Hindawi
Security and Communication Networks
Volume 2021, Article ID 3300769, 15 pages
https://doi.org/10.1155/2021/3300769

Research Article

A Lightweight Three-Factor Authentication and Key Agreement Scheme for Multigateway WSNs in IoT

Lingyan Xue,[1] Qinglong Huang,[1] Shuaiqing Zhang,[1] Haiping Huang,[1,2] and Wenming Wang[1,3]

[1]*School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210003, Jiangsu, China*
[2]*Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, Jiangsu, China*
[3]*School of Computer and Information, Anqing Normal University, Anqing 246011, Anhui, China*

Correspondence should be addressed to Haiping Huang; hhp@njupt.edu.cn

Received 10 April 2021; Revised 23 May 2021; Accepted 5 June 2021; Published 22 June 2021

Academic Editor: Jinbo Xiong

The Internet of Things (IoT) has built an information bridge between people and the objective world, wherein wireless sensor networks (WSNs) are an important driving force. For applications based on WSN, such as environment monitoring, smart healthcare, user legitimacy authentication, and data security, are always worth exploring. In recent years, many multifactor user authentication schemes for WSNs have been proposed using smart cards, passwords, as well as biometric features. Unfortunately, these schemes are revealed to various vulnerabilities (e.g., password guessing attack, impersonation attack, and replay attack) due to nonuniform security evaluation criteria. Wang et al. put forward 12 pieces of widely accepted evaluation criteria by investigating quantities of relevant literature. In this paper, we first propose a lightweight multifactor authentication protocol for multigateway WSNs using hash functions and XOR operations. Further, BAN logic and BPR model are employed to formally prove the correctness and security of the proposed scheme, and the informal analysis with Wang et al.'s criteria also indicates that it can resist well-known attacks. Finally, performance analysis of the compared schemes is given, and the evaluation results show that only the proposed scheme can satisfy all 12 evaluation criteria and keep efficient among these schemes.

1. Introduction

As the third revolution of the information technology industry, Internet of Things (IoT) has been developing for over 20 years. During this period, more and more physical objects embedded with sensors and terminal devices are constantly connected to IoT to exchange information. For an instance, in wireless sensor networks (WSNs), tens of thousands of different sensors are deployed everywhere (e.g., architectures, bridges, and intelligent terminals). These devices collect the real-time data from surrounding environment or target objects and, at fixed periods, forward the collected data directly to nearby gateway nodes for further analysis. Then, application systems access the data through the network, to further provide various personalized services. In heterogeneous WSNs, any insecure terminal nodes possibly threaten the whole network's security as the flexible access mode; potential vulnerabilities continually come forth due to the complexity of heterogeneous networks [1]. Thus, it is necessary to design an authentication protocol to ensure that only legitimate users have access to the network [2]. Generally, as far as sensor nodes are resource-constrained in some aspects such as low energy, insufficient computing capabilities, and lack of memory space, many expensive cryptographic primitives are not suitable. As a whole, the designed proposal for WSNs should be balanced well in both security and efficiency.

When it was 1981, Lamport [3] proposed the password-based authentication scheme, and in 1991, Chang and Wu [4] pioneered the smart card-based authentication scheme. Henceforth, achievements on single-factor identity authentication protocols for WSNs emerge in an endless stream. Until 2009, combining the smart card with password, Das [5] put forward a pioneering work on multifactor

authentication protocols for WSNs. However, it was revealed to many weaknesses, i.e., destitution of mutual authentication, and vulnerabilities to password guessing attack, sensor node capture attack, and denial-of-service attack (DoS) [6–8]. Later, many multifactor authentication schemes that asserted high security and efficiency were proposed yet they were prone to various attacks [9, 10]. Xue et al. [11] presented a temporal-credential-based mutual authentication and key agreement scheme for WSNs. Soon afterwards, loopholes were pointed out in their scheme, i.e., vulnerabilities to offline password guessing attack, user tracing, impersonation attack, and stolen-verifier attack, as well as the lack of user anonymity [12–14]. In recent years, biological information of human bodies, such as fingerprint and iris, has been excavated for authentication. With its unforgeability, uniqueness, and stability, biometric authentication technology is inherently convenient, reliable, and promising [15]. Yuan [16] took human's fingerprint as a third factor to achieve user authentication for WSNs, which was lightweight. Nevertheless, their scheme was pointed out that it did not withstand offline password guessing attack, privileged insider attack, and gateway impersonation attack. Then, Li et al. [17] introduced a three-factor authentication scheme for WSNs using biometric features. Subsequently, their scheme was illustrated that it could not resist to stolen smart card attack and support forward secrecy [18]. Additionally, in the practical applications of WSNs, multiple gateways are usually deployed to jointly manage multiple areas. As such, the user can access any sensor node for the real-time data in any area. Research on multigateway-based authentication protocols is also a deserving discussion. Amin et al. [19] proposed a two-factor multiple gateways' authentication protocol using hash functions. Later, Wu et al. [20] believed that their scheme did not realize mutual authentication and resist impersonation attack; then, they put forward a new scheme. And, Srinivas et al. [21] also found many flaws in [19], i.e., stolen smart card attack and sensor node spoofing attack, and then, they presented a three-factor authentication scheme using hash functions. However, their scheme was also revealed to vulnerability to sensor node capture attack and nonsupport for user anonymity. In 2019, Guo et al. [22] found that the scheme designed by Wu et al. [20] could not resist to stolen smart card attack and session key reveal attack. In order to address these drawbacks, Guo et al. [22] presented a new scheme based on biometric features. Recently, Vinoth et al. [23] proposed a secure multifactor authentication key agreement scheme for industrial IoT, which was insecure as they claimed. It actually could not deal with such attacks such as sensor node capture attack, DoS attack, and replay attack.

As all mentioned above, these schemes are exposed to various vulnerabilities constantly, which in fact are trapped into a "break-propose-break" cycle. Security properties of one scheme is determined by an evaluation standard system, thereby researchers always find new flaws under different systems. In 2018, on the basis of the previous research studies, Wang and Wang [24] summarized and put forward security criteria for two-factor authentication protocols, which are recognized by the industry at present. In these criteria, 12 pieces of independent and fundamental rules are contained that multifactor authentication protocols shall satisfy. Specific content of the criteria can be referred to [24]; we call it "12-Criteria" here for the sake of convenience.

In terms of 12-Criteria, most existing multifactor authentication protocols cannot satisfy all. This paper will put forward a new lightweight three-factor authentication and key agreement scheme for multigateway WSNs, and main contributions are summed up as below:

(1) We first reanalyse Guo et al.'s protocol [22]. And, in accordance with 12-Criteria, we further point out some vulnerabilities and drawbacks that still exist in their scheme, including no repairability, improper treatment of biological factors, offline password guessing attack, and lack of forward secrecy.

(2) In the light of the 12-Criteria, we put forward a new lightweight three-factor authentication and key agreement scheme for the multigateway environment. In our scheme, biometric features, as an important factor, are extracted and validated by fuzzy extractor [25]. And, honey_list [24] is introduced to assist the effective smart card logout.

(3) Formal and informal security analyses are given amply to prove the correctness and security of the proposed scheme, and comparisons with similar research studies show that this new scheme achieves a superior balance between security and efficiency.

The reminder of this paper is organized as follows. The relevant background is introduced in Section 2. In Section 3, discussions of some security flaws in Guo et al.'s work [22] are given. The proposed protocol and the corresponding security analysis are presented in Sections 4 and 5, respectively. The performance of the proposed protocol is evaluated in Section 6, and finally, the whole paper is concluded in Section 7.

2. Preliminaries

This section briefly introduces some necessary notations, system model, and adversary model, as well as preknowledge about formal proofs.

2.1. Notations. The related notations used in this paper are described in Table 1.

2.2. System Model. A multigateway system model is illustrated in Figure 1, wherein three roles, i.e., users, gateway nodes (GWNs), and sensor nodes, are included. Considering the distance measure, the relatively close node is referred to the home gateway node (HGWN), while the opposite is the foreign gateway node (FGWN). The communication processes are summarized as follows.

While a legitimate user attempts to communicate with the sensor node, first he needs to login successfully and send a message to inform HGWN. After the reception of the message, HGWN first checks its database with the key

Table 1: Notations.

Notation	Description
U_i, ID_i, PW_i, and BIO_i	The identity ID_i, password PW_i, and biological factor BIO_i of the user U_i
S_j and SID_j	The identity SID_j of the sensor node S_j
HGWN, ID_{hg}, and x_{hg}	The identity ID_{hg} and the private key x_{hg} of home gateway node HGWN
FGWN, ID_{fg}, and x_{fg}	The identity ID_{fg} and the private key x_{fg} of home gateway node FGWN
SA	The system administrator
SC	The smart card
ΔT	The maximum permitted transmission delay
SK_u, SK_s, SK_{fg}, and SK_{hg}	The negotiated session key
$h(\cdot)$ and $H(\cdot)$	The hash function
$Gen(\cdot)$ and $Rep(\cdot)$	The biometric feature extraction function and verification function
$\oplus$	The XOR operator
$\|$	The concatenation operator
$A \longrightarrow B$	A sends messages to B over a public channel
$A \Longrightarrow B$	A sends messages to B over a private channel

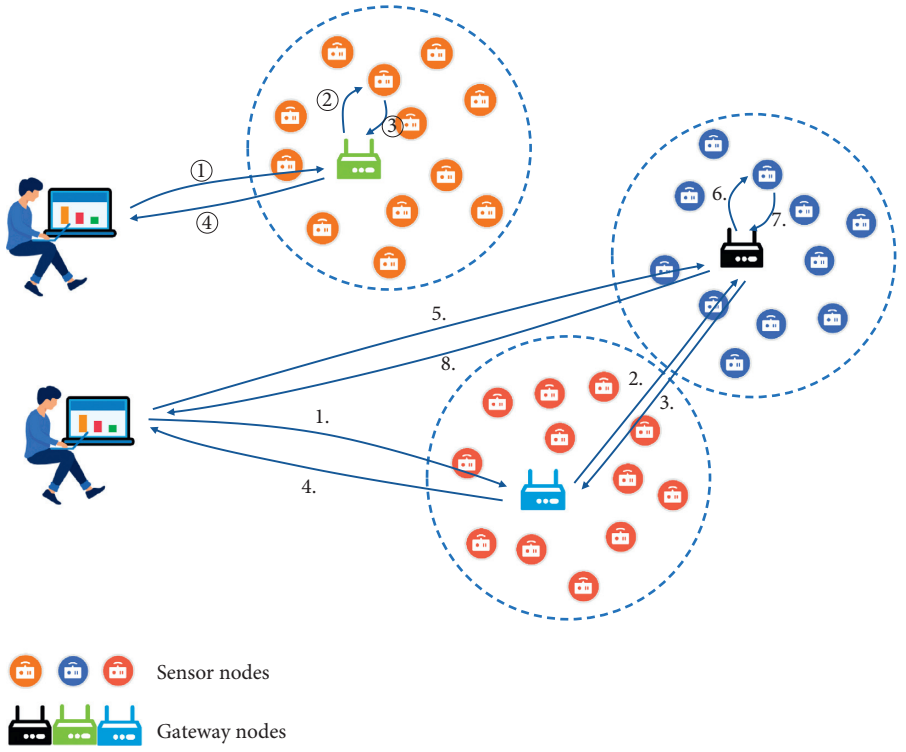

Figure 1: System model.

information of the target sensor node as an index. Here, two cases would be taken into an account. Case 1 is presented in steps ①–④, wherein if the target sensor node exists in the database, HGWN authenticates the user and sends a message to the sensor node. Then, the sensor node authenticates HGWN and returns a message. After the complete verification of the returned message, HGWN returns a message to the user. Similarly, once the message is verified correctly by the user, the three parties can derive a common session key for further communication. While Case 2 is shown in steps 1–8, that is, the target sensor node does not exist in the database, HGWN broadcasts the request message to other nodes. When FGWN receives that and finds that the wanted sensor node exists in its database, it sends a message to HGWN. Then, HGWN returns a message to the user. After a complete authentication process, the user, FGWN, and the sensor node can negotiate the very session key.

