IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 June 2023, accepted 24 July 2023, date of publication 1 August 2023, date of current version 3 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3300440

RESEARCH ARTICLE

A Lightweight User Authentication Scheme for Multi-Gateway Based Wireless Sensor Networks Using Rabin Cryptosystem

XINGWEN ZHAO AND DEXIN LI

State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China
School of Cyber Engineering, Xidian University, Xi'an 710000, China

Corresponding author: Dexin Li (lidexin@stu.xidian.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61772404.

ABSTRACT The existing authentication schemes in wireless sensor networks (WSNs) are mostly used in single-gateway mode. With the wide deployment of WSNs, the drawbacks of single-gateway mode are gradually becoming more noticeable. In traditional single-gateway WSNs, high-speed data streams are prone to conflict during data aggregation, which may reduce the performance of the network. Most of the existing multi-gateway schemes are based on lightweight operations such as hash functions, exclusive OR (XOR), and symmetric encryption algorithms, which cannot achieve forward secrecy. In this paper, we propose a lightweight multi-gateway authentication scheme based on the Rabin cryptosystem. Since Rabin only requires a module-square operation for encryption, its computational overhead is relatively low. Therefore, the encryption operation is usually used on the sensor side with resource constraints to save resources. In addition, Scyther is used to prove the security of the proposed scheme. The analysis shows that the proposed scheme can achieve higher security with lower computational overhead.

INDEX TERMS Wireless sensor networks, authentication, multi-gateway, rabin cryptosystem.

I. INTRODUCTION

Wireless Sensor Networks (WSNs), as one of the core technologies of the perception layer of the Internet of Things (IoT), consist mostly of terminal devices with limited resources. Due to the simple structure and low computing power of WSNs, ensuring their access security has been an important research direction for IoT security. Insecure terminal devices will pose a threat to the security of the whole network. EI-Haij et al. [1] pointed out that a single compromised node can be turned into a malicious one that can bring down the whole system or cause disasters. Atzori et al. [2] pointed out that authentication is a key issue in IoT security, and it is essential for network security to verify the identities of the entities accessing the IoT. In traditional single-gateway authentication schemes, high-speed data streams are prone to conflict during data aggregation. When edge sensors are too far away from gateway nodes, it will lead to increased communication costs and reduced network performance. Therefore, it is of great significance to study the multi-gateway authentication protocol for WSNs.

The associate editor coordinating the review of this manuscript and approving it for publication was Marco Martalo.

A. LITERATURE REVIEW

This paper mainly classifies the current mainstream authentication protocols into two kinds: single-gateway protocols and multi-gateway protocols.

In 1981, Lamport [3] first proposed a single-factor authentication protocol based on password. In 2004, Watro et al. [4] proposed a user authentication protocol using RSA and Diffie-Hellman algorithms that placed the computationally expensive operations on parties external to WSNs. However, Das [5] pointed out that Watro et al.'s protocol suffered from impersonation attack. In 2006, Wong et al. [6] proposed a dynamic strong password solution to access control in WSNs, which only required simple hash function and XOR operations. However, in 2007, Tseng et al. [7] showed that

Wong et al.'s scheme was vulnerable to replay attack and forgery attack, then proposed a lightweight dynamic user authentication scheme for WSNs, which was in possession of many advantages, including resistance to replay attack and forgery attack. Since many schemes relied only on password security, they were vulnerable to off-line password guessing attack. To avoid these problems, two-factor authentication protocols based on password and smart card have been proposed one after another. In 2009, Das [5] established a novel two-factor authentication scheme, where the user is in possession of a password and a smart card. In the same year, Nyang and Lee [8] pointed out that Das's protocol [5] was vulnerable to off-line password guessing attack, and presented a countermeasure to overcome this drawback. In 2010, Chen and Shih [9] showed Das's scheme [5] failed to achieve mutual authentication, and they proposed a robust mutual authentication protocol. In 2010, Cheikhrouhou et al. [10] proposed a lightweight authentication scheme based on the symmetric algorithm AES, in which mutual authentication and key establishment mechanisms were used to ensure the confidentiality and data integrity of the protocol. Public key algorithm-based schemes can accomplish more security attributes than symmetric algorithm-based schemes, but they also use up more system resources. In 2011, Yeh et al. [11] found that Chen and Shih's scheme [9] failed to provide a secure method for updating user passwords and was vulnerable to the insider attack, where the privileged insider can obtain the user's password. To address these existing issues, they first applied the elliptic curve cryptography (ECC) algorithm and smart card to construct WSNs authentication protocol. Han [12] pointed out that the above protocol cannot provide perfect forward secrecy and cannot achieve mutual authentication or key agreement between user and sensor nodes. In 2013, Shi and Gong [13] established an authentication protocol to achieve perfect forward secrecy, mutual authentication, and key agreement between user and sensor nodes. Choi et al. [14] pointed out that Shi and Gong's protocol [13] was vulnerable to session key attack, stolen smart card attack, and node energy consumption attack, and improved on these drawbacks. Authentication protocols based on ECC are one of the methods used in authentication protocols in order to improve the security and privacy of RFID systems. In 2016, Dinarvand and Barati [15] examined and compared protocols that utilized this method to establish security. In 2019, a RFID authentication protocol wad presented using ECC for mutual authentication overcome weakness of the existing schemes by Dinarvand and Barati [16]. In 2020, Srinivas et al. [17] came up with a novel user authentication scheme for secure authentication of medical data using Rabin, which could achieve mutual authentication between a user and a wearable sensor node and establish a secret key that is used for future secure communications. In 2021, Wang et al. [18] utilized the ECC to propose an enhanced anonymous authentication scheme for a smart healthcare system. In 2022, Hayouni [19] presented a lightweight authentication protocol for IoT-based WSNs to provide mutual authentication services for connected objects. In 2022, Nezhad et al. [20] proposed a secure routing method to prevent the intrusion of malicious nodes, consisting of star structure, key distribution, and intra-cluster communication. Hossