first line of defense for security protection in the IIoT environment. In this paper, we propose a cloud—assisted authentication scheme based on Chebyshev polynomial encryption, in which only authorized users can access the sensing devices in the Internet of Things (IoT) to obtain real-time data. The scheme uses fuzzy extraction technology to verify biometric characteristics. There are three factors to verify the user's login request: the smart card, password and the user's personal biometrics. The commonly adopted formal security analysis, the ROR model, is applied to prove the semantic security of session key, and a detailed informal security analysis is performed to show that the proposed scheme can withstand multiple known attacks. Compared with other related user authentication schemes, the proposed scheme provides several extra functionality features, including offline sensor node registration, updating user passwords and biometrics, adding new sensor node deployment, user anonymity and untraceability. In addition, the cost of computation, communication and security is compared with similar schemes, and results show that our scheme has more security performance while the cost is acceptable.

Keywords: key agreement; protocol; sensor; authentication; security

Citation: Hu, H.; Liao, L.; Zhao, J. Secure Authentication and Key Agreement Protocol for Cloud-Assisted Industrial Internet of Things. *Electronics* **2022**, *11*, 1652. https://doi.org/10.3390/electronics11101652

Academic Editor: Nurul I. Sarkar

Received: 20 April 2022
Accepted: 17 May 2022
Published: 22 May 2022

1. Introduction

The Internet of Things (IoT) uses sensors to connect any device over the Internet to share information, with the growth of the number of mobile devices and the development of 6G technology (which, in comparison with 5G technology, has greatly improved in data rate and transmission delay [1,2], making it more suitable for the application of the IoT) [3], IoT can connect to billions of devices and to provide the foundation for systems such as telemedicine, smart homes, and industrial monitoring [4]. However, sensors with limited computing and storage resources cannot support the utilization of large quantity of heterogeneous data generated by a massive amount of IoT devices. In order to avoid resource waste and make better use of the data obtained by devices in the IoT in the monitoring or production process, the sensor can periodically transfer the data to the cloud for storage, so that users can use the strong computing power of cloud servers to analyze and process the data. In addition, cloud computing technology [5] can provide almost unlimited computing power and storage to compensate for resource limitations of the IoT [6]. At the same time, IoT also brings a lot of real data to cloud computing. The merger of cloud computing and the IoT brings new opportunities and security challenges, such as information leaks caused by data stored in cloud servers being accessed by illegal attackers. Among all kinds of security measures, user authentication is an effective method to ensure system security.

The Industrial Internet of Things (IIoT) is one of the most important features of the development of industry 4.0 in the 21st century and also a major driving force for the

application and promotion of the IoT. It has been widely used in many industries, including medical automation, transportation, and environmental protection. For example, it can help realize real-time monitoring, vehicle tracking, and intelligent parking in the field of intelligent transportation. In addition to the above applications, IIoT has been widely used to improve the production efficiency of industrial products and help enterprises utilize automatic and intelligent manufacturing. In the process of industrial production, we use cloud computing technology to process and store a large amount of data generated by sensors, so as to extract the results of industrial production guidance suggestions to improve production efficiency. However, the secure production of the IIoT is faced with two key problems: first, data stored on cloud servers may be accessed illegally by competitors, which leads to information leakage; and, second, when users need to access information, we need to provide users with privacy protection and achieve user anonymity.

1.1. Motivation

In the IIoT environment, sensors deployed in the factory can detect the surrounding environment in real time, and the sensors upload the monitored data to the gateway node (GWN) through the public network. However, the data are transmitted through open (insecure) channels, which means that all kinds of attacks are possible [7]. For example, attackers can eavesdrop, modify, intercept communication messages, or impersonate trusted entities, which threatens the security and privacy of messages. Furthermore, there is the possibility of physical and irreparable damage, such as an adversary issuing malicious instructions to the sensor. The design of an authentication protocol can ensure the mutual authentication of communication participants in an open and chaotic network environment, and establish an exclusive session key for secure access later.

This paper designs a user authentication scheme based on Chebyshev polynomial encryption for IIoT constructs mutual authentication among communication participants, and establishes session keys between users and sensors. The proposed scheme uses the user's biometric characteristics, password, and smart card as three factors to authenticate legitimate users. At the same time, the protocol needs to be able to resist some common attacks and can guarantee the security of session keys even when temporary secret values are disclosed.

1.2. System Model

Network models and threat models are described for cloud-assisted IIoT systems in this section.

1.2.1. Network Model

The cloud-assisted IIoT model is shown in Figure 1, and the main components are: sensors, base stations, gateway node, users, and the cloud server. We deploy a set of sensors in the target area of the factory to monitor the environment, collect information about the production of the product or the operation of the production equipment, etc. Then, sensors send the collected information to GWN through the base station [8]. GWN sends the massive, heterogeneous data that it collects to the cloud server. Meanwhile, in this factory, there will be a need for users of different identities to access the data in the sensors in real time, such as the personnel on duty who can detect the sensors to determine whether the factory equipment is abnormal, or the manager who can use the information to make decisions or analysis.

The process of information transmission through the public channel is as follows: firstly, the user sends the login information to GWN, and GWN verifies the validity of the user's identity. Secondly, if the authentication is successful, GWN sends the encrypted authentication information to the sensor, which returns the key exchange information to the user. Finally, the session key is generated for future communication between user and sensor.

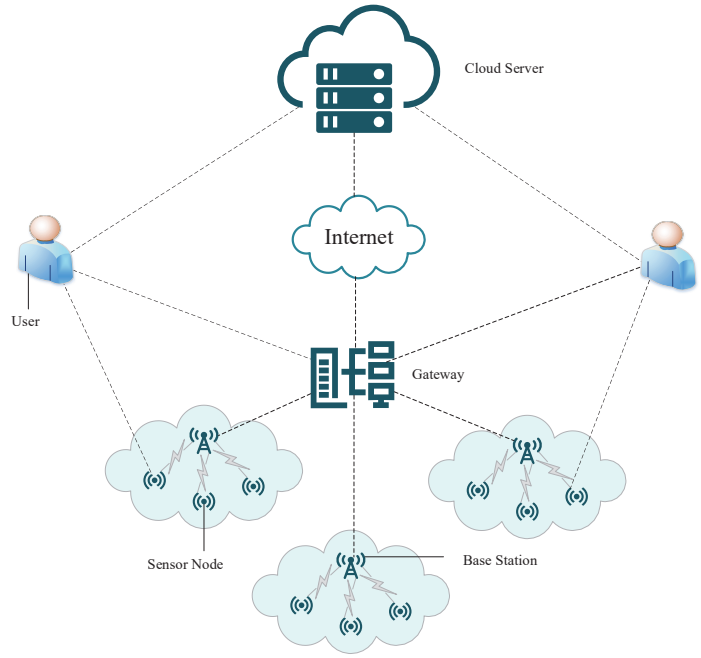

Figure 1. Cloud-assisted authentication model in IIoT.

In the cloud-assisted IIoT architecture, the sensor sends the collected information to *GWN*, which periodically uploads it to the cloud server via the Internet. It should be noted that the sensitive information uploaded is through a pre-shared security mechanism between *GWN* and the cloud server. In this model, by treating the cloud server as a node and granting a unique identity, and performing a sensor-like registration process, mutual identity authentication can be performed with users, and establish session keys with each other. After successful authentication, users can access massive amounts of data stored in the cloud server. This mechanism helps users to systematically analyze previous data and is also of great benefit to decision-making and forecasting.

1.2.2. Threat Model

The scheme uses the widely used Dolev–Yao (DY) threat model [9]. According to the definition of the model, the adversary is allowed to insert, delete, and modify the transmitted information on any two parties through a common channel. Due to some malicious attack, an adversary may be able to capture the sensors deployed in the factory and obtain information stored therein. *GWN* is believed to be a trusted third party entity in the model. Smart cards accidentally lost by users or stolen by adversaries can be used to perform sophisticated power analysis [10] to extract secret credentials stored in them. Additionally, the widely accepted ROR model uses rigorous mathematical analysis to verify the security of the scheme.

1.3. Contribution

The contributions of the paper are summarized as follows:

- A user authentication scheme based on Chebyshev polynomial encryption in cloud-assisted IIoT environment is proposed, which can help users and sensor nodes to complete mutual authentication and negotiate session keys. The scheme uses smart cards, passwords, and biometrics to detect the identity of legitimate users. Furthermore, the scheme also allows users to update passwords and biometrics locally, revoke smart cards, and deploy new sensors.
- Mutual identity authentication and session key agreement can be achieved between the cloud server and users, which can prevent unauthorized users from accessing resources from the cloud server.

- The security of the scheme is proved by formal (mathematical) and informal security analysis. The formal security proves the semantic security of the proposed scheme and guarantees the security of the session key between the user and the sensor node (described in Section 4.1). In addition, the proposed scheme is demonstrated to be secure against other possible multiple known attacks through the use of informal security analysis (discussed in Section 4.2).
- The proposed scheme has advantages in security performance and functional features (described in Section 5) under the premise of lower cost.

1.4. Paper Outline

Section 2 gives a brief overview of the related work. Section 3 describes the mathematical preliminaries used in the protocol and the detailed phases of the proposed scheme. Section 4 provides the ROR model proof and informal security analysis . Section 5 gives the comparison between the proposed scheme and the related schemes in terms of overhead, performance, and functional features. Section 6 presents the conclusions.

2. Related Work

Das [11] mainly summarized some security defects and inefficient problems existing in Jiang et al. [12] 's scheme and improved it on the basis, making it more applicable to actual scenarios. Chang and Le [13] designed two authentication protocols for sensing devices with limited resources, therein which $\mathcal{P}1$ is lightweight because only hashing and XOR operations are used, and $\mathcal{P}2$ uses the elliptic curve encryption (ECC) algorithm, $\mathcal{P}2$ has higher cost, but can also meet higher security features. Li et al. [14] proposed an authentication scheme to protect user anonymity in the IIoT, which applied biometric fuzzy extractor features and solved the problem of fault tolerance in user biometric information extraction. Wazid et al. [15] proposed a solution to meet various safety performances in a smart home environment, which used only some lightweight operations and is ideal for smart appliances with limited computing resources.

Das et al. [16] designed an authentication protocol to enable users to access data stored in wearable devices in real time, and the simulation results of various network parameters (such as delay, throughput, etc.) display that can be used in real scenarios. Liu et al. [17] proposed a protocol based on cross authentication, which realized security authentication and synchronous authentication at the same time, and can integrate two related devices into one session. However, Srinivas et al. [18] pointed out that Liu et al. [17] did not have functions such as dynamic wearable devices and changing users' mobile terminals. In order to eliminate these limitations, Srinivas et al. [18] proposed an authentication protocol applicable to the medical monitoring system. In this scheme, user registration was performed in the big data registry, and the authentication process with wearable devices was completed with the assistance of the cloud center.

Chatterjee et al. [19] designed an authentication scheme based on Chebyshev chaotic mapping, biological hashing, and symmetric key encryption/decryption for multi-server environments, which allowed users to manage authentication for different servers with a single identity and password. Srinivas et al. [20] proposed an anonymous user authentication scheme using fuzzy extraction technology for biometric authentication in IIoT. Three-factor authentication was used in [19,20]. Yu et al. [21] introduced an effective universal three-factor authentication framework, which can upgrade two-factor to three-factor authentication on the premise of maintaining user anonymity, and improve the security of the system. Wazid et al. [22] proposed a solution that would allow real-time access to data monitored by unmanned aerial vehicle. Chatterjee et al. [23] have developed an authentication protocol that uses identity-based encryption, physical non-clonable function (PUF), and hash function, and a secure video surveillance camera model has been established based on this protocol.

To sum up, most of the user authentication schemes proposed for the IoT have some defects, such as insecurity against various known attacks and inability to provide security protection for user anonymity. Most of the encryption technologies used in existing schemes

are hash functions or XOR operations, which are lightweight but not secure, while ECC algorithms require more overhead. At the same time, we also noticed that these solutions support users to access the sensor in real time, but do not take into account the data storage generated by the sensor, which cannot provide users with previous data for analysis, resulting in a waste of resources. Inspired by the above references and [24], IIoT needs a user authentication scheme, compared with other user authentication protocols before, and the proposed scheme can provide higher safety and extra features while meeting lower overhead, and the data generated by the sensor can be stored, for example, uploaded to the cloud server. After mutual authentication between the user and the cloud server, users can securely access the cloud server.

3. The Proposed Scheme

The proposed scheme is a three-party authentication protocol based on Chebyshev polynomial encryption, and includes three stages: sensor node registration, user registration, and authentication key exchange. In this section, we first give the mathematical preparations used in this paper, and then present the detail process of the proposed scheme. and Table 1 introduces the mathematical notation in the scheme.

Table 1. The mathematical symbols.

Symbol	Description
ID_{GWN}	Identity of GWN
U_i,SN_j	ith user and jth sensor, respectively
ID_i,ID_{SNj}	Identity of U_i and identity of SN_j, respectively
PW_i	Password of U_i
SC_i	smart card of U_i
BIO_i	Biometrics of U_i
σ_i, τ_i	Biometric secret keys of U_i and public replication parameters, respectively
$Gen(\cdot)$, $Rep(\cdot)$	Fuzzy extractor generation and reproduction methods, respectively
t	Fuzzy extractor threshold parameter
$T_n(x)$	Chebyshev polynomial of degree n
$\|, \oplus$	Bitwise XOR & concatenation, respectively
$h(\cdot)$	hash function
$\mathcal{A}$	Adversary

3.1. One-Way Hash Function

$h : \{0,1\}^* \rightarrow \{0,1\}^n$ is a one-way hash function that compresses an input message of arbitrary length into a fixed-length output [25].

Definition 1. *$Adv_{\mathcal{A}}^{hash}(rt) = \Pr[(n_1, n_2) \in_R \mathcal{A} : n_1 \neq n_2, h(n_1) = h(n_2)]$, where the input pair $(n_1, n_2) \in_R \mathcal{A}$ indicates that $\mathcal{A}$ randomly selects two numbers n_1 and n_2, $Adv_{\mathcal{A}}^{hash}(rt)$ is denoted as the probability that $\mathcal{A}$ finds a hash conflict within the running time rt. An $(\xi, rt)-$ adversary $\mathcal{A}$ attacking the collision resistance of $h(\cdot)$, which implies that the running time of $\mathcal{A}$ is at most rt and $Adv_{\mathcal{A}}^{hash}(rt) \leq \xi$.*

3.2. Chebyshev Polynomial

Chebyshev polynomials not only have much lower computational overhead than traditional ECC algorithms, but also meet our security requirements [19,20]. In this paper, the basic concepts of Chebyshev polynomials are introduced to design our scheme. The recursion of the improved Chebyshev polynomial is shown below:

$$T_n(x) = \begin{cases} 1 & n = 0 \\ x & n = 1 \\ 2xT_{n-1}(x) - T_{n-2}(x) & n \geq 2. \end{cases}$$

where $x \in (-\infty, +\infty)$, $T_u(T_v(x)) \equiv T_{uv}(x) \equiv T_v(T_u(x)) (\mathrm{mod}\ p)$, and p is a large prime number, the computational problem associated with Chebyshev polynomial is in Definition 2.

Definition 2. *Given any x and y, it is hard to find an integer n that satisfies* $T_n(x) = y(\mathrm{mod}\ p)$. *It is known as the chaotic map-based discrete logarithm problem (CMDLP). The probability advantage of* $\mathcal{A}$ *associated with CMDLP is*

$$Adv_{\mathcal{A}}{}^{CMDLP}(t_2) = \Pr\left[\mathcal{A}(x,y) = n : n \in Z_p{}^*, y = T_n(x)(\mathrm{mod}\ p)\right]$$

where $Z_p{}^* = \{r|0 < r < p, \gcd(r,p) = 1\} = \{1,2,\ldots,p-1\}$. $Pr[E]$ *is the probability of an event E. An* (ς, t_1)*-adversary* $\mathcal{A}$ *breaking CMDLP means that* $Adv_{\mathcal{A}}{}^{CMDLP}(t_1) \leq \varsigma$ *at most within the running time* t_1.

3.3. Sensor Node Registration

During the sensor node registration phase, GWN selects a 160-bit random number as the master key s, and assigns the unique identity ID_{SN_j} to each deployed sensor node SN_j, then calculates temporal credential of ID_{SN_j} as $TC_{SN_j} = h\left(ID_{SN_j} \parallel s\right)$. GWN stores $\left\{TC_{SN_j}, ID_{SN_j}\right\}$ into SN_j's memory before it was deployed to the IoT environment, and stores information $\left\{ID_{GWN}, s, TC_{SN_j}, ID_{SN_j}\right\}$ into its database.

3.4. User Registration Phase

The process of registering a user with GWN is done offline, and the following is the process and is shown in Figure 2:

- $Reg1$: U_i chooses her/his own ID_i and PW_i, selects a 160-bit random numbers n_a, and calculates RID_i=$h(ID_i \parallel n_a)$ and $RPW_i = h(PW_i \parallel ID_i \parallel n_a)$, then securely transmits the registration request $\langle RID_i, RPW_i\rangle$ to GWN.
- $Reg2$: Upon receipt of the request from the U_i, GWN calculates $C_i = RID_i \oplus RPW_i \oplus h(s \parallel ID_{GWN})$ and distributes a smart card $SC_i = \{C_i,\ h(\cdot)\}$ to U_i through a secure channel.
- $Reg3$: U_i imprints her/his biometrics BIO_i on the sensor of the specified device after receiving SC_i, and computes $(\sigma_i, \tau_i) = Gen(BIO_i)$, $A_i = n_a \oplus h(ID_i \parallel \sigma_i)$, $B_i = h(ID_i \parallel PW_i \parallel \sigma_i)$, $C_i{}' = C_i \oplus h(\sigma_i \parallel ID_i)$. Then, U_i replaces C_i with $C_i{}'$ and stores A_i, B_i, $C_i{}'$, σ_i, τ_i, $Gen(\cdot)$, $Rep(\cdot)$ and t into SC_i, where $SC_i = \left\{A_i, B_i, C_i{}',\ h(\cdot), Gen(\cdot), Rep(\cdot), \tau_i, t\right\}$.

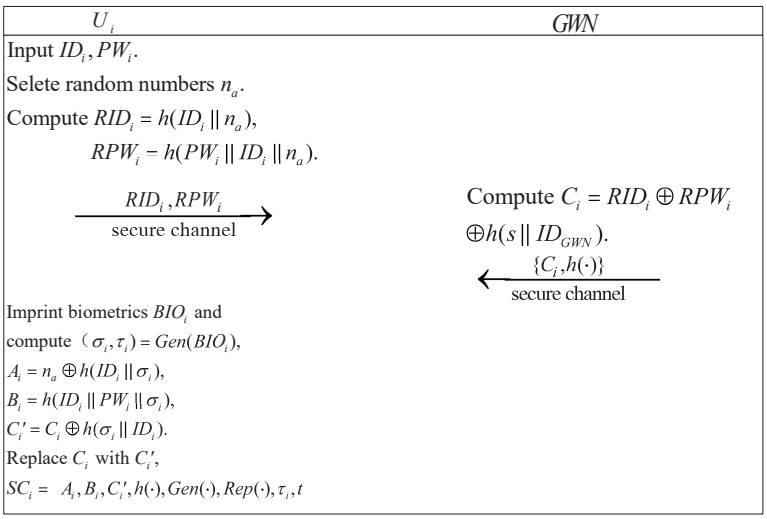

Figure 2. The user registration phase.

3.5. Login Phase

The following shows the detailed U_i login process:

- $Log1$: The SC_i is inserted by U_i into the reader of the terminal equipment, and U_i inputs his/her ID_i, PW_i and $BIO_i{}^*$. When the hamming distance between BIO_i and

BIO_i^* does not exceed the threshold value t, SC_i calculates $\sigma_i^* = Rep(BIO_i^*, \tau_i)$, $B_i^* = h(ID_i \| PW_i \| \sigma_i^*)$.

- $Log2$: SC_i checks whether $B_i^* = B_i$, if the equation does not hold, the ID_i, PW_i and BIO_i^* entered by U_i is considered to be an invalid authentication information and the login request is rejected. Otherwise, U_i is considered to be a legitimate user and proceeds with the next step, SC_i further calculates $n_a^* = A_i \oplus h(ID_i \| \sigma_i^*)$, $RID_i = h(ID_i \| n_a^*)$, $RPW_i = h(PW_i \| RID_i)$, $C_i = C_i^{'} \oplus h(\sigma_i^* \| ID_i)$, $D_i = C_i \oplus RID_i \oplus RPW_i$, then U_i selects the ID_{SN_j} of the sensor node SN_j in the IoT that he/she wants to access.
- $Log3$: SC_i then generates a random number n_i and the timestamp TS_1 of the current system, and calculates $E_i = h(D_i \| h(ID_i \| PW_i) \| TS_1)$, $M_1 = T_{n_i}(RID_i \| ID_{SN_j} \| E_i)$, $M_1^{'} = M_1 \oplus h(RID_i \| D_i \| TS_1)$, $A_{UG} = h(RID_i \| M_1 \| ID_{SN_j} \| TS_1)$, $E_i^{'} = E_i \oplus h(RID_i \| D_i \| TS_1)$, $RID_i^{'} = RID_i \oplus h(ID_{SN_j} \| TS_1)$, $ID_{SN_j}^{'} = ID_{SN_j} \oplus h(E_i^{'} \| D_i \| TS_1)$. U_i's login request message $MSG1 = \{E_i^{'}, RID_i^{'}, A_{UG}, ID_{SN_j}^{'}, TS_1, M_1^{'}\}$ is sent to GWN over a public channel.

3.6. Authentication and Key Agreement Phase

The user and sensor will authenticate with each other and share a session key with the assistance of GWN in this stage, and the specific steps are as follows:

- $AK1$: When GWN receives the user's login request message $MSG1$ at $TS_1^{'}$, the condition $| TS_1^{'} - TS_1| \leq \Delta T$ is checked, where ΔT is the maximum transmission delay. If the inequality is not met, GWN rejects the login request; otherwise, it calculates $G_i = h(x \| ID_{GWN})$, $ID_{SN_j} = ID_{SN_j}^{'} \oplus h(E_i^{'} \| G_i \| TS_1)$, $RID_i = RID_i^{'} \oplus h(ID_{SN_j} \| TS_1)$ and $M_1^* = M_1^{'} \oplus h(RID_i \| G_i \| TS_1)$. Then, GWN checks whether the equation $A_{UG} = h(RID_i \| M_1^* \| ID_{SNj} \| TS_1)$ is true; if not, the process is terminated.
- $AK2$: Otherwise, GWN generates a timestamp TS_2 for the current system, and computes $E_i = E_i^{'} \oplus h(RID_i \| G_i \| TS_1)$, $M_2 = E_{TC_{SNj}}(RID_i, M_1^*, E_i)$ and $A_{GS} = h(TC_{SN_j} \| ID_{SN_j} \| M_1^* \| TS_2)$. Then, GWN transmits the message $MSG2 = \{M_2, A_{GS}, TS_2\}$ to SN_j through a public channel.
- $AK3$: After receiving the message $MSG2$ from GWN at time $TS_2^{'}$, SN_j determines whether the condition $\left|TS_2^{'} - TS_2\right| \leq \Delta T$ is satisfied. If the validation fails, SN_j rejects GWN. Otherwise, SN_j uses the secret key TC_{SN_j} to decrypt (RID_i, M_1^*, E_i), then calculates whether this satisfies the equation $A_{GS} = h(TC_{SN_j} \| ID_{SN_j} \| M_1^* \| TS_2)$.
- $AK4$: If the validation fails, SN_j rejects $MSG2$. Otherwise, SN_j picks a random value n_j and the timestamp TS_3 of the system at the moment, and computes $F_j = T_{n_j}(RID_i \| ID_{SN_j} \| E_i)$. Then, SN_j generates and saves the secret key of the session $SK_{ij} = h(T_{n_j}(M_1^*) \| RID_i \| ID_{SN_j} \| TS_3)$, which is used for subsequent communication between SN_j with U_i. Finally, SN_j calculates $A_{US} = h(SK_{ij} \| F_j \| TS_3)$, $F_j^{'} = F_j \oplus h(RID_i \| ID_{SN_j} \| TS_3)$, and transmits the message $MSG3$ publicly to U_i.
- $AK5$: As soon as U_i receives $MSG3$, the message is checked for freshness $| TS_3^{'} - TS_3| \leq \Delta T$, where $TS_3^{'}$ is the time when the $MSG3$ is received. If the condition is met, U_i computes $F_j = F_j^{'} \oplus h(RID_i \| ID_{SN_j} \| TS_3)$, and generates the session key $SK_{ij}^* = h(T_{n_i}(F_j) \| RID_i \| ID_{SN_j} \| TS_3)$ with SN_j. Then, U_i verifies that equation $A_{US} = h(SK_{ij}^* \| F_j \| TS_3)$ is satisfied; if the authentication passes, U_i saves the session key SK_{ij}^*, and U_i and SN_j can communicate securely using SK_{ij}^*. The specific login and authentication process is shown in Figure 3.

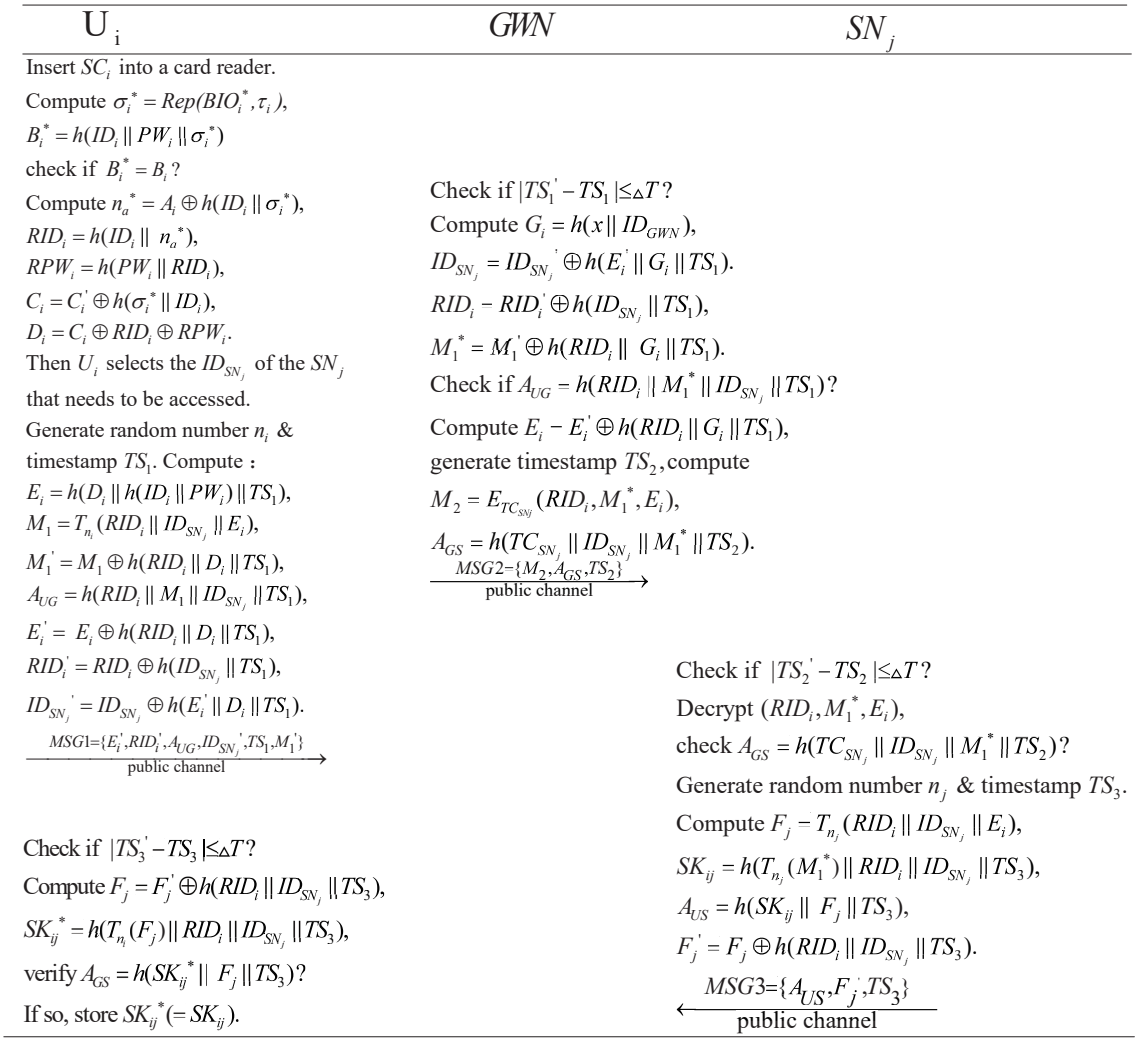

Figure 3. Login and authentication phases.

3.7. Password and Biometric Update Phase

Registered users can update their passwords and biometric keys by following these steps.

- *Step*1: After inserting SC_i into the reader, U_i enters his/her ID_i and previous password PW_i^{old} and biometric key BIO_i^{old}, SC_i then computes $\sigma_i^{old} = Rep(BIO_i^{old}, t_i)$, $n_a = A_i \oplus h(ID_i \parallel \sigma_i^{old})$,$RID_i = h(ID_i \parallel n_a)$, $RPW_i^{old} = h(PW_i^{old} \parallel RID_i)$, $h(s \parallel ID_{GWN})$=$RID_i \oplus RPW_i \oplus C_i' \oplus h(\sigma_i^{old} \parallel ID_i)$, $B_i^{old} = h(ID_i \parallel PW_i^{old} \parallel \sigma_i^{old})$ and checks the equation $B_i^{old} = B_i$. If this matches, this means that U_i is legal and can update password and biometric key (if necessary).
- *Step*2: U_i enters a new password PW_i^{new} and biometric key BIO_i^{new}, then SC_i computes $(\sigma_i^{new}, \tau_i^{new}) = Gen(BIO_i^{new})$, $RPW_i^{new} = h(PW_i^{new} \parallel RID_i)$, $A_i^{new} = n_a \oplus h(ID_i \parallel \sigma_i^{new})$, $B_i^{new} = h(ID_i \parallel PW_i^{new} \parallel \sigma_i^{new})$, $C_i^{'new} = h(s \parallel ID_{GWN}) \oplus RID_i \oplus RPW_i^{new} \oplus h(\sigma_i^{new} \parallel ID_i)$.
- *Step*3: SC_i finally updates A_i, B_i and C_i' with A_i^{new}, B_i^{new} and $C_i^{'new}$ in its memory, respectively.

3.8. Smartcard Revocation Phase

A registered user who has lost his/her smart card can perform the following steps to obtain a new smart card SC_i^{new}.

- *Step*1: U_i enters the previous ID_i, but selects a different password $PW_i^{'}$, U_i then selects a random number $n_a^{'}$ and computes $RID_i = h(ID_i \parallel n_a^{'})$, $RPW_i = h(PW_i^{'} \parallel RID_i)$ and submits the revocation request $\{RID_i, RPW_i\}$ to the GWN via a safe channel.
- *Step*2: GWN receives the message and checks the availability of RID_i in the database. If RID_i is not available, then GWN assigns a new smart card SC_i^{new} to U_i.
- *Step*3: Upon receipt of the smart card SC_i^{new}, U_i imprints biometrics BIO_i and computes $(\sigma_i, \tau_i) = Gen(BIO_i)$, $A_i = n_a^{'} \oplus h(ID_i \parallel \sigma_i)$, $B_i = h(ID_i \parallel PW_i \parallel \sigma_i)$ and $C_i^{'} = C_i \oplus h(\sigma_i \parallel ID_i)$. U_i replaces C_i in SC_i^{new} with $C_i^{'}$, and stores A_i and B_i into SC_i^{new}'s memory. Thus, $SC_i^{new} = \{A_i, B_i, C_i^{'}, h(\cdot), Gen(\cdot), Rep(\cdot), \tau_i, t\}$.

3.9. New Sensing Node Deployment Phase

By performing the following steps, GWN can deploy new IoT sensor nodes SN_j^{new} into an existing network while offline.

- *Step*1: GWN assigns a unique identity ID_{SNj}^{new} to the new sensor nodes SN_j^{new} and then uses its own master key s to calculate $TC_{SNj}^{new} = h(ID_{SNj}^{new} \parallel s)$.
- *Step*2: GWN stores the credentials TC_{SNj}^{new} and ID_{SNj}^{new} into SN_j^{new}'memory prior to its deployment. GWN also stores TC_{SNj}^{new} and ID_{SNj}^{new} in its database. Finally, GWN broadcasts information about the newly deployed sensor nodes SN_j^{new} to all the registered users, who can access it according to their own needs.

4. Security Analysis

The formal and informal security analysis of the scheme will be demonstrated in this section.

4.1. Formal Security

The security of the session key is proved by detailed and rigorous mathematical analysis of protocol using an ROR model (given in Theorem 1). We also provide primitives related to ROR models. The proposed scheme mainly involves three participants, namely sensor node SN_j, user U_i, and GWN. The following is a detailed description of the ROR model [20,26].

Participants: the instances u, v and t of U_i, GWN and SN_j are represented as $\prod_{U_i}^{u}$, $\prod_{GWN}^{v}$ and $\prod_{SD_j}^{t}$, respectively, which are known as oracles.

Accepted state: When the instance $\prod^t$ receives the last message during protocol communication, then its state changes to the accept state. $\prod^t$ concatenates all the communicated messages sequentially, and $\prod^t$ forms the current session identifier (*sid*).

Partnering: Instances $\prod^{t1}$ and $\prod^{t2}$ are partners to each other as long as they meet the following three conditions: (1) They are in a state of acceptance; (2) $\prod^{t1}$ and $\prod^{t2}$ are mutually authenticated and share *sid*. (3) They are partners of each other.

Freshness: The instances $\prod_{U_i}^{u}$ or $\prod_{SD_j}^{t}$ are considered fresh if $\mathcal{A}$ does not acquire SK_{ij} through using the following $Reveal(\prod^t)$ query.

Adversary: Assume that $\mathcal{A}$ has complete control over all communication in the system and can not only intercept messages but also tamper with them. In addition, $\mathcal{A}$ can also perform the following queries.

- $Execute(\prod^t, \prod^u, \prod^v)$: $\mathcal{A}$ can carry out an eavesdropping attack under this query because $\mathcal{A}$ can intercept all communications of three party entities during protocol execution.
- $Reveal(\prod^t)$: $\mathcal{A}$ can obtain the session keys created by the two instances by executing the query.
- $Send(\Pi^t, msg)$: $\mathcal{A}$ can send a message msg to the instance Π^t through this query, In response, $\mathcal{A}$ also can receive a message from instance Π^t.
- $CorruptSC(\prod_{U_i}^{u})$: $\mathcal{A}$ can obtain credentials stored in the smart card through this query, which is modeled as the smart card loss attack.

- $CorruptSN(\prod_{SN_j}^{t})$: It is an attack in which the secret TC_{SN_j} stored in the sensor divulges to $\mathcal{A}$. $CorruptSC$ and $CorruptSN$ are both weak-corruption models, where the internal information related to the instance and the temporary key is not corrupted.
- $test(\prod^{t})$: Toss an unbiased coin c, the result of which determines the output of the *test* query and is known only to $\mathcal{A}$, who executes the *test* query on the premise that SK_{ij} is fresh, the instance $\prod^{t}$ returns SK_{ij} at c = 1 or a random number at c = 0; otherwise, it prints $\perp$(null).

Random Oracle: All communication participants including $\mathcal{A}$ have access to the hash function $h(\cdot)$ (which is described in Section 3.1), a random oracle modeling $h(\cdot)$, say *hash*.

Theorem 1. *Let $\mathcal{A}$ represents an adversary running in polynomial time t versus our proposed protocol $\mathcal{P}$. In addition, q_{send} and q_h represent the number of Send and hash queries, respectively. $|hash|$ the range space of the hash function, m indicates the number of bits of the biological key σ_i, $|\mathcal{D}|$ is used to represent a uniformly distributed password dictionary, $Adv_{\mathcal{P}}^{CMDLP}$ represents the probability that $\mathcal{A}$ breaks our protocol' s semantic security in run time t and is estimated as*

$$Adv_{\mathcal{P}}(t) \leq \frac{{q_h}^2}{|hash|} + \frac{q_{send}}{2^{m-1} \cdot |\mathcal{D}|} + 2Adv^{CMDLP}(t)$$

Proof . The proof is similar to that proved in [18,26]. There are five games, say $Game_i$, $i = 0, 1, 2, 3, 4$ in total. $succ_i$ is used to represent the probability that $\mathcal{A}$ correctly guesses bit c in the $Game_i$. The games are defined in the following:

$Game_0$: In the execution game, $\mathcal{A}$ follows the ROR model to make a real attack on the proposed protocol $\mathcal{P}$. $\mathcal{A}$ must guess bit c before the game begins, it follows that

$$Adv_{\mathcal{P}}(t) = |2.Pr[succ_0] - 1| \tag{1}$$

$Game_1$: This game simulates $\mathcal{A}$ using *Execute* query for eavesdropping attack. $\mathcal{A}$ performs the *test* query to check whether the result is a real session key or a random number at the end of the game. Note that the SK_{ij} is calculated as $SK_{ij} = h(T_{n_j}(M_1{}^*) \parallel RID_i \parallel ID_{SN_j} \parallel TS_3)$, where $RID_i = h(ID_i \parallel n_a{}^*)$, $F_j = T_{n_j}(RID_i \parallel ID_{SN_j} \parallel E_i)$, $E_i = h(D_i \parallel h(ID_i \parallel PW_i) \parallel TS_1)$, $D_i = C_i \oplus RID_i \oplus RPW_i = h(x \parallel ID_{GWN})$. Because SK_{ij} contains the long-term secrets x, ID_{GWN}, $RID_i{}'$, $ID_{SN_j}{}'$, and short-term random secrets n_j and n_i. Without these secrets, the probability of $\mathcal{A}$'s winning $Game_1$ is not changed by eavesdropping the messages $MSG1 = \{E_i{}', RID_i{}', A_{UG}, ID_{SN_j}{}', TS_1, M_1{}'\}$, $MSG2 = \{M_2, A_{GS}, TS_2\}$ and $MSG3 = \{A_{US}, F_j{}', TS_3\}$ to compute the session key SK_{ij}. It follows that

$$Pr[succ_0] = Pr[succ_1] \tag{2}$$

$Game_2$: $Game_1$ is transformed into $Game_2$ by adding *Send* and *hash* queries, where $\mathcal{A}$ tries to trick the participants into receiving the error messages. Under this game, $\mathcal{A}$ can perform *hash* queries repeatedly to check for conflicts in the hash digest. All the communication messages $MSG1$, $MSG2$, and $MSG3$ contain the identity information of the entities, random nonces, timestamps, and long-term secrets. Therefore, there is no conflict when $\mathcal{A}$ makes the *Send* queries. The birthday paradox gives the following results:

$$|\Pr[succ_1] - \Pr[succ_2]| \leq {q_h}^2/(2|hash|) \tag{3}$$

$Game_3$: $Game_2$ is transformed into $Game_3$ by simulating $CorruptSC(\prod_{U_i}^{u})$ query. Under this game, $\mathcal{A}$ will obtain the credentials A_i, B_i and $C_i{}'$ stored by SC_i. $\mathcal{A}$ tries to utilize the dictionary attacks from these credentials to guess right ID_i and PW_i. Since $A_i = n_a \oplus h(ID_i \parallel \sigma_i)$, $B_i = h(ID_i \parallel PW_i \parallel \sigma_i)$, and $\mathcal{A}$ also needs other secret parameters

such as n_a and σ_i. Suppose the system specifies the number of times an incorrect password can be entered. Then, the results are as follows:

$$|\Pr[succ_2] - \Pr[succ_3]| \leq q_{Send}/(2^m.|\mathcal{D}|) \tag{4}$$

$Game_4$: This is the last game, where $\mathcal{A}$ can physically capture a sensor by simulating the $CorruptSN$ query, then $\mathcal{A}$ can retrieve the information $\left\{TC_{SN_j}, ID_{SN_j}\right\}$ stored in the SD_j. $\mathcal{A}$ wants to calculate the SK_{ij}, wherein $SK_{ij} = h(T_{n_j}({M_1}^*) \parallel RID_i \parallel ID_{SN_j} \parallel TS_3) = h(T_{n_i}(F_j) \parallel RID_i \parallel ID_{SN_j} \parallel TS_3)$, suppose $\mathcal{A}$ also intercepts all messages $MSGi(i = 1, 2, 3)$. It is clear that $\mathcal{A}$ needs RID_i and $T_{n_j}({M_1}^*)$ or $T_{n_i}(F_j)$, but $\mathcal{A}$ can not get random numbers n_j or n_i. $\mathcal{A}$ needs to solve CMDLP in run time t; for this case, the probability is at most

$$|\Pr[succ_4] - \Pr[succ_3]| \leq Adv^{CMDLP}(t) \tag{5}$$

All the random oracles are simulated; $\mathcal{A}$ only needs to guess bit c after executing the *Test* query to win the game. Thus, we get the following:

$$\Pr[succ_4] = 1/2 \tag{6}$$

From Equations (1), (2), and (6), we have

$$\frac{1}{2}Adv_{\mathcal{P}}(t) = \left|Pr[succ_0] - \frac{1}{2}\right| = |Pr[succ_1] - Pr[succ_4]| \tag{7}$$

Using Equations (3)–(5) gives the following:

$$\begin{aligned} \frac{1}{2}Adv_{\mathcal{P}}(t) &\leq |\Pr[\text{succ}_1] - \Pr[\text{succ}_2]| + |\Pr[\text{succ}_2] - \Pr[\text{succ}_4]| \\ &\leq |\Pr[\text{succ}_1] - \Pr[\text{succ}_2]| + |\Pr[\text{succ}_2] - \Pr[\text{succ}_3]| + |\Pr[\text{succ}_3] - \Pr[\text{succ}_4]| \\ &\leq q_h^2/(2|\,\text{hash}\,|) + q_{Send}/(2^m.|\mathcal{D}|) \; + Adv^{CMDLP}(t) \end{aligned} \tag{8}$$

□

4.2. Informal Security Analysis

4.2.1. Stolen Smart Card

Under this attack, assuming the adversary $\mathcal{A}$ has acquired a registered user U_i's smart card, power analysis is used to extract information $\{A_i, B_i, {C_i}', h(\cdot), Gen(\cdot), Rep(\cdot), \tau_i, t\}$ from the SC_i, however, it is almost impossible for $\mathcal{A}$ to figure out ID_i and PW_i from the one-way hash function without knowing random number n_a. In addition, if $\mathcal{A}$ intends to get ID_i and PW_i from $A_i, B_i, {C_i}'$, it is almost impossible to compute the ID_i and the PW_i if $\mathcal{A}$ does not know n_a.

4.2.2. Privileged-Insider Attack

Assume that $\mathcal{A}$ is a malicious internal user who has obtained the trust of GWN. In the stage of user registration, U_i sends a registration request message $\{RID_i, RPW_i\}$ to GWN in a secure manner. Even if $\mathcal{A}$ knows this information, without knowing the random secret keys n_a and ID_i, it is impossible to calculate PW_i for the collision-resistant properties of $h(\cdot)$. Hence, our scheme can effectively defend against privileged insider attacks.

4.2.3. User Impersonation Attack

$\mathcal{A}$ can intercept the login message $MSG1$ sent by the user in this attack. $\mathcal{A}$ attempts to get some useful information from $MSG1$ to deceive GWN that he/she is a legal user. To deceive GWN, $\mathcal{A}$ utilizes the existing information to generate a valid login request, but without knowing the parameters D_i and ID_{SNj} used to calculate parameters such as ${E_i}'$

and $M_1{}^{'}$. Therefore, $\mathcal{A}$ is nearly impossible to get a valid request login message, so our protocol can resist this attack.

4.2.4. GWN Impersonation Attack

Let $\mathcal{A}$ intercepts the messages $MSG2 = \{M_2, A_{GS}, TS_2\}$ to SN_j, and attempts to generate a valid authentication message $MSG2$ to convince SN_j that it is a real GWN. However, $\mathcal{A}$ cannot get the secret TC_{SN_j} shared with SN_j and $M_1{}^*$. Thus, our solution is secure enough to resist GWN impersonation attack.

4.2.5. Sensing Node Impersonation Attack

Suppose $\mathcal{A}$ intercepts the message $MSG3 = \{A_{US}, F_j{}^{'}, TS_3\}$ sent by the sensor to the user. $\mathcal{A}$ needs to generate a valid $MSG3$ to masquerade as a real sensor node. However, $\mathcal{A}$ cannot obtain parameters E_i, F_j and random secret n_j to computeA_{US}, $F_j{}^{'}$ and SK_{ij}, so a valid response message cannot be generated. Therefore, it is not computationally feasible for $\mathcal{A}$ to impersonate SN_j. Combined with the above, our scheme can resist sensor impersonation attacks.

4.2.6. User Anonymity and Untraceability

If $\mathcal{A}$ intercepts all communications messages $MSG1$, $MSG2$, and $MSG3$. However, in the case that $\mathcal{A}$ does not know the secret credentials E_i, D_i, M_1, and n_i, it is difficult to calculate the correct user identity ID_i only from the intercepted messages. Therefore, our scheme can protect the user's identity from disclosure and protect user anonymity.

4.2.7. Resilience against Sensing Node Capture Attack

Suppose the sensor SN_j deployed in the factory is physically captured by $\mathcal{A}$ and the credentials ID_{SN_j} and TC_{SN_j} stored therein are also extracted by $\mathcal{A}$, where $TC_{SN_j} = h(ID_{SN_j} \parallel s)$, because the identity ID_{SN_j} of each sensor node SN_j is unique, and $\mathcal{A}$ cannot calculate the session key between other uncaptured sensors and users from the captured SN_j's memory. Therefore, even if some sensors are captured, the session key between other normal sensors and the user will not be compromised or affected. Thus, the scheme can resist sensor capture attack.

4.2.8. Mutual Authentication

U_i, GWN, and SN_j authenticate each other in the following three ways in this paper:

1. The mutual authentication between U_i and GWN mainly relies on shared secret credentials $D_i = C_i \oplus RID_i \oplus RPW_i$=$h(x \parallel ID_{GWN})$ and $A_{UG} = h(RID_i \parallel M_1 \parallel ID_{SN_j} \parallel TS_1)$. When receiving login request message from U_i, GWN can use the stored information to calculate $G_i = h(x \parallel ID_{GWN})$ and $MSG1$ to verify A_{UG};
2. The key to mutual authentication between GWN and SN_j is the secret credential $TC_{SN_j} = h\left(ID_{SN_j} \parallel s\right)$ that they previously shared, since $M_2 = E_{TC_{SNj}}(RID_i, M_1{}^*, E_i)$, $A_{GS} = h(TC_{SN_j} \parallel ID_{SN_j} \parallel M_1{}^* \parallel TS_2)$, SN_j can decrypt M_2 using TC_{SN_j} stored during registration, then verifies A_{GS} with the received $MSG2$;
3. U_i can directly verify SN_j and establish session key by checking A_{US}. Therefore, our protocol can provide mutual authentication.

4.2.9. Replay Attack

Let us assume that $\mathcal{A}$ wants to intercept messages $MSGi (i = 1, 2, 3)$ during communication. Because of the timestamp added to these messages, and ΔT is typically very small, the replay message will be invalid because it fails to pass the timestamp threshold validation. Upon receiving the message, participants validate the attached current timestamp, which ensures that our solution is resistant to replay attacks.

4.2.10. Man-in-the-Middle Attack

Suppose $\mathcal{A}$ wants to intercept and tamper with the messages $MSGi(i = 1, 2, 3)$ to convince the participant that the information received is real. $\mathcal{A}$ must obtain parameters RID_i and ID_{SN_j} to calculate $RID_i^{'}$, A_{UG}, $ID_{SN_j}^{'}$, and $M_1^{'}$ to modify $MSG1$. Similarly, $\mathcal{A}$ cannot modify other messages $MSG2$ and $MSG3$ for the same reason. Therefore, the scheme can resist man-in-the-middle attack.

4.2.11. Ephemeral Secret Leakage (ESL) Attack

After mutual authentication as shown above (Section 4.2.8), both U_i and SN_j establish a shared session key $SK_{ij} = h(T_{n_j}(M_1{}^*) \parallel RID_i \parallel ID_{SN_j} \parallel TS_3)(= SK_{ij}{}^*)$, we now consider session key security in two cases:

1. Now suppose that $\mathcal{A}$ knows the short-term secret credentials n_i and n_j, but $\mathcal{A}$ cannot create the session key SK_{ij} because it lacks the long-term secrets RID_i, ID_{SN_j}, x, ID_{GWN} and TC_{SN_j};
2. If the long-term keys RID_i, ID_{SN_j}, x, and ID_{GWN} are accidentally leaked to $\mathcal{A}$, but only knowing these without knowing the temporary secret credentials n_i and n_j, $\mathcal{A}$ also cannot calculate the correct session key.

From what has been discussed above, the session key SK_{ij} can be calculated only when $\mathcal{A}$ obtains both the short and long-term secret credentials. Furthermore, even if for some reason SK_{ij} is compromised, previous and future session keys are different because of long-term secrets and dynamically generate random numbers, thus protecting the forward and backward security of session keys. Previous and future session keys are not affected if some session keys are compromised under some attacks. Thus, our scheme is still safe under ESL attack.

5. Comparative Analysis

We compare the proposed scheme with other related works in the aspects of computing and communication overhead in this section. In addition, we also analyzed safety performance.

5.1. Comparison of Computation Costs

Table 2 summarizes the computational overhead of our scheme and other schemes [13–15]. The data are based on the results of experiment [19,20,27]. The running time of different operations are $T_{ecm} \approx 0.063075$ s, $T_h \approx 0.00032$ s, $T_{f_e} \approx 0.063075$ s, $T_{E/D} \approx 0.0056$ s, $T_c \approx 0.0171$ s, and the schemes of Chang-Le [13], Li et al. [14], Wazid [15], and ours are approximately 259.02, 259.34, 92.515, and 150.035 milliseconds, respectively. Our scheme requires less computational overhead than scheme [13,14]. Although our scheme requires a bit more computational overhead than scheme [15], we can provide more security features. Figure 4 shows the computational overhead of each scheme on user, GWN, and sensor. The overhead of GWN in our scheme is relatively low, which means that we can process user login requests more quickly. In Chang-Le [13]'s solution, GWN has a lower computational overhead, but sensors occupy higher computational resources, our solution is more acceptable for resource-constrained sensors in the IIoT environment.

Table 2. Computation costs comparison.

Scheme	User	GWN	Sensing Node
Chang-Le. [13]	$2T_{ecm} + 7T_h$	$9T_h$	$2T_{ecm} + 5T_h$
Li et al. [14]	$9T_h + 2T_{ecm} + T_{fe}$	$9T_h + T_{ecm}$	$4T_h$
Wazid [15]	$T_{fe} + 7T_h + T_D$	$8T_h + 2T_E$	$7T_h + T_D$
Our.	$T_{fe} + 13T_h + 2T_{cm}$	$6T_h + T_E$	$2T_{cm} + 4T_h + T_D$

T_{ecm}: the time of an ECC point multiplication; T_{fe}: the time for a fuzzy extraction operation; T_h: the time for a one-way hash function; $T_{E/D}$: the time for an encryption/decryption using symmetric cryptographic technique; T_{cm}: the time of a Chebyshev chaotic map operation.

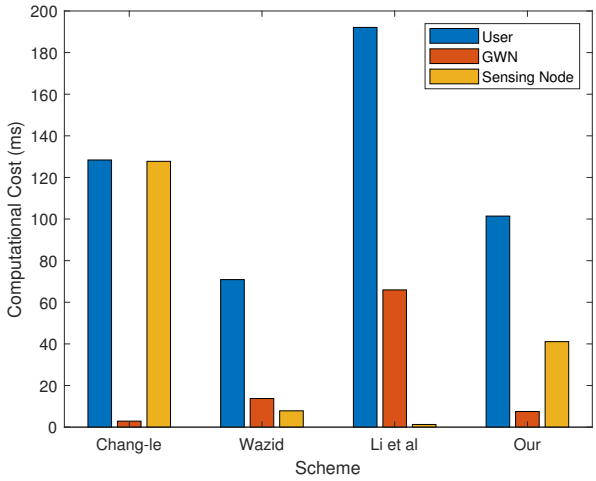

Figure 4. Computational overhead.

5.2. Comparison of Communication Costs

The communication costs of this scheme and other related schemes during the user login and authentication phases are shown in Table 3. We use 160 bits for random numbers, 160 bits for identities, 128 bits for symmetrically encrypted or decrypted block (using AES-128 algorithm [28]), 160 bits for hash digests (using SHA-1 hashing algorithm [29]), and 32 bits for timestamp. The number of bits consumed in the three messages transmitted by the scheme is $|MSG1| = 160 + 160 + 160 + 160 + 32 + 160 = 832$ bits, $|MSG2|$ = 160 + 32 + 384 = 576 bits, $|MSG3|$ = 160 + 160 + 32 = 352 bits, respectively, the total communication overhead is 1760 bits.

Table 3. Communication Cost Comparison.

Scheme	User	GWN	Sensing Node	Total (Bits)
Chang-Le. [13]	672	512	1088	2272
Li et al. [14]	1120	1120	320	2560
Wazid [15]	512	1088	384	1984
Our.	832	576	352	1760

It is reasonable that the communication overhead of some nodes in our scheme is slightly higher than that of others under the condition of satisfying more security performance and resist more attacks (see Table 4). Firstly, schemes [13,15] do not support stolen smart card attacks, resulting in a slightly lower communication overhead of users than our scheme. In order to achieve this function, it is necessary to perform multiple disguises on the user's identity, which will consume more communication overhead. Secondly, since scheme [13] does not support GWN impersonation attack, but in our scheme, TC_{SN_j} stored in GWN is an important certificate to identify the identity of GWN, and TC_{SN_j} needs to be encrypted and transmitted, so the GWN communication cost of scheme [13] is lower than ours. Finally, for sensor nodes, because scheme [14] does not support adding new sensor device functions, its communication overhead is also low. Moreover, our solution has the lowest total communication cost. Figure 5 intuitively shows the communication consumption of each scheme on user, GWN, and sensor, respectively.

Table 4. Safety performance and function comparison.

Feature	Chang-Le.	Wazid.	Li et al.	Our.
*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	×	√	×	√

*SF*1: Stolen Mobile/Smart card attack; *SF*2: Privileged-Insider attack; *SF*3: User Impersonation Attack; *SF*4: GWN Impersonation Attack; *SF*5: Sensing Node Impersonation Attack; *SF*6: User Anonymity and Untraceability; *SF*7: Sensing Node Capture Attack; *SF*8: Mutual Authentication; *SF*9: Man-in-the-Middle Attack; *SF*10: Replay Attack; *SF*11: ESL Attack; *SF*12: Support Password and Biological Key Update; *SF*13: Supports Adding Sensor Devices.

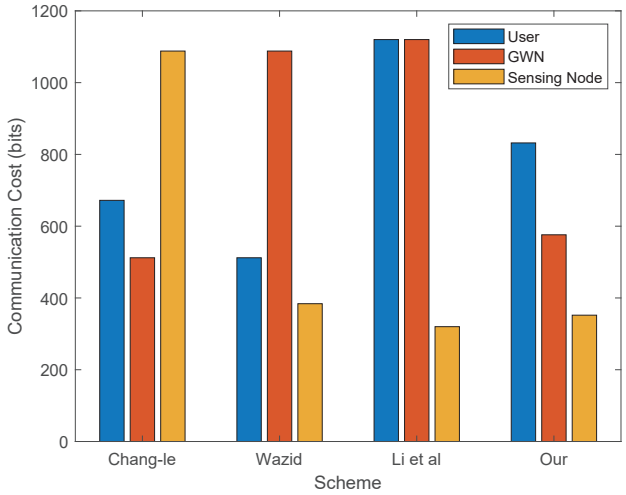

Figure 5. Communication overhead.

5.3. Comparison of Security and Functionality Features

Informal security analysis shows that our scheme can resist most known attacks (see Section 4.2), while also supporting password and biometric key changes, smart-card revocation, and adding new sensors' deployments. The security and functional features of our scheme are compared with those in [13–15] as shown in Table 4. According to the table, none of the other schemes can resist ESL attack except ours. In addition, neither scheme [13] nor [14] can satisfy the dynamic sensor node addition function. Compared with other solutions, ours can meet better security performance and function.

6. Conclusions

This paper designs a user authentication scheme based on Chebyshev polynomial encryption and fuzzy extraction operation for cloud-assisted IIoT, which can realize mutual authentication and key agreement between users and sensors. Furthermore, users can also implement the scheme to authenticate with the cloud server for access to the data in the cloud server. The protocol supports password/biometric updates, smart card reactivation, and new IoT sensor device addition without the involvement of GWN. A detailed formal (mathematical) and informal security analysis of the scheme shows that it can withstand 11 known attacks. Rigorous mathematical analysis also proves that the probability of the session key between the user and the sensor node being cracked by the adversary is almost

slim. Compared with other schemes, our scheme can meet higher security performance, and the communication cost is the lowest, and the computing cost is also within the acceptable range, which is more suitable for the IIoT environment.

Author Contributions: Conceptualization, H.H. and L.L.; methodology, H.H.; software, H.H.; validation, L.L. and H.H.; formal analysis, H.H.; investigation, L.L. and J.Z.; resources, J.Z.; writing—original draft preparation, H.H.; writing—review and editing, H.H., L.L., and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (U2001213 and 61971191) and the Beijing Natural Science Foundation under Grant Nos. L182018 and L201011 and National Key Research and Development Project (2020YFB1807204) and the Key project of the Natural Science Foundation of Jiangxi Province (20202ACBL202006).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in [19,20,27].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cao, J.; Yan, Z.; Ma, R.; Zhang, Y.; Fu, Y.; Li, H. LSAA: A Lightweight and Secure Access Authentication Scheme for Both UE and mMTC Devices in 5G Networks. *IEEE Internet Things J.* **2020**, *7*, 5329–5344. [CrossRef]
2. Zhao, J.; Liu, J.; Yang, L.; Ai, B.; Shanjin, N. Future 5G-oriented system for urban rail transit: Opportunities and challenges. *China Commun.* **2021**, *18*, 1–12. [CrossRef]
3. Guo, F.; Yu, F.R.; Zhang, H.; Li, X.; Ji, H.; Leung, V.C.M. Enabling Massive IoT Toward 6G: A Comprehensive Survey. *IEEE Internet Things J.* **2021**, *8*, 11891–11915. [CrossRef]
4. Zhao, J.; Ni, S.; Yang, L.; Zhang, Z.; Gong, Y.; You, X. Multiband Cooperation for 5G HetNets: A Promising Network Paradigm. *IEEE Veh. Technol. Mag.* **2019**, *14*, 85–93. [CrossRef]
5. Jiang, Q.; Zhang, N.; Ni, J.; Ma, J.; Ma, X.; Choo, K.K.R. Unified Biometric Privacy Preserving Three-Factor Authentication and Key Agreement for Cloud-Assisted Autonomous Vehicles. *IEEE Trans. Veh. Technol.* **2020**, *69*, 9390–9401. [CrossRef]
6. Zhao, J.; Li, Q.; Gong, Y.; Zhang, K. Computation Offloading and Resource Allocation For Cloud Assisted Mobile Edge Computing in Vehicular Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 7944–7956. [CrossRef]
7. Huang, Y.; Li, X.; Akram, Z.; Zhu, H.; Qi, Z. Generation of Millimeter-Wave Nondiffracting Airy OAM Beam Using a Single-Layer Hexagonal Lattice Reflectarray. *IEEE Antennas Wirel. Propag. Lett.* **2021**, *20*, 1093–1097. [CrossRef]
8. Zhao, J.; Yang, L.; Xia, M.; Motani, M. Unified Analysis of Coordinated Multi-Point Transmissions in mmWave Cellular Networks. *IEEE Internet Things J.* **2021**. [CrossRef]
9. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
10. Messerges, T.S.; Dabbish, E.A.; Sloan, R.H. Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Comput.* **2002**, *51*, 541–552. [CrossRef]
11. Das, A.K. A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks. *Peer-to-Peer Netw. Appl.* **2016**, *9*, 223–244. [CrossRef]
12. Jiang, Q.; Ma, J.; Lu, X.; Tian, Y. An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. *Peer-to-Peer Netw. Appl.* **2015**, *8*, 1070–1081. [CrossRef]
13. Chang, C.C.; Le, H.D. A Provably Secure, Efficient, and Flexible Authentication Scheme for Ad hoc Wireless Sensor Networks. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 357–366. [CrossRef]
14. Li, X.; Niu, J.; Kumari, S.; Wu, F.; Sangaiah, A.K.; Choo, K.K.R. A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments. *J. Netw. Comput. Appl.* **2018**, *103*, 194–204. [CrossRef]
15. Wazid, M.; Das, A.K.; Odelu, V.; Kumar, N.; Susilo, W. Secure Remote User Authenticated Key Establishment Protocol for Smart Home Environment. *IEEE Trans. Dependable Secur. Comput.* **2020**, *17*, 391–406. [CrossRef]
16. Das, A.K.; Wazid, M.; Kumar, N.; Khan, M.K.; Choo, K.K.R.; Park, Y. Design of Secure and Lightweight Authentication Protocol for Wearable Devices Environment. *IEEE J. Biomed. Health Inf.* **2018**, *22*, 1310–1322. [CrossRef]
17. Liu, W.; Liu, H.; Wan, Y.; Kong, H.; Ning, H. The yoking-proof-based authentication protocol for cloud-assisted wearable devices. *Pers. Ubiquitous Comput.* **2016**, *20*, 469–479. [CrossRef]
18. Srinivas, J.; Das, A.K.; Kumar, N.; Rodrigues, J.J.P.C. Cloud Centric Authentication for Wearable Healthcare Monitoring System. *IEEE Trans. Dependable Secur. Comput.* **2020**, *17*, 942–956. [CrossRef]
19. Chatterjee, S.; Roy, S.; Das, A.K.; Chattopadhyay, S.; Kumar, N.; Vasilakos, A.V. Secure Biometric-Based Authentication Scheme Using Chebyshev Chaotic Map for Multi-Server Environment. *IEEE Trans. Dependable Secur. Comput.* **2018**, *15*, 824–839. [CrossRef]

20. Srinivas, J.; Das, A.K.; Wazid, M.; Kumar, N. Anonymous Lightweight Chaotic Map-Based Authenticated Key Agreement Protocol for Industrial Internet of Things. *IEEE Trans. Dependable Secur. Comput.* **2020**, *17*, 1133–1146. [CrossRef]
21. Yu, J.; Wang, G.; Mu, Y.; Gao, W. An Efficient Generic Framework for Three-Factor Authentication with Provably Secure Instantiation. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 2302–2313. [CrossRef]
22. Wazid, M.; Das, A.K.; Kumar, N.; Vasilakos, A.V.; Rodrigues, J.J.P.C. Design and Analysis of Secure Lightweight Remote User Authentication and Key Agreement Scheme in Internet of Drones Deployment. *IEEE Internet Things J.* **2019**, *6*, 3572–3584. [CrossRef]
23. Chatterjee, U.; Govindan, V.; Sadhukhan, R.; Mukhopadhyay, D.; Chakraborty, R.S.; Mahata, D.; Prabhu, M.M. Building PUF Based Authentication and Key Exchange Protocol for IoT Without Explicit CRPs in Verifier Database. *IEEE Trans. Dependable Secur. Comput.* **2019**, *16*, 424–437. [CrossRef]
24. Zhao, J.; Sun, X.; Li, Q.; Ma, X. Edge Caching and Computation Management for Real-Time Internet of Vehicles: An Online and Distributed Approach. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 2183–2197. [CrossRef]
25. Sarkar, P. A simple and generic construction of authenticated encryption with associated data. *ACM Trans. Inf. Syst. Secur. (TISSEC)* **2010**, *13*, 1–16. [CrossRef]
26. Wazid, M.; Das, A.K.; Odelu, V.; Kumar, N.; Conti, M.; Jo, M. Design of Secure User Authenticated Key Management Protocol for Generic IoT Networks. *IEEE Internet Things J.* **2018**, *5*, 269–282. [CrossRef]
27. Challa, S.; Wazid, M.; Das, A.K.; Khan, M.K. Authentication Protocols for Implantable Medical Devices: Taxonomy, Analysis and Future Directions. *IEEE Consum. Electron. Mag.* **2018**, *7*, 57–65. [CrossRef]
28. Heron, S. Advanced encryption standard (AES). *Netw. Secur.* **2009**, *2009*, 8–12. [CrossRef]
29. *FIPS PUB 180-4*; Secure Hash Standard. U.S. Department of Commerce: Washington, DC, USA, 2015. [CrossRef]