2.3. Notations and Formulas of Ban Logic. The Burrows-Abadi-Needham logic [26], BAN logic for short, plays a positive and effective role when proving that one scheme can support authentication and key agreement among communicating participants. Formally, it needs three steps including idealization of interaction messages in the protocol, initial assumptions according to specific situations, and achievements of expected goals by inference rules. We first present the basic notations of BAN logic in Table 2.

Table 2: Notations of BAN logic.

Notations	Descriptions
$P\vert \equiv X$	P believes X is true
$P \triangleleft X$	P sees X and is capable of reading and repeating it
$P\vert \sim X$	P once said X; at some time, P has sent the message containing X
$P\vert \Longrightarrow X$	P has control or jurisdiction over X
$\#(X)$	X is fresh which means it was never sent before the current execution of the protocol
$P \overset{K}{\longleftrightarrow} Q$	Both P and Q can use the shared key K to communicate with each other, and K is an intact key
$P \overset{X}{\Longleftrightarrow} Q$	X is a secret only known to P and Q and possibly to principals trusted by them
$\langle X \rangle_Y$	X combined with Y

The basic formulas of BAN logic are described as follows.

(i) (R1) Message-meaning rule: if P concludes that the secret K or Y is shared with Q and sees $\langle X \rangle_Y$or $(X)_K$, then P believes Q once said X:

$$\frac{P\vert \equiv P \overset{Y}{\Longleftrightarrow} Q, P \triangleleft \langle X \rangle_Y}{P\vert \equiv Q\vert \sim X}. \tag{1}$$

(ii) (R2) Freshness rule: if P believes X is fresh, then P believes $(X,\ Y)$ is also fresh:

$$\frac{P\vert \equiv \#(X)}{P\vert \equiv \#(X,Y)}. \tag{2}$$

(iii) (R3) Belief rule: if P believes X and Y, then P believes the combination of X and Y:

$$\frac{P\vert \equiv X, P\vert \equiv Y}{P\vert \equiv (X,Y)}. \tag{3}$$

(iv) (R4) Nonce-verification rule: if P believes that X is fresh and Q once said X, then P believes that Q believes X:

$$\frac{P\vert \equiv \#(X), P\vert \equiv Q\vert \sim X}{P\vert \equiv Q\vert \equiv X}. \tag{4}$$

(v) (R5) Jurisdiction rule: if P believes Q has jurisdiction over X and Q believes X, then P believes X:

$$\frac{P\vert \equiv Q\vert \Longrightarrow X, P\vert \equiv Q\vert \equiv X}{P\vert \equiv X}. \tag{5}$$

(vi) (R6) Seeing rule: if P once received a formula and knew the associated key, then P once saw the components of the formula:

$$\frac{P \triangleleft (X,Y)}{P \triangleleft X}, \quad \frac{P \triangleleft \langle X \rangle_Y}{P \triangleleft X}. \tag{6}$$

(vii) (R7) Session key rule: if P believes X is fresh and Q believes X, then P believes he shares the key K with Q:

$$\frac{P\vert \equiv \#(X), P\vert \equiv Q\vert \equiv X}{P\vert \equiv P \overset{K}{\leftrightarrow} Q}. \tag{7}$$

2.4. Adversary Model. Combing with the 12-Criteria, we list pieces of widely accepted valid assumptions to show the capabilities of an adversary $\mathscr{A}$, accordingly to analyse the security of the authentication and key agreement protocols.

(i) When entities in WSN communicate with each other over an insecure wireless channel, $\mathscr{A}$ can eavesdrop and intercept all messages transmitted over a public channel and is capable of tempering with and deleting the intercepted messages. In addition, $\mathscr{A}$ can participate in running the protocol as a legitimate entity.

(ii) In reality, users' devices and sensors are usually equipped with the hardware to prevent reading and tempering with data illegally [27], but to adhere to the extreme-adversary principle [28], it is reasonable to assume that when the user's device or the sensor is captured by $\mathscr{A}$, $\mathscr{A}$ has the ability to obtain the data stored in the memory of the captured sensors through side channel attack [24].

(iii) $\mathscr{A}$ is capable of enumerating the Cartesian products of the user's identity and password. Besides, in the n-factor authentication protocol, $\mathscr{A}$ can obtain $(n-1)$factors at most.

(iv) Only when evaluating the forward secrecy of the protocol, $\mathscr{A}$ can obtain the long-term private key of a gateway node or a sensor node.

2.5. Security Model. To formalize our proposed proposal later, the BPR model [29] can be introduced in this section, i.e., depictions of the random oracle model and definition of authentication and key-exchange (AKE) security.

Participants. The authentication protocol $\mathscr{P}$ involves three communication participants, i.e., the user, HGWN/FGWN, and sensor node. Each participant has many diverse instances which are called oracles. For a specific session, the three entities are instanced into Π_U^i, Π_{HGWN}^k/Π_{FGWN}^k, and Π_S^j, respectively. Here, let Π_I^* denote any instance.

Queries. $\mathscr{A}$ can only interact with honest participants through oracle queries and attempt to collect the returned messages to break the protocol. Thus, the following queries simulate $\mathscr{A}$'s abilities in practice.

(i) Execute$(\Pi_U^i, \Pi_{\mathrm{HGWN}}^k, \Pi_S^j)$: it simulates the passive attack, through which $\mathscr{A}$ can obtain all messages

among the three communicators during a normal interaction.

(ii) Send(Π_I^*, m): it represents the active attack, which allows $\mathscr{A}$ intercepts, forges the message, further sends it to Π_I^*, and obtains the corresponding response.

(iii) Reveal(Π_I^*): it models abuse of the session key. Once Π_I^* accepts the current session and generates a session key SK, it will return SK to $\mathscr{A}$; otherwise, return $\perp$.

(iv) Corrupt(Π_U^i, a): it simulates that $\mathscr{A}$ can corrupt any two of the three factors of a legal user U_i, but not at the same time. (1) If $a = 1$, $\mathscr{A}$ can obtain $\mathrm{PW}_i$and all parameters stored in SC; (2) if $a = 2$, $\mathscr{A}$ can receive $\mathrm{BIO}_i$and all parameters stored in SC; (3) if $a = 3$, $\mathscr{A}$ can get $\mathrm{PW}_i$and BIO_i.

(v) Test(Π_I^*): it represents the semantic security of the session key. Flip a coin b at random; if $b = 1$, it returns $\mathscr{A}$ the session key of Π_I^*; if $b = 0$, returns a random number equal in length to the session key to $\mathscr{A}$. If the session key of Π_I^* does not exist, it returns $\perp$. It is noted that it can only be invoked once at any time for fresh sessions.

Partners. Let *sid* denote the session identifier; *pid* is the session identifier of partners. Π_U^i and Π_S^j are partners if and only if (1) they are both authenticated successfully; (2) they both have the same *sid*; (3) *pid* of Π_U^i is Π_S^j, while *pid* of Π_S^j is Π_U^i.

Freshness. A fresh Π_I^* satisfies that (1) Π_I^* is accepted and owns its session key; (2) $\mathscr{A}$ does not query Reveal$(*)$ to Π_I^* or its partner; (3) since $\mathscr{P}$ runs, $\mathscr{A}$ queries Corrupt$(*)$to Π_I^* or its partner once at most.

Definition 1. (AKE security) Given Succ$(\mathscr{A})$ denotes an event, that is, $\mathscr{A}$ makes Test$(*)$ queries to several new accepted instances and can guess the right b' satisfying $b = b'$. Then, the advantage of $\mathscr{A}$ breaking the AKE security of $\mathscr{P}$ can be defined as $\mathrm{Adv}_{\mathscr{P}}^{\mathrm{AKE}}(\mathscr{A}) = |2\Pr[\mathrm{Succ}(\mathscr{A})] - 1| = |2\Pr[b' = b] - 1|$. For any adversary capable of breaking $\mathscr{P}$ in probability polynomial time (PPT), $\mathrm{Adv}_{\mathscr{P}}^{\mathrm{AKE}}(\mathscr{A})$is negligible; then, we say $\mathscr{P}$ achieves AKE security.

3. Cryptanalysis of Guo et al.'s Scheme

The scheme designed by Guo et al. [22] is composed of five parts, including system setup, registration, login, authentication, and password change. Here, we have to leave out the review of their scheme due to space constraints, and readers can refer to [22]. Thus, on the basis of the aforementioned assumptions, security flaws in their scheme are analysed in this section later.

No Sound Repairability. As a usual case, those discarded smart cards are not in the safe keeping of users. If unfortunate, his smart card is captured by an attacker $\mathscr{A}$. $\mathscr{A}$ possibly launches the offline password guessing attack. Therefore, it is essential to provide a method to cancel the smart card of the user in multifactor authentication protocols.

Improper Treatment of Biometric Factors. As described in this protocol, after the user enters his biometric factor BIO_i, SC calculates $O_i = H(\mathrm{BIO}_i)$ which is a key parameter to verify the true identity of the user. In practice, however, a certain error bit always occurs in the extraction of biometric features (e.g., fingerprint and iris) by reading devices, that is, biometric features extracted each time are not always identical. Therefore, O_i calculated by SC may not equal to that obtained during the user's registration phase, which may result in the failed authentication even if the user has input the right password.

Offline Password Guessing Attack. In the login phase, $\mathscr{A}$ is assumed to have the ability to obtain two of the three authentication factors. Given that $\mathscr{A}$ has accessed the user's identity ID_i and biometric factor BIO_i, then he can launch offline password guessing attack as the following process.

$\mathscr{A}$ guesses a possible password PW_i^*, calculates $O_i = H(\mathrm{BIO}_i)$, $r_i^* = B_1 \oplus h(O_i \| \mathrm{ID}_i \| \mathrm{PW}_i^*)$, and $\mathrm{MP}_i^* = h(r_i^* \| \mathrm{PW}_i^*)$, and checks whether the equation $B_2 = h(\mathrm{MP}_i^* \| \mathrm{ID}_i \| O_i \| r_i^*)$ holds. $\mathscr{A}$ can repeat these operations until the calculated B_2 equals to $h(\mathrm{MP}_i^* \| \mathrm{ID}_i \| O_i \| r_i^*)$. Finally, $\mathscr{A}$ can succeed in obtaining the user's correct PW_i.

Lack of Forward Secrecy. Given that the long-term secret key of the GWN is revealed, $\mathscr{A}$ can grab the private key of the sensor and further restore previous session keys.

(i) Case 1:

(1) $\mathscr{A}$ obtains x_{hg} of HGWN and eavesdrops the message M_1 to gain the identity SID_j of the user-pointed communication object S_j. Then, $\mathscr{A}$ computes $f_j = h(\mathrm{SID}_j \| x_{hg})$.

(2) $\mathscr{A}$ eavesdrops messages M_2 and M_3 and then calculates $Y_j = h(f_j \| T_2)$, $r_{hg} = D_3 \oplus Y_j$, $r_u = D_4 \oplus h(r_{hg} \| f_j \| T_2)$, and $r_s = D_6 \oplus h(r_{hg} \| f_j \| T_3)$. In this way, the session key can be derived by $\mathscr{A}$ as $\mathrm{SK} = h(r_s \| r_{hg} \| r_u)$.

(ii) Case 2:

(1) $\mathscr{A}$ obtains x_{fg} of FGWN and computes $f_j = h(\mathrm{SID}_j \| x_{fg})$ after eavesdropping the message M_1.

(2) $\mathscr{A}$ eavesdrops messages M_6 and M_7 and then calculates $Y_j = h(f_j \| T_2)$, $r_{fg} = D_{10} \oplus Y_j$, $r_{uu} = D_{11} \oplus h(r_{fg} \| f_j \| T_2)$, and $r_s = D_{13} \oplus h(r_{fg} \| f_j \| T_3)$. Thus, $\mathscr{A}$ can figure out $\mathrm{SK} = h(r_s \| r_{fg} \| r_u)$ with ease.

4. The Proposed Scheme

In this section, we present a lightweight three-factor authentication and key agreement scheme for multigateway

WSNs in IoT, which involves users, sensor nodes, HGWNs, and FGWNs. Our scheme includes 6 phases: system initialization, registration, login, authentication and key agreement, password update, and smart card logout.

4.1. System Initialization. SA assigns the identity ID_{hg} and private key x_{hg} to HGWN, similarly, ID_{fg} and $x_{fg}$to FGWN, and SID_j to the sensor S_j. Then, SA sets up a shared key K_{hf} for the communication between HGWN and FGWN. Beyond that, HGWN and FGWN need to select three random numbers R_h, R_f, and R_{fh}, respectively.

4.2. Registration. As shown in Figure 2, this phase involves two parts, sensor registration and user registration. Both sensor nodes and users need to register their essential information with the closest gateway, namely, HGWN.

4.2.1. Sensor Registration

Step 1: $S_j \Longrightarrow$HGWN: SID_j. $S_j$sends its identity $SID_j$to HGWN over a private channel, and HGWN stores $SID_j$to its database for checking whether or not $S_j$is registered.

Step 2: HGWN$\Rightarrow S_j$: $x_j = h(SID_j \| x_{hg}) \oplus R_h$. HGWN calculates $x_j = h(SID_j \| x_{hg}) \oplus R_h$ and sends $x_j$to $S_j$via a private channel. After the reception of x_j, S_j saves it secretly.

4.2.2. User Registration

Step 1: $U_i \Longrightarrow$HGWN: $\{ID_i, HPW_i, \beta_i\}$.

$U_i$inputs his username ID_i, the password PW_i, and his biometric information BIO_i. Next, he chooses a number $r_i \in Z_p^*$at random and then computes $(\alpha_i, \beta_i) = \text{Gen}(BIO_i)$ and $HPW_i = h(PW_i \| \alpha_i \| r_i)$.

Step 2: HGWN$\Longrightarrow U_i$: SC$\{TID_i, \beta_i, e_i, ID_{hg}\}$.

HGWN selects a pseudoidentity $TID_i$for U_i and calculates $x_i = h(TID_i \| x_{hg}) \oplus R_h$, $K_i = h(ID_i \| \beta_i)$, and $e_i = HPW_i \oplus K_i \oplus x_i$. Then, HGWN stores $\{ID_i, K_i,$ honey_list = 0$\}$ into its database and $\{TID_i, \beta_i, e_i, ID_{hg}\}$ to *SC*, where honey_listrecords the number of the user logon failures.

Step 3: U_i computes $B_1 = h(\alpha_i \| ID_i \| PW_i) \oplus r_i$ and $B_2 = h(HPW_i \| \alpha_i \| ID_i \| r_i) \bmod n_0$, where $n_0 \in [2^4, 2^8]$. Next, U_i stores $\{B_1, B_2\}$ into his *SC*.

4.3. Login

Step 1: U_i first inputs ID_i, PW_i, and BIO_i; then, *SC* computes $\alpha_i = \text{Rep}(BIO_i, \beta_i)$, $r_i = B_1 \oplus h(\alpha_i \| ID_i \| PW_i)$, and $HPW_i = h(PW_i \| \alpha_i \| r_i)$ and checks whether $B_2 = h(HPW_i \| \alpha_i \| ID_i \| r_i) \bmod n_0$ holds. If so, turn to the next step; otherwise, return a logon failure message and terminate this session.

Step 2: $U_i \longrightarrow$ HGWN: $M_1 = \{TID_i, ID_{hg}, SID_j, D_0, D_1, D_2, D_3, T_1\}$. *SC* chooses a timestamp T_1 and a random number $r_u \in Z_p^*$ and then calculates $K_i = h(ID_i \| \beta_i)$, $x_i = e_i \oplus K_i \oplus HPW_i$, $D_0 = \beta_i \oplus h(x_i \| r_u)$, $D_1 = r_u \oplus x_i$, $D_2 = ID_i \oplus h(r_u \| x_i)$, and $D_3 = h(TID_i \| ID_i \| SID_j \| r_u \| x_i \| K_i \| T_1)$.

4.4. Authentication and Key Agreement. After the reception of U_i's request to communicate with SID_j, HGWN first confirms whether the specified sensor $S_j$is located within its communication range. Specifically, if HGWN can query its local database for SID_j, then the authentication can be conducted as described in Case 1 (see Figure 3); otherwise, run as shown in Case 2 (see Figure 4).

(i) Case 1:

Step 1: after receiving M_1, HGWN records the current timestamp T_2. If $|T_2 - T_1| \le \Delta T$ is true, then $M_1$is valid; otherwise, this session would be closed up. Next, HGWN computes $x_i = h(TID_i \| x_{hg}) \oplus R_h$, $r_u = D_1 \oplus x_i$, $\beta_i = D_0 \oplus h(x_i \| r_u)$, $ID_i = D_2 \oplus h(r_u \| x_i)$, and $K_i = h(ID_i \| \beta_i)$ and verifies whether the equation $D_3 = h(TID_i \| ID_i \| SID_j \| r_u \| x_i \| K_i \| T_1)$ is true; if so, it turns into the next step; otherwise, it sets honey_list = honey_list + 1 and returns a logon failure message to U_i. Note that once honey_list ≥ 10, U_i's account would be frozen, and the session is also terminated.

Step 2: HGWN $\longrightarrow S_j$: $M_2 = \{D_4, D_5, D_6, T_2\}$. HGWN selects $r_{hg} \in Z_p^*$ randomly and then computes $x_j = h(SID_j \| x_{hg}) \oplus R_h$, $D_4 = r_{hg} \oplus h(x_j \| T_2)$, $D_5 = r_u \oplus h(r_{hg} \| x_j \| T_2)$, and $D_6 = h(SID_j \| ID_{hg} \| r_u \| r_{hg} \| x_j \| T_2)$.

Step 3: After the reception of M_2, S_j records the timestamp T_3 and checks the freshness of T_2. Next, $S_j$calculates $r_{hg} = D_4 \oplus h(x_j \| T_2)$ and $r_u = D_5 \oplus h(r_{hg} \| x_j \| T_2)$ and checks whether the equation $D_6 = h(SID_j \| ID_{hg} \| r_u \| r_{hg} \| x_j \| T_2)$; if so, it turns to the next step; otherwise, it terminates the current session.

Step 4: $S_j \longrightarrow$ HGWN: $M_3 = \{D_7, D_8, T_3\}$. $S_j$chooses a random number $r_s \in Z_p^*$ and computes $SK_s = h(r_u \| r_{hg} \| r_s \| ID_{hg})$, $D_7 = r_s \oplus h(x_j \| r_{hg} \| T_4)$, and $D_8 = h(ID_{hg} \| SID_j \| x_j \| SK_s \| r_s \| T_3)$.

Step 5: when receiving $M_3$from S_j, HGWN records the present timestamp T_4 and verifies the freshness of T_3. Next, HGWN calculates $r_s = D_7 \oplus h(x_j \| r_{hg} \| T_4)$ and $SK_{hg} = h(r_u \| r_{hg} \| r_s \| ID_{hg})$ and checks whether $D_8 = h(ID_{hg} \| SID_j \| x_j \| SK_s \| r_s \| T_3)$ holds; if so, it turns to the next step; otherwise, it aborts this session.

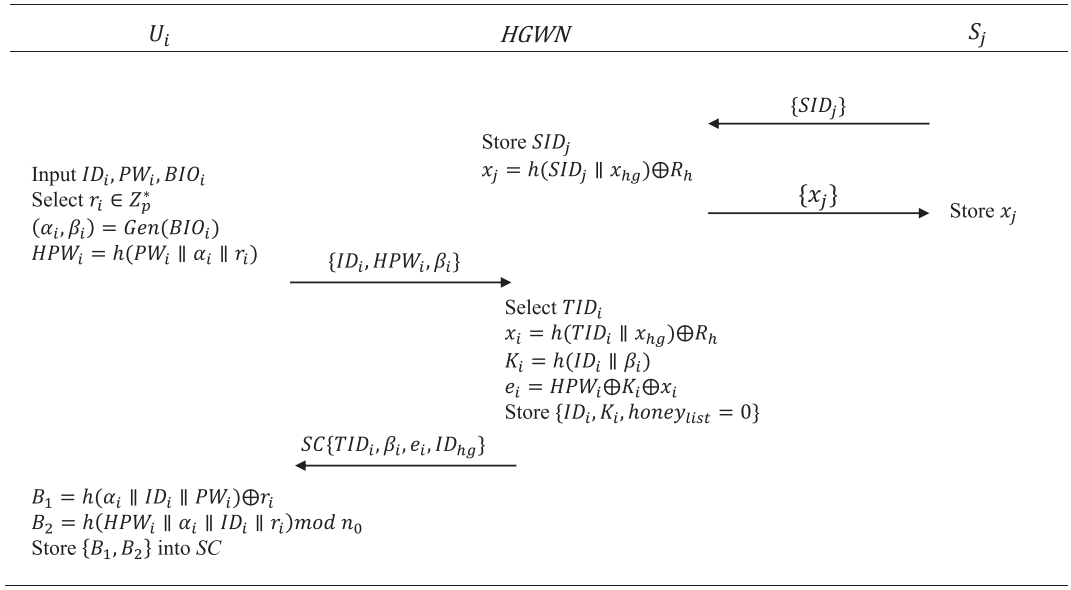

Figure 2: Registration phase.

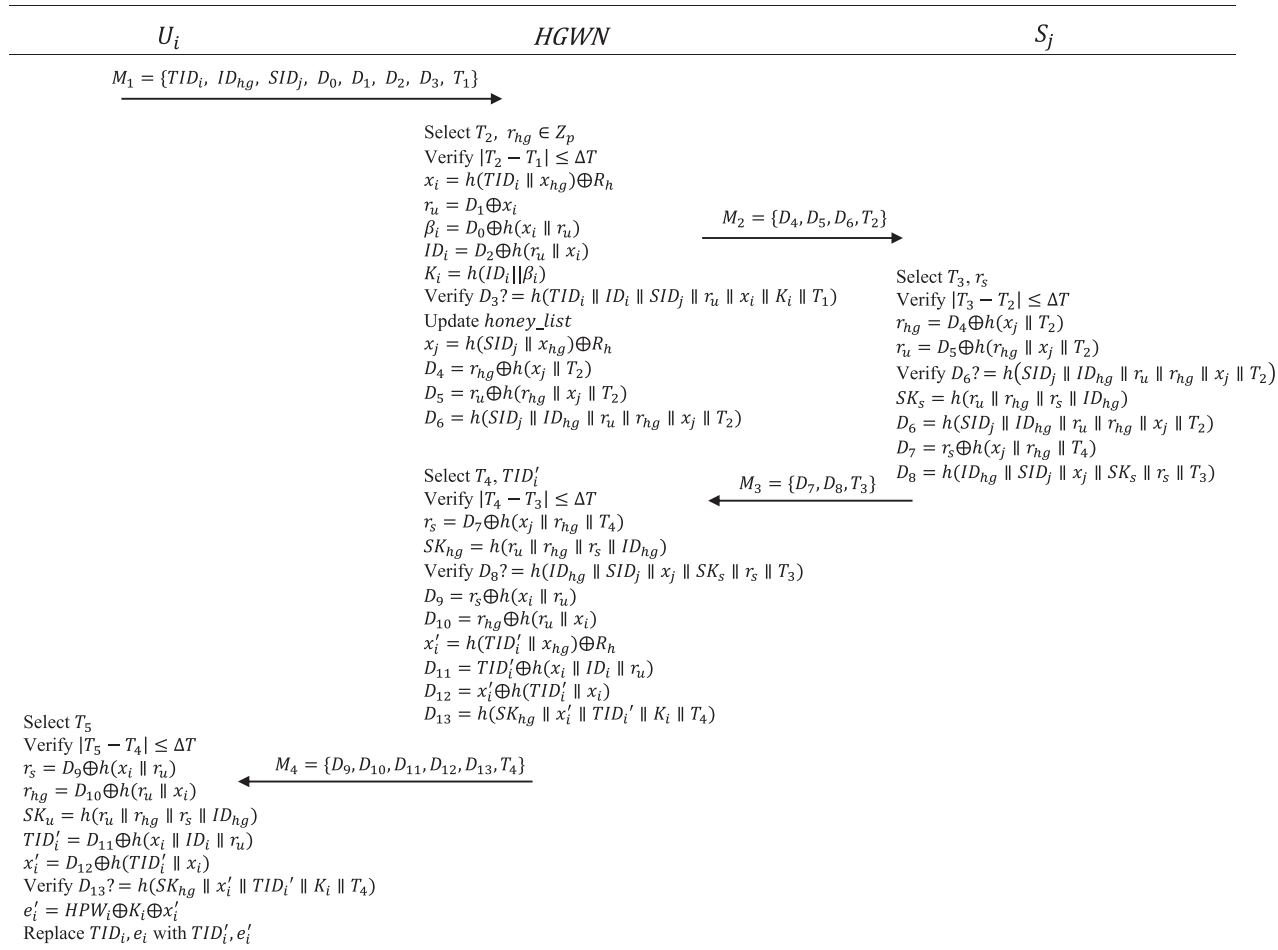

Figure 3: Case 1 of theauthentication and key agreement phase.

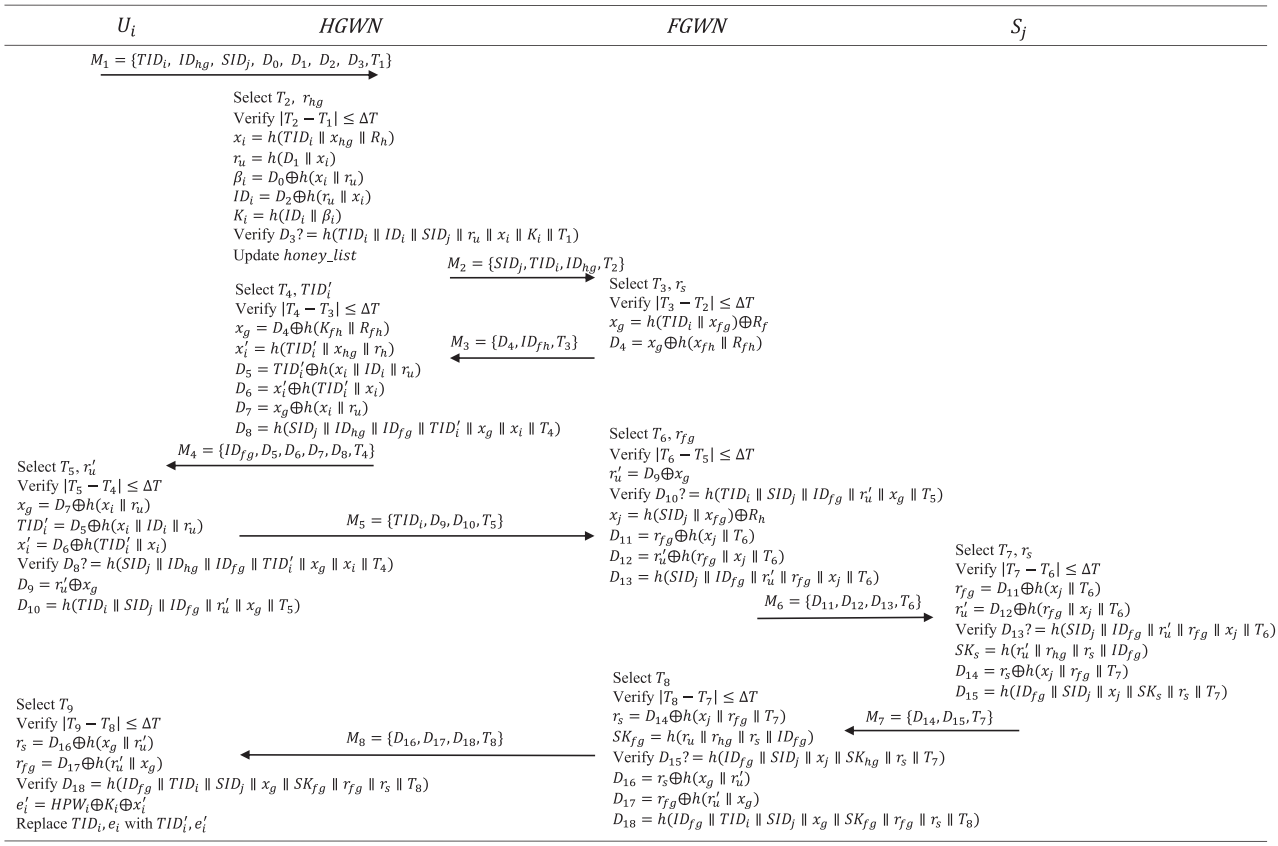

Figure 4: Case 2 of the authentication and key agreement phase.

Step 6: HGWN $\longrightarrow U_i$: $M_4 = \{D_9, D_{10}, D_{11}, D_{12}, D_{13}, T_4\}$. HGWN chooses a new pseudonym TID_i' for $U_i$and continues to compute $D_9 = r_s \oplus h(x_i \| r_u)$, $D_{10} = r_{hg} \oplus h(r_u \| x_i)$, $x_i' = h(\mathrm{TID}_i' \| x_{hg}) \oplus R_h$, $D_{11} = \mathrm{TID}_i' \oplus h(x_i \| ID_i \| r_u)$, $D_{12} = x_i' \oplus h(\mathrm{TID}_i' \| x_i)$, and $D_{13} = h(\mathrm{SK}_{hg} \| x_i' \| \mathrm{TID}_i' \| K_i \| T_4)$.

Step 7: after the reception of M_4, U_i takes down the current timestamp T_5 and checks the validity of T_4. Next, $U_i$computes $r_s = D_9 \oplus h(x_i \| r_u)$, $r_{hg} = D_{10} \oplus h(r_u \| x_i)$, $\mathrm{SK}_u = h(r_u \| r_{hg} \| r_s \| \mathrm{ID}_{hg})$, $\mathrm{TID}_i' = D_{11} \oplus h(x_i \| \mathrm{ID}_i \| r_u)$, and $x_i' = D_{12} \oplus h(\mathrm{TID}_i' \| x_i)$and verifies whether the equation $D_{13} = h(\mathrm{SK}_{hg} \| x_i' \| \mathrm{TID}_i' \| K_i \| T_4)$matches; if so, then it turns to the next step; otherwise, it discontinues the session.

Step 8: SC calculates $e_i' = \mathrm{HPW}_i \oplus K_i \oplus x_i'$ and substitutes $\{\mathrm{TID}_i', e_i'\}$for $\{\mathrm{TID}_i, e_i\}$.

(ii) Case 2:

Step 1: similarly, after the reception of M_1, HGWN takes down the current timestamp T_2. If $|T_2 - T_1| \le \Delta T$, then $M_1$is valid; otherwise, the session is discontinued. Next, HGWN computes $x_i = h(\mathrm{TID}_i \| x_{hg}) \oplus R_h$, $r_u = D_1 \oplus x_i$, $\beta_i = D_0 \oplus h(x_i \| r_u)$, $\mathrm{ID}_i = D_2 \oplus h(r_u \| x_i)$, and $K_i = h(ID_i \| \beta_i)$ and verifies $D_3 = h(\mathrm{TID}_i \| \mathrm{ID}_i \| \mathrm{SID}_j \| r_u \| x_i \| K_i \| T_1)$. If the equation holds, HGWN runs the next step; otherwise, it sets honey_list = honey_list + 1, returns a logon failure message to U_i, and aborts the session.

Step 2: HGWN broadcasts $M_2 = \{\mathrm{SID}_j, \mathrm{TID}_i, \mathrm{ID}_{hg}, T_2\}$ to other gateway nodes.

Step 3: FGWN $\longrightarrow$ HGWN: $M_3 = \{D_4, \mathrm{ID}_{fh}, T_3\}$. FGWN finds SID_jin its database, then records the present timestamp T_3, and computes $x_g = h(\mathrm{TID}_i \| x_{fg}) \oplus R_f$and $D_4 = x_g \oplus h(x_{fh} \| R_{fh})$.

Step 4: HGWN $\longrightarrow U_i$: $M_4 = \{\mathrm{ID}_{fg}, D_5, D_6, D_7, D_8, T_4\}$. When receiving M_3 from FGWN, HGWN takes down the timestamp T_4 and verifies the freshness of T_3. HGWN selects a new pseudonym $\mathrm{TID}_i'$and calculates $x_g = D_4 \oplus h(K_{fh} \| R_{fh})$, $x_i' = h(\mathrm{TID}_i' \| x_{hg} \| r_h)$, $D_5 = \mathrm{TID}_i' \oplus h(x_i \| \mathrm{ID}_i \| r_u)$, $D_6 = x_i' \oplus h(\mathrm{TID}_i' \| x_i)$, $D_7 = x_g \oplus h(x_i \| r_u)$, and $D_8 = h(\mathrm{SID}_j \| \mathrm{ID}_{hg} \| \mathrm{ID}_{fg} \| \mathrm{TID}_i' \| x_g \| x_i \| T_4)$.

Step 5: after receiving M_4, U_i records the time stamp T_5 and checks the validity of T_4. Then, $U_i$computes $x_g = D_7 \oplus h(x_i \| r_u)$, $\mathrm{TID}_i' = D_5 \oplus h(x_i \| \mathrm{ID}_i \| r_u)$, and $x_i' = D_6 \oplus h(\mathrm{TID}_i' \| x_i)$ and checks $D_8 = h(\mathrm{SID}_j \| \mathrm{ID}_{hg} \| \mathrm{ID}_{fg} \| \mathrm{TID}_i' \| x_g \| x_i \| T_4)$. If the

equation holds, $U_i$continues the next step; otherwise, it terminates the session.

Step 6: $U_i \longrightarrow$ FGWN: $M_5 = \{\text{TID}_i, D_9, D_{10}, T_5\}$. $U_i$selects a random number $r_u' \in Z_p^*$and computes $D_{10} = h(\text{TID}_i \| \text{SID}_j \| \text{ID}_{fg} \| r_u' \| x_g \| T_5)$.

Step 7: after the reception of M_5, FGWN records T_6 and verifies the freshness of T_5. Next, FGWN computes $r_u' = D_9 \oplus x_g$and further checks whether $D_{10} = h(\text{TID}_i \| \text{SID}_j \| \text{ID}_{fg} \| r_u' \| x_g \| T_5)$matches. If so, FGWN continues the next step; otherwise, it discontinues the session.

Step 8: FGWN $\longrightarrow S_j$: $M_6 = \{D_{11}, D_{12}, D_{13}, T_6\}$. FGWN selects r_{fg}at random and computes $x_j = h(\text{SID}_j \| x_{fg}) \oplus R_h$, $D_{11} = r_{fg} \oplus h(x_j \| T_6)$, $D_{12} = r_u' \oplus h(r_{fg} \| x_j \| T_6)$, and $D_{13} = h(\text{SID}_j \| \text{ID}_{fg} \| r_u' \| r_{fg} \| x_j \| T_6)$.

Step 9: after the reception of M_6, S_j takes down the timestamp T_7 and verifies the freshness of T_6. Next, $S_j$calculates $r_{fg} = D_{11} \oplus h(x_j \| T_6)$and $r_u' = D_{12} \oplus h(r_{fg} \| x_j \| T_6)$ and checks the equation $D_{13} = h(\text{SID}_j \| \text{ID}_{fg} \| r_u' \| r_{fg} \| x_j \| T_6)$. If the equation holds, $S_j$turns to the next step; otherwise, it terminates the session.

Step 10: $S_j \longrightarrow$ FGWN: $M_7 = \{D_{14}, D_{15}, T_7\}$. $S_j$selects r_s at random and computes $\text{SK}_s = h(r_u' \| r_{hg} \| r_s \| \text{ID}_{fg})$, $D_{14} = r_s \oplus h(x_j \| r_{fg} \| T_7)$, and $D_{15} = h(\text{ID}_{fg} \| \text{SID}_j \| x_j \| \text{SK}_s \| r_s \| T_7)$.

Step 11: once receiving M_7, FGWN takes down T_8 and verifies the freshness of T_7. Further, FGWN computes $r_s = D_{14} \oplus h(x_j \| r_{fg} \| T_7)$and $\text{SK}_{fg} = h(r_u' \| r_{fg} \| r_s \| \text{ID}_{fg})$and checks whether the equation $D_{15} = h(\text{ID}_{fg} \| \text{SID}_j \| x_j \| \text{SK}_{fg} \| r_s \| T_7)$is true; if so, it continues the next step; otherwise, it terminates the session.

Step 12: FGWN $\longrightarrow U_i$: $M_8 = \{D_{16}, D_{17}, D_{18}, T_8\}$. FGWN computes $D_{16} = r_s \oplus h(x_g \| r_u')$, $D_{17} = r_{fg} \oplus h(r_u' \| x_g)$, and $D_{18} = h(\text{ID}_{fg} \| \text{TID}_i \| \text{SID}_j \| x_g \| \text{SK}_{fg} \| r_{fg} \| r_s \| T_8)$.

Step 13: after receiving M_8, $U_i$thereupon records the timestamp T_9 and checks the validity of T_8. Further, U_i computes $r_s = D_{16} \oplus h(x_g \| r_u')$, $r_{fg} = D_{17} \oplus h(r_u' \| x_g)$, and $\text{SK}_u = h(r_u' \| r_{fg} \| r_s \| \text{ID}_{fg})$and checks whether the equation $D_{18} = h(\text{ID}_{fg} \| \text{TID}_i \| \text{SID}_j \| x_g \| \text{SK}_u \| r_{fg} \| r_s \| T_8)$ holds; if so, it continues the next step; otherwise, it discontinues the session.

Step 14: SC computes $e_i' = \text{HPW}_i \oplus K_i \oplus x_i'$ and replaces $\{\text{TID}_i, e_i\}$with $\{\text{TID}_i', e_i'\}$.

4.5. Password Update

Step 1: $U_i$first inputs his ID_i, PW_i, and BIO_i. SC computes $\alpha_i = \text{Rep}(\text{BIO}_i, \beta_i)$, $r_i = B_1 \oplus h(\alpha_i \| \text{ID}_i \| \text{PW}_i)$, and $\text{HPW}_i = h(\text{PW}_i \| \alpha_i \| r_i)$and checks the equation $B_2 = h(\text{HPW}_i \| \alpha_i \| \text{ID}_i \| r_i) \bmod n_0$. If the equation holds, the next step can be run; otherwise, a logon failure message would be returned and the login request also would be terminated.

Step 2: $U_i$inputs a new password PW_i', and SC computes $K_i = h(\text{ID}_i \| \beta_i)$, $\text{HPW}_i' = h(\text{PW}_i' \| \alpha_i \| r_i)$, $e_i' = \text{HPW}_i' \oplus e_i \oplus \text{HPW}_i$, $B_1' = h(\alpha_i \| \text{ID}_i \| \text{PW}_i') \oplus r_i$, and $B_2' = h(\text{HPW}_i' \| \alpha_i \| \text{ID}_i \| r_i) \bmod n_0$ and then replaces $\{B_1, B_2, e_i\}$with $\{B_1', B_2', e_i'\}$.

4.6. Smart Card Logout

Step 1: $U_i$inserts his smart card SC and inputs ID_i, PW_ias well as BIO_i. Further, SC computes $\alpha_i = \text{Rep}(\text{BIO}_i, \beta_i)$, $r_i = B_1 \oplus h(\alpha_i \| \text{ID}_i \| \text{PW}_i)$, and $\text{HPW}_i = h(\text{PW}_i \| \alpha_i \| r_i)$ and checks whether $B_2 = h(\text{HPW}_i \| \alpha_i \| \text{ID}_i \| r_i) \bmod n_0$ matches; if so, it turns to Step 2; otherwise, it returns a logon failure message and terminates this session.

Step 2: $U_i \longrightarrow$ HGWN: $M_0 = \{\text{TID}_i, \beta_i, R_0, T_1\}$. $U_i$selects the current timestamp T_1, thereupon computes $K_i = h(\text{ID}_i \| \beta_i)$, $x_i = e_i \oplus K_i \oplus \text{HPW}_i$, and $R_0 = K_i \oplus (x_i \| T_1)$.

Step 3: after the reception of M_0, HGWN records the timestamp T_2. If $|T_2 - T_1| \le \Delta T$is true, then $M_0$is fresh. Then, HGWN computes $x_i = h(\text{TID}_i \| x_{hg}) \oplus R_h$and $K_i' = R_0 \oplus (x_i \| T_1)$and continues to check whether $K_i' = K_i = h(\text{ID}_i \| \beta_i)$. If the equation holds, it runs the next step; otherwise, it aborts the session.

Step 4: HGWN deletes all local records $\{\text{ID}_i, K_i, \text{honey_list}\}$of U_i.

5. Security Analysis

This section provides a rigorous security analysis for the proposed authentication scheme. On the basis of 12-Criteria, informal analysis first discusses how the proposed scheme resists against some well-known attacks. Second, the well-popular BAN logic is utilized to validate the correctness of the proposed scheme as well as the feasibility for authentication and key negotiation. Finally, the BPR model-based formal security proof demonstrates the security of the proposed scheme well.

5.1. Informal Analysis

Resistance to Insider Attack. In multifactor authentication schemes, the user's password, as a second factor, is of vital for the server/gateway to authenticate the user. The server/gateway in its usual sense is worth

trusting, while it is facing a real possibility that insiders may disclose users' sensitive information. At the registration phase, U_i's password $PW_i$is masked by $HPW_i = h(PW_i \| \alpha_i \| r_i)$to transmit to HGWN. Though $\mathscr{A}$ has the ability to obtain HPW_i, he cannot guess the correct PW_i. That is because $r_i$is a random number, only known to U_i, and $\alpha_i$and derived information from U_i's biometric factors are also secret. Additionally, the two parameters never appear in any communication channel, and $\mathscr{A}$does not possess the ability to crack hash functions. As a consequence, the proposed scheme can resist insider attack.

Resistance to Password Guessing Attack. Assuming that $\mathscr{A}$ has generated the Cartesian products $\{(ID_i, PW_i)\}$of U_i and maliciously obtained the biometric factors $BIO_i$and *SC* through the reading device, then $\mathscr{A}$ can calculate $\alpha_i = \mathrm{Rep}(BIO_i, \beta_i)$, $r_i = B_1 \oplus h(\alpha_i \| ID_i \| PW_i)$, and $HPW_i = h(PW_i \| \alpha_i \| r_i)$and further check whether the equation $B_2 = h(HPW_i \| \alpha_i \| ID_i \| r_i) \bmod n_0$holds to find out a correct password. It is noted that there are 2^{32}[24] passwords satisfying the equation, the attempts of which are enormous, thus the offline password guessing attack bounds to fail. Furthermore, honey_listrecords the number of user logon failure when HGWN verifies the identity of U_i, which makes it extremely unlikely that $\mathscr{A}$can guess the right password through online password guessing within finite attempts. Clearly, the proposed scheme can resist diverse password guessing attacks.

Resistance to Replay Attack. It is known that $\mathscr{A}$has the ability to eavesdrop and intercept messages over the public channel. So, $\mathscr{A}$may retransmit the eavesdropped or intercepted messages in a new round of the protocol implementation, to make the other party believe that "he" is legitimate to communicate with him. In the proposed protocol, however, the timestamp is employed to demonstrate the freshness of each message, so as to filter out old messages intercepted by $\mathscr{A}$. For an instance, $\mathscr{A}$ has intercepted $M_1 = \{TID_i, ID_{hg}, SID_j, D_0, D_1, D_2, D_3, K_i, T_1\}$, where $D_3 = h(TID_i \| ID_i \| SID_j \| r_u \| x_i \| K_i \| T_1)$, and at time T_1', he attempts to resend $M_1' = \{TID_i, ID_{hg}, SID_j, D_0, D_1, D_2, D_3, K_i, T_1'\}$to HGWN for login. However, $\mathscr{A}$ can only change the timestamp in the message but not that in D_3, thus the launched replay attack bounds to fail. This instance illustrates that the proposed scheme can withstand replay attack.

User Anonymity. In terms of user anonymity, it is required that $\mathscr{A}$ cannot find out the true identities of users or trace their communication trajectories. In this scheme, each user $U_i$is assigned a pseudonym TID_i, and after a round of key negotiation, his pseudonym will be updated with a new pseudonym TID_i'. Moreover, the calculation of TID_i'depends on U_i's private key $x_i$and identity ID_i, neither of which is exposed to the open channel. Therefore, $\mathscr{A}$ cannot trace the communication trajectory of the user via the pseudonym. As analysed above, user anonymity is effective.

Forward Secrecy. According to the proposed protocol, U_i's and S_j's private keys are both calculated by a random number and the gateway node's long-term key. It helps that even if the long-term key of the gateway node is leaked for some reason, $\mathscr{A}$ cannot figure out U_i's or S_j's private key due to no idea of the random number. As the session key $SK = SK_u = SK_s = SK_{hg} = h(r_u \| r_{hg} \| r_s \| ID_{hg})$depends on r_u, r_{hg}, as well as r_s, three of which are severally masked by private keys of three parties, $\mathscr{A}$ cannot compute the right *SK* at all. Consequently, the presented scheme supports forward secrecy.

Effective Smart Card Logout. For those smart cards not used any more, improper handling may pose a huge safety hazard. On the basis of the smart card logout method described in this protocol, $U_i$must enter his right ID_i, PW_i, and BIO_isimultaneously while cancelling his *SC*, so as to prevent $\mathscr{A}$ from launching malicious cancellation after the smart card is lost. In addition, $\mathscr{A}$ cannot achieve password guessing attack and obtain three authentication factors at the same time, so there is no way for $\mathscr{A}$ to masquerade as a legitimate user to cancel the smart card. Hence, the smart card logout method presented in this protocol is effective and secure.

5.2. Formal Analysis Based on BAN Logic. In the light of BAN logic, a detailed analysis in this section will illustrate that the interacting parties (U_i, HGWN, and S_j) can achieve mutual authentication and negotiate a common session key properly and securely. The analytic procedures for two cases in the proposed scheme are described as follows.

5.2.1. Security Analysis for Case 1

(i) Goals:

$G1$: $U_i| \equiv \mathrm{HGWN} \overset{SK}{\longleftrightarrow} U_i$
$G2$: $U_i| \equiv \mathrm{HGWN}| \equiv \mathrm{HGWN} \overset{SK}{\longleftrightarrow} U_i$
$G3$: $\mathrm{HGWN}| \equiv U_i \overset{SK}{\longleftrightarrow} \mathrm{HGWN}$
$G4$: $\mathrm{HGWN}| \equiv U_i| \equiv U_i \overset{SK}{\longleftrightarrow} \mathrm{HGWN}$
$G5$: $\mathrm{HGWN}| \equiv S_j \overset{SK}{\longleftrightarrow} \mathrm{HGWN}$
$G6$: $\mathrm{HGWN}| \equiv S_j| \equiv S_j \overset{SK}{\longleftrightarrow} \mathrm{HGWN}$
$G7$: $S_j| \equiv \mathrm{HGWN} \overset{SK}{\longleftrightarrow} S_j$
$G8$: $S_j| \equiv \mathrm{HGWN}| \equiv \mathrm{HGWN} \overset{SK}{\longleftrightarrow} S_j$

(ii) Idealized forms:

M_1: $U_i \overset{K_i}{\longrightarrow} \mathrm{HGWN}$: $TID_i, ID_{hg}, SID_j, D_0, \langle r_u \rangle_{xi}$, $D_2, \langle U_i \overset{K_i}{\Longleftrightarrow} \mathrm{HWGN}, r_u \rangle_{x_i}$
M_2: $\mathrm{HGWN} \longrightarrow S_j$: $D_4, D_5, \langle r_u, r_{hg} \rangle_{x_j}$
M_3: $S_j \longrightarrow \mathrm{HGWN}$: $D_7, \langle S_j \overset{SK}{\leftrightarrow} \mathrm{HGWN}, r_s \rangle_{x_j}$
M_4: $\mathrm{HGWN} \longrightarrow U_i$: $D_9, D_{10}, D_{11}, D_{12}$, $\langle \mathrm{HGWN} \overset{SK}{\longleftrightarrow} U_i, x_i', TID_i' \rangle_{K_i}$

(iii) Assumptions:

A_1: $U_i| \equiv \#(r_u, r_{hg}, r_s)$
A_2: $\mathrm{HGWN}| \equiv \#(r_u, r_{hg}, r_s)$
A_3: $S_j| \equiv \#(r_u, r_{hg}, r_s)$

A_4: $U_i| \equiv U_i \overset{x_i}{\Longleftrightarrow} \text{HGWN}_{x_j}$
A_5: $\text{HGWN}|_{x_j} \equiv \text{HGWN} \overset{}{\Longleftrightarrow} S_j$
A_6: $S_j| \equiv S_j \overset{}{\Longleftrightarrow} \text{HGWN}_{x_i}$
A_7: $\text{HGWN}|_{K_i} \equiv \text{HGWN} \overset{}{\Longleftrightarrow} U_i$
A_8: $U_i| \equiv U_i \overset{}{\Longleftrightarrow} \text{HGWN}$
A_9: $\text{HGWN}| \equiv \text{HGWN} \overset{K_i}{\Longleftrightarrow} U_i$
A_{10}: $\text{HGWN}| \equiv U_i| \Longrightarrow r_u$
A_{11}: $S_j| \equiv \text{HGWN}| \Longrightarrow r_{hg}$
A_{12}: $\text{HGWN}| \equiv S_j| \Longrightarrow r_s$
A_{13}: $U_i| \equiv \text{HGWN}| \Longrightarrow r_{hg}$
A_{14}: $U_i| \equiv \text{HGWN}| \Longrightarrow \text{HGWN} \overset{\text{SK}}{\leftrightarrow} U_i$
A_{15}: $\text{HGWN}| \equiv S_j| \Longrightarrow S_j \overset{\text{SK}}{\leftrightarrow} \text{HGWN}$

(iv) Main proofs:

From M_1 and $R6$, we can know $S1$: $\text{HGWN} \triangleleft \langle r_u \rangle_{x_i}$.
From $S1$, A_4, and $R1$, we can get $S2$: $\text{HGWN}| \equiv U_i| \sim r_u$.
From $S2$, A_2, $R2$, and $R4$, we can get $S3$: $\text{HGWN}| \equiv U_i| \equiv r_u$.
From $S3$, A_{10}, and $R5$, we can get $S4$: $\text{HGWN}| \equiv r_u$.
From A_2, $R2$, and $\text{SK} = h(r_u \| r_{hg} \| r_s \| \text{ID}_{hg})$, we can get $S5$: $\text{HGWN}| \equiv \#(\text{SK})$.
From $S3$, $S5$, and $R7$, we can get $S6$: $\text{HGWN}| \equiv (U_i \overset{\text{SK}}{\leftrightarrow} \text{HGWN})$.
Here, we have achieved $G3$.
From $S6$, A_2, and $R4$, we can get $S7$: $\text{HGWN}| \equiv U_i| \equiv U_i \overset{\text{SK}}{\leftrightarrow} \text{HGWN}$.
Then, $G4$ has been also achieved.
From M_2 and $R6$, we can know $S8$: $S_j \triangleleft \langle r_u, r_{hg} \rangle_{x_i}$.
From $S8$, A_6, and $R1$, we can gain $S9$: $S_j| \equiv \text{HGWN}| \sim (r_u, r_{hg})$.
From $S9$, A_3, $R2$, and $R4$, we can gain $S10$: $S_j| \equiv \text{HGWN}| \equiv (r_u, r_{hg})$.
From $S10$ and $R3$, we can gain $S11$: $S_j| \equiv \text{HGWN}| \equiv r_{hg}$.
From $S11$, A_{11}, and $R5$, we can gain $S12$: $S_j| \equiv r_{hg}$.
From A_3, $R2$, and $\text{SK} = h(r_u \| r_{hg} \| r_s \| \text{ID}_{hg})$, we can gain $S13$: $S_j| \equiv \#(\text{SK})$.
From $S11$, $S13$, and $R7$, we can gain $S14$: $S_j| \equiv \text{HGWN} \overset{\text{SK}}{\leftrightarrow} S_j$. Here, G7 has been proved.
From $S14$, A_3, and $R4$, we can gain $S15$: $S_j| \equiv \text{HGWN}| \equiv \text{HGWN} \overset{\text{SK}}{\leftrightarrow} S_j$.
So, $G8$ has been also gained.
From M_3 and $R6$, we can get $S16$: $\text{HGWN} \triangleleft \langle S_j \overset{\text{SK}}{\leftrightarrow} \text{HGWN}, r_s \rangle_{x_i}$.
From $S16$, A_5, and $R1$, we can get $S17$: $\text{HGWN}| \equiv S_j| \sim (S_j \overset{\text{SK}}{\leftrightarrow} \text{HGWN}, r_s)$.
From $S17$, A_2, $R2$, and $R4$, we can get $S18$: $\text{HGWN}| \equiv S_j| \equiv (S_j \overset{\text{SK}}{\leftrightarrow} \text{HGWN}, r_s)$.
From $S18$ and $R3$, we can get $S19$: $\text{HGWN}| \equiv S_j| \equiv S_j \overset{\text{SK}}{\leftrightarrow} \text{HGWN}$.
Here, we have achieved $G6$.
From $S19$, A_{15}, and $R5$, we can get $S20$: $\text{HGWN}| \equiv S_j \overset{\text{SK}}{\leftrightarrow} \text{HGWN}$.
So, $G5$ has been also gained.
From M_4 and $R6$, we can gain $S21$: $U_i \triangleleft \langle \text{HGWN} \overset{\text{SK}}{\leftrightarrow} U_i, x_i', \text{TID}_{iK_i}' \rangle_{K_i}$.
From $S21$, A_8, and $R1$, we can obtain $S22$: $U_i| \equiv \text{HGWN}| \sim (\text{HGWN} \overset{\text{SK}}{\leftrightarrow} U_i, x_i', \text{TID}_i')$.
From $S22$, A_1, $R2$, and $R4$, we can obtain $S23$: $U_i| \equiv \text{HGWN}| \equiv (\text{HGWN} \overset{\text{SK}}{\leftrightarrow} U_i, x_i', \text{TID}_i')$.
From $S23$ and $R3$, we can obtain $S24$: $U_i| \equiv \text{HGWN}| \equiv \text{HGWN} \overset{\text{SK}}{\leftrightarrow} U_i$.
So, we have achieved $G2$.
From $S24$, A_{14}, and $R5$, we can obtain $S25$: $U_i| \equiv \text{HGWN} \overset{\text{SK}}{\leftrightarrow} U_i$.
Finally, we have gained $G1$.

5.2.2. Security Analysis for Case 2

(i) Goals:

$G1$: $U_i| \equiv \text{FGWN} \overset{\text{SK}}{\leftrightarrow} U_i$
$G2$: $U_i| \equiv \text{FGWN}| \equiv \text{FGWN} \overset{\text{SK}}{\leftrightarrow} U_i$
$G3$: $\text{FGWN}| \equiv U_i \overset{\text{SK}}{\leftrightarrow} \text{FGWN}$
$G4$: $\text{FGWN}| \equiv U_i| \equiv U_i \overset{\text{SK}}{\leftrightarrow} \text{FGWN}$
$G5$: $\text{FGWN}| \equiv S_j \overset{\text{SK}}{\leftrightarrow} \text{FGWN}$
$G6$: $\text{FGWN}| \equiv S_j| \equiv S_j \overset{\text{SK}}{\leftrightarrow} \text{FGWN}$
$G7$: $S_j| \equiv \text{FGWN} \overset{\text{SK}}{\leftrightarrow} S_j$
$G8$: $S_j| \equiv \text{FGWN}| \equiv \text{FGWN} \overset{\text{SK}}{\leftrightarrow} S_j$

(ii) Idealized forms:

M_5: $U_i \longrightarrow \text{FGWN}$: $\text{TID}_i, D_9, \langle r_u' \rangle_{x_g}$
M_6: $\text{FGWN} \longrightarrow S_j$: $D_{11}, D_{12}, \langle r_u', r_{fg} \rangle_{x_j}$
M_7: $S_j \longrightarrow \text{FGWN}$: $D_{14}, \langle \text{FGWN} \overset{\text{SK}}{\leftrightarrow} S_j, r_s \rangle_{x_j}$
M_8: $\text{FGWN} \longrightarrow U_i$: $D_{16}, D_{17}, \langle \text{FGWN} \overset{\text{SK}}{\leftrightarrow} U_i, r_{fg}, r_s \rangle_{x_g}$

(iii) Assumptions:

A_1: $U_i| \equiv \#(r_u', r_{fg}, r_s)$
A_2: $\text{FGWN}| \equiv \#(r_u', r_{fg}, r_s)$
A_3: $S_j| \equiv \#(r_u', r_{fg}, r_s)$
A_4: $U_i| \equiv U_i \overset{x_g}{\Longleftrightarrow} \text{FGWN}$
A_5: $\text{FGWN}| \equiv \text{FGWN} \overset{x_j}{\Longleftrightarrow} S_j$
A_6: $S_j| \equiv S_j \overset{x_j}{\Longleftrightarrow} \text{FGWN}$
A_7: $\text{FGWN}| \equiv \text{FGWN} \overset{x_g}{\Longleftrightarrow} U_i$
A_8: $\text{FGWN}| \equiv U_i| \Rightarrow r_u'$
A_9: $S_j| \equiv \text{FGWN}| \Rightarrow r_{fg}$
A_{10}: $\text{FGWN}| \equiv S_j| \Rightarrow r_s$
A_{11}: $U_i| \equiv \text{FGWN}| \Rightarrow r_{fg}$
A_{12}: $U_i| \equiv \text{FGWN}| \Rightarrow \text{FGWN} \overset{\text{SK}}{\leftrightarrow} U_i$
A_{13}: $\text{FGWN}| \equiv S_j| \Rightarrow S_j \overset{\text{SK}}{\leftrightarrow} \text{FGWN}$

(iv) Main proofs:

From M_5 and $R6$, we obtain $S26$: FGWN$\triangleleft\langle r_u'\rangle x_g$.
From $S26$, A_7, and $R1$, we obtain $S27$: FGWN$| \equiv U_i| \sim r_u'$.
From $S27$, A_2, $R2$, and $R4$, we obtain $S28$: FGWN$| \equiv U_i| \equiv r_u'$.
From $S28$, A_8, and $R5$, we obtain $S29$: FGWN$| \equiv r_u'$.
From A_2, $R2$, and SK $= h(r_u' \| r_{fg} \| r_s \| \mathrm{ID}_{fg})$, we obtain $S30$: FGWN$| \equiv \#(\mathrm{SK})$.
From $S28$, $S30$, and $R7$, we obtain $S31$: FGWN$| \equiv U_i \overset{\mathrm{SK}}{\leftrightarrow}$FGWN.
So, $G3$ has been achieved.
From $S31$, A_2, and $R4$, we obtain $S32$: FGWN$| \equiv U_i| \equiv U_i \overset{\mathrm{SK}}{\leftrightarrow}$FGWN.
Here, $G4$ has been also obtained.
From $M_6$and $R6$, we get$S33$: $S_j \triangleleft\langle r_u', r_{fg}\rangle_{x_j}$.
From $S33$, A_6, and $R1$, we get $S34$: $S_j| \equiv$ FGWN$| \sim (r_u', r_{fg})$.
From $S34$, A_3, $R2$, and $R4$, we get $S35$: $S_j| \equiv$ FGWN$| \equiv (r_u', r_{fg})$.
From $S35$ and $R3$, we get $S36$: $S_j| \equiv$ FGWN$| \equiv r_{fg}$.
From $S36$, A_9, and $R5$, we get $S37$: $S_j| \equiv r_{fg}$.
From A_3, $R2$, and SK $= h(r_u' \| r_{fg} \| r_s \| \mathrm{ID}_{fg})$, we get $S38$: $S_j| \equiv \#(\mathrm{SK})$.
From $S36$, $S38$, and $R7$, we get $S39$: $S_j| \equiv$ FGWN$\overset{\mathrm{SK}}{\leftrightarrow} S_j$. Here, we have proved G7.
From $S39$, A_3, and $R4$, we get $S40$: $S_j| \equiv$ FGWN$| \equiv$ FGWN$\overset{\mathrm{SK}}{\leftrightarrow} S_j$.
Here, we have achieved $G8$.
From $M_7$and $R6$, we gain$S41$: FGWN$\triangleleft\langle$FGWN$\overset{\mathrm{SK}}{\leftrightarrow} S_j, r_s\rangle_{x_j}$.
From $S41$, A_5, and $R1$, we gain $S42$: FGWN$| \equiv S_j| \sim$ (FGWN$\overset{\mathrm{SK}}{\leftrightarrow} S_j, r_s$).
From $S42$ and $R3$, we gain $S43$: FGWN$| \equiv S_j| \sim$ FGWN$\overset{\mathrm{SK}}{\leftrightarrow} S_j$.
From A_2, $R2$, and SK $= h(r_u' \| r_{fg} \| r_s \| \mathrm{ID}_{fg})$, we gain $S44$: FGWN$| \equiv \#(\mathrm{SK})$.
From $S43$, $S44$, $R2$, and $R4$, we gain $S45$: FGWN$| \equiv S_j| \equiv$ FGWN$\overset{\mathrm{SK}}{\leftrightarrow} S_j$.
Here, we have achieved $G6$.
From A_{13}, $S45$, and $R5$, we gain $S46$: FGWN$| \equiv$ FGWN$\overset{\mathrm{SK}}{\leftrightarrow} S_j$.
So, we have also achieved $G5$.
From $M_8$and $R6$, we know $S47$: $U_i \triangleleft\langle$FGWN$\overset{\mathrm{SK}}{\leftrightarrow} U_i, r_{fg}, r_s\rangle_{x_g}$.
From $S47$, A_4, and $R1$, we get $S48$: $U_i| \equiv$ FGWN$| \sim$ (FGWN$\overset{\mathrm{SK}}{\leftrightarrow} U_i, r_{fg}, r_s$).
From $S48$ and $R3$, we get $S49$: $U_i| \equiv$ FGWN$| \sim$ FGWN$\overset{\mathrm{SK}}{\leftrightarrow} U_i$.
From A_1, $R2$, and SK $= h(r_u' \| r_{fg} \| r_s \| \mathrm{ID}_{fg})$, we get $S50$: $U_i| \equiv \#(\mathrm{SK})$.
From $S49$, $S50$, $R2$, and $R4$, we get $S51$: $U_i| \equiv$ FGWN$| \equiv$ FGWN$\overset{\mathrm{SK}}{\leftrightarrow} U_i$.
So, $G2$ has been gained.
From $S51$, A_{12}, and $R5$, we get $S52$: $U_i| \equiv$ FGWN$\overset{\mathrm{SK}}{\leftrightarrow} U_i$. So, G1 has been also obtained.

Consequently, all security goals are amply demonstrated, both in Case 1 and in Case 2. In the meantime, it also confirms that the communication participants (U_i, HGWN/FGWN, and S_j), can authenticate mutually and negotiate a common key successfully.

5.3. Formal Analysis Based on BPR Model

Theorem 1. *For the protocol $\mathscr{P}$, assuming that, in a polynomial time t, $\mathscr{A}$makes up to q_s Send(Π_I^*, m)queries, q_e Excute$(\Pi_U^i, \Pi_{HGWN}^k, \Pi_S^j)$queries, and $q_h$oracle queries. Let $\mathscr{D}$ represent the password space subject to Zipf distribution, wherein $C'$and $s'$are Zipf parameters; let l denote the output length of hash functions. Now, we can get*

$$\% \ \mathrm{Adv}_{\mathscr{P}}^{\mathrm{AKE}}(\mathscr{A}) \le 2C' q_s^{s'} + \frac{q_s}{2^{l-1}} + \frac{q_h^2}{2^l} + \frac{(q_s + q_e)^2}{p-1}. \tag{8}$$

Proof. Five games $\mathscr{G}_i$ ($i = 0, 1, 2, 3,$ and 4)are considered to demonstrate Theorem 1, and simulation process of each game is analysed as below, wherein $S_i$indicates an event that $\mathscr{A}$ outputs the right random bit b in $\mathscr{G}_i$, where $i = 0, 1, 2, 3,$ and 4.

$\mathscr{G}_0$: it simulates a true attack under the random oracle model. $\mathscr{A}$ has the ability to access all oracles; so according Definition 1, we have

$$\mathrm{Adv}_{\mathscr{P}}^{\mathrm{AKE}}(\mathscr{A}) = 2\Pr[S_0] - 1. \tag{9}$$

$\mathscr{G}_1$: it maintains two lists, $L_H$and L_M, respectively, recording oracle queries and communications during the execution of $\mathscr{P}$. Besides, all other queries are run as the actual protocol. In $\mathscr{G}_1$, $\mathscr{A}$ launches the passive attack to intercept all messages M_j ($j = 1, 2, 3, 4$)through Excute$(*)$query and then guesses the output result of Test(Π_I^*)query. Due to the impossibility of figuring out SK $= h(r_u \| r_{hg} \| r_s \| \mathrm{ID}_{hg})$, the advantage of a successful attack does not increase for $\mathscr{A}$, so we can get

$$\Pr[S_1] = \Pr[S_0]. \tag{10}$$

$\mathscr{G}_2$: here, $\mathscr{A}$ can make Send(Π_I^*, m)queries and $\mathscr{H}$queries to convince the true communicator of forged messages. Only when $\mathscr{A}$ happens to find some collisions and succeeds in constructing credible messages, the simulation terminates. In $\mathscr{G}_2$, two kinds of collisions may be contained: output collisions of hash functions and collisions of random numbers selected in $\mathscr{P}$. According to Birthday Paradox [30], the probabilities of their occurrence are $(q_h^2/2^{l+1})$ and $((q_s + q_e)^2/2(p-1))$, respectively. Therefore, we obtain

$$\left|\Pr[S_2] - \Pr[S_1]\right| \leq \frac{q_h^2}{2^{l+1}} + \frac{(q_s + q_e)^2}{2(p-1)}. \quad (11)$$

$\mathcal{G}_3$: this game differs from the above games in the case that when $\mathcal{A}$ can guess the correct authentication factors D_3, D_6, D_8, and D_{13}without $\mathcal{H}$ queries, the simulation terminates. It is indistinguishable from the previous games except that some instance refuses the right authentication. Thus, we have

$$\left|\Pr[S_3] - \Pr[S_2]\right| \leq \frac{q_s}{2^l}. \quad (12)$$

$\mathcal{G}_4$: in this game, $\mathcal{A}$ has abilities to reach more information through Corrupt(Π_U^i, a) query.

(i) $\mathcal{A}$ queries Corrupt$(\Pi_U^i, 1)$, which means he has got the user's password and parameters stored in *SC*. Then, in q_s Send(Π_I^*, m) queries, $\mathcal{A}$ succeeds in guessing $\alpha_i$with the length l_α, the possibility of which is $(q_s/2^{l_\alpha})$.

(ii) $\mathcal{A}$ queries Corrupt$(\Pi_U^i, 2)$, that is, $\mathcal{A}$ has accessed the user's biometric factors and parameters stored in SC. Then, in q_sSend(Π_I^*, m)queries, $\mathcal{A}$ succeeds in guessing the victim's password, the possibility of which is $C' q_s^{s'}$.

(iii) $\mathcal{A}$ queries Corrupt$(\Pi_U^i, 3)$; similarly, $\mathcal{A}$ has the user's password and biometric factors. Then, the possibility of $\mathcal{A}$ guessing the right x_i is $(q_s/2^l)$.

$\mathcal{G}_4$ and $\mathcal{G}_3$ are indistinguishable unless the above attack is successful. So, we have

$$\left|\Pr[S_4] - \Pr[S_3]\right| \leq \max\left\{\frac{q_s}{2^{l_\alpha}}, C' q_s^{s'}, \frac{q_s}{2^l}\right\} = C' q_s^{s'}. \quad (13)$$

When $\mathcal{A}$ has no efficient input to make queries to $\mathcal{H}$, there is no advantage to distinguish the real SK from a random number with the same size through Test(Π_I^*). Therefore,

$$\Pr[S_4] = \frac{1}{2}. \quad (14)$$

From (2)–(7), we can draw conclusion (1) or (8); this is

$$\begin{aligned} \mathrm{Adv}_{\mathcal{P}}^{\mathrm{AKE}}(\mathcal{A}) &= 2\left|\Pr[S_4] - \Pr[S_0]\right| \leq 2C' q_s^{s'} \\ &+ \frac{q_s}{2^{l-1}} + \frac{q_h^2}{2^l} + \frac{(q_s + q_e)^2}{p-1}. \end{aligned} \quad (15)$$

□

6. Performance Comparison

In this section, the proposed protocol is compared with several existing multifactor authentication protocols in terms of performance, involving security features, computation overhead, and storage costs. Specific comparison results and analysis are described as follows.

6.1. Security Features. On the basis of the security 12-Criteria, Table 3 presents the comparison results of these diverse authentication protocols, i.e., Guo et al. [22], Wu et al. [20], Srinivas et al. [21], Amin [19], and our proposed protocol. Definitely, the proposed protocol can satisfy all 12 evaluation criteria whereas others can meet 8 pieces at most. In particular, the new protocol in this paper exclusively provides the repairability and forward security, as well as resistance against stolen smart card attack. The protocol presented by Guo et al. [22] has weaknesses in no repairability, improper treatment of biometric features, and offline password guessing attack; the protocol of Wu et al. [20] cannot resist insider attack, stolen smart card attack, and offline password guessing attack; the protocol proposed by Srinivas et al. [21] does not protect against insider attack and offline password guessing attack and ensure that the user will be not traced; Amin's protocol [19] does not provide resistance to insider attack and guarantee of untraceability of the user. Furthermore, none of these protocols, except the proposed one, implements forward secrecy.

It should be noted that, the 12 security evaluation criteria was proposed by Wang and Wang [24]: C1 for no password verifier-table; C2 for password-friendly; C3 for no password exposure; C4 for no smart card loss attack; C5 for resistance to known attacks; C6 for sound repairability; C7 for provision of key agreement; C8 for no clock synchronization; C9 for timely typo detection; C10 for mutual authentication; C11 for user anonymity; C12 for forward secrecy.

6.2. Computation Overhead. In this section, we compare the computation overhead among the above relevant schemes. In reality, login and authentication are much more frequent than registration, thus the performance of authentication and key-agreement protocols depends primarily on the computational costs of login and authentication phases. As depicted in Table 4, the proposed scheme is more computationally expensive than other schemes at the user side. This happens unsurprisingly because that fuzzy extractor is employed in this paper to extract and verify the biometric features, which is more applicable for high security systems. As for the gateways and resource-constrained sensor nodes, the computational costs are nearly the same. At any side, the schemes proposed by Wu et al. [20] and Amin [19] have the least computational overhead as they trade low safety features for high efficiency. In summary, despite other schemes outperforming in computational complexity, the proposed scheme can protect against all security threats faced by other schemes, which is more feasible in the real world.

6.3. Storage Costs. Comparison of storage costs among the proposed scheme and other relevant schemes is stated in this section, see Table 5and Figure 5. Primarily, it is recommended that 32 bits for the (pseudo-) identity, 160 bits for the hash output, 128 bits for the fuzzy extractor public data, and 128 bits for a random number, as well as 32 bits for a timestamp are agreed, and these parameters are denoted separately as L_{ID}, L_h, L_{fe}, L_r, and L_T. As shown in Figure 5, storage overhead on the user and sensor nodes sides is nearly the same, but that on the gateway nodes is higher as in the proposed scheme; smart card logout is achieved with the assistance of honey_list saving in

Table 3: Comparison of security features.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Guo et al. [22]	✓	✓	✓	✓	×	×	✓	✓	×	✓	✓	×
Wu et al. [20]	✓	✓	×	×	×	×	✓	×	×	✓	✓	×
Srinivas et al. [21]	✓	✓	×	×	✓	×	✓	×	×	✓	×	×
Amin [19]	✓	✓	×	×	×	×	✓	×	×	✓	×	×
Ours	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

"✓" means the protocol satisfies this property; "×," the opposite.

Table 4: Comparison of computation overhead.

		Guo et al. [22]	Wu et al. [20]	Srinivas et al. [21]	Amin [19]	Ours
U_i	Case 1	$13T_h$	$9T_h$	$10T_h$	$7T_h$	$13T_h + T_{fe}$
	Case 2	$18T_h$	$11T_h$	$14T_h$	$8T_h$	$15T_h + T_{fe}$
HGWN	Case 1	$17T_h$	$11T_h$	$14T_h$	$8T_h$	$18T_h$
	Case 2	$10T_h$	$7T_h$	$6T_h$	$1T_h$	$11T_h$
FGWN	Case 1	0	0	0	0	0
	Case 2	$14T_h$	$7T_h$	$17T_h$	$7T_h$	$12T_h$
S_j	Case 1	$6T_h$	$4T_h$	$7T_h$	$5T_h$	$6T_h$
	Case 2	$6T_h$	$4T_h$	$6T_h$	$5T_h$	$6T_h$
Total	Case 1	$36T_h$	$24T_h$	$31T_h$	$20T_h$	$37T_h + T_{fe}$
	Case 2	$48T_h$	$29T_h$	$43T_h$	$21T_h$	$44T_h + T_{fe}$

Table 5: Comparison of storage costs.

	Guo et al. [22]	Wu et al. [20]	Srinivas et al. [21]	Amin [19]	Ours
SC	$L_{ID} + 3L_h + L_r$	$L_{ID} + 3L_h + L_r$	$L_{ID} + 4L_h$	$2L_{ID} + 3L_h + L_r$	$2L_{ID} + 3L_h + L_{fe}$
HGWN/FGWN	$3L_{ID} + 2L_r$	$3L_{ID} + 2L_r$	$4L_{ID} + L_r + L_T$	$4L_{ID} + L_r + L_h$	$3L_{ID} + 4L_r + L_h$
S_j	$2L_{ID} + L_h$	$2L_{ID} + L_h$	$L_{ID} + L_h + L_T$	$L_{ID} + L_h$	$L_{ID} + L_h$

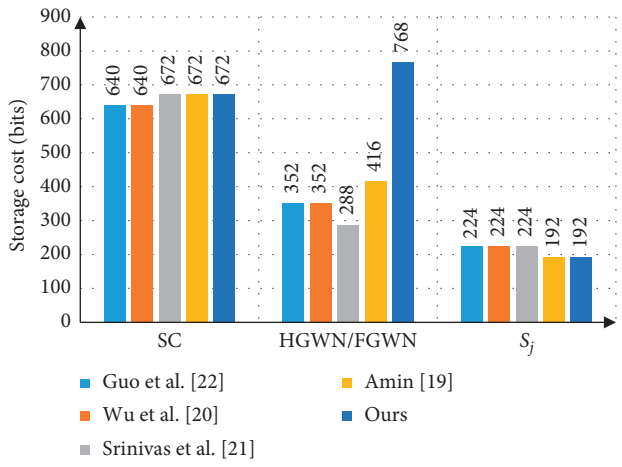

Figure 5: Comparison of storage costs.

gateway nodes' memories. However, in terms of storage capacity, gateway nodes are much better than smart cards and sensor nodes, thus the overhead is acceptable.

7. Conclusion

WSNs are becoming increasingly vital in IoT applications. Inevitably, multifactor and multigateway authentication protocols have become a focus. In this paper, through analysing weaknesses in the existing schemes, we introduced the widely accepted criteria for evaluating security protocols. In line with the criteria, we revisited Guo et al.'s scheme and found some security flaws, i.e., no repairability, improper treatment of biometric factors, offline password guessing, and no forward secrecy. Then, we proposed a new three-factor authentication protocol for multiple gateways using fuzzy extractor and honey_list technique. Following that, we proved the correctness and security of the proposed scheme by BAN logic and BPR model. As a whole, our proposed scheme outperformed other relevant schemes for keeping efficient in performance, meanwhile satisfying the security criteria.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

L. Xue and Q. Huang contributed equally to this work.

Acknowledgments

This work was supported in part by the National Key Research and Development Program (2019YFB2101704 and 2018YFB0803403), National Natural Science Foundation of China (61872194 and 62072252), and Key Project on Anhui Provincial Natural Science Study by Colleges and Universities (KJ2019A0579, KJ2020A0513 and KJ2020A0497).

References

[1] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in DWSNs," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.

[2] R. Hajian, S. ZakeriKia, S. H. Erfani, and M. Mirabi, "SHAPARAK: scalable healthcare authentication protocol with attack-resilience and anonymous key-agreement," *Computer Networks*, vol. 183, Article ID 107567, 2020.

[3] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[4] C.-C. Chang and T.-C. Wu, "Remote password authentication with smart cards," *IEE Proceedings E Computers and Digital Techniques*, vol. 138, no. 3, pp. 165–168, 1991.

[5] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.

[6] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'," *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.

[7] T.-H. Chen and W.-K. Shih, "A robust mutual authentication protocol for wireless sensor networks," *ETRI Journal*, vol. 32, no. 5, pp. 704–712, 2010.

[8] D. He, Y. Gao, S. Chan, C. Chen, and J. Bu, "An enhanced two-factor user authentication scheme in wireless sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 10, pp. 361–371, 2010.

[9] S. G. Yoo, K. Y. Park, and J. Kim, "A security-performance-balanced user authentication scheme for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 8, no. 3, Article ID 382810, 2012.

[10] P. Kumar and H.-J. Lee, "Cryptanalysis on two user authentication protocols using smart card for wireless sensor networks," in *Proceedings of the 2011 Wireless Advanced*, pp. 241–245, London, UK, June 2011.

[11] K. Xue, C. Ma, P. Hong, and R. Ding, "A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 316–323, 2013.

[12] C.-T. Li, C.-Y. Weng, and C.-C. Lee, "An advanced temporal credential-based security scheme with mutual authentication and key agreement for wireless sensor networks," *Sensors*, vol. 13, no. 8, 2013.

[13] Q. Jiang, J. Ma, X. Lu, and Y. Tian, "An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks," *Peer-to-Peer Networking and Applications*, vol. 8, no. 6, pp. 1070–1081, 2015.

[14] D. He, N. Kumar, and N. Chilamkurti, "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Information Sciences*, vol. 321, pp. 263–277, 2015.

[15] C.-T. Li and M.-S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 1–5, 2010.

[16] J.-J. Yuan, "An enhanced two-factor user authentication in wireless sensor networks," *Telecommunication Systems*, vol. 55, 2013.

[17] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, and K.-K. R. Choo, "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments," *Journal of Network and Computer Applications*, vol. 103, pp. 194–204, 2018.

[18] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, "New enhanced authentication protocol for internet of things," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 1–9, 2021.

[19] R. Amin, "A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Networks*, vol. 36, 2015.

[20] F. Wu, L. Xu, S. Kumari et al., "An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment," *Journal of Network and Computer Applications*, vol. 89, 2016.

[21] J. Srinivas, S. Mukhopadhyay, and D. Mishra, "Secure and efficient user authentication scheme for multi-gateway wireless sensor networks," *Ad Hoc Networks*, vol. 54, pp. 147–169, 2017.

[22] H. Guo, Y. Gao, T. Xu, X. Zhang, and J. Ye, "A secure and efficient three-factor multi-gateway authentication protocol for wireless sensor networks," *Ad Hoc Networks*, vol. 95, Article ID 101965, 2019.

[23] R. Vinoth, L. J. Deborah, P. Vijayakumar, and N. Kumar, "Secure multi-factor authenticated key agreement scheme for industrial IoT," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3801–3811, 2020.

[24] D. Wang and P. Wang, "Two birds with one stone: two-factor authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 708–722, 2018.

[25] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: how to generate strong keys from biometrics and other noisy data," in *Advances in Cryptology—EUROCRYPT 2004*, C. Cachin and J. L. Camenisch, Eds., Springer, Berlin, Germany, pp. 523–540, 2004.

[26] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," in *Proceedings of the Twelfth ACM Symposium on Operating Systems Principles*, Litchfield Park, AZ, USA, November 1989.

[27] M. Tunstall, K. E. Mayes, and K. Markantonakis, "Smart card security. secure smart embedded devices, platforms and applications," 2014.

[28] F. Hao, "On robust key agreement based on public key authentication," *Security & Communication Networks*, vol. 7, no. 1, pp. 77–87, 2014.

[29] E. Bresson, O. Chevassut, and D. Pointcheval, "Security proofs for an efficient password-based key exchange," in *Proceedings of the 10th ACM Conference on Computer and Communication Security: CCS '03*, pp. 241–250, Washington, DC, USA, October 2003.

[30] V. Boyko, P. MacKenzie, and S. Patel, "Provably secure password-authenticated key exchange using diffie-hellman," in *Advances in Cryptology—EUROCRYPT 2000*, B. Preneel, Ed., Springer, Berlin, Germany, pp. 156–171, 2000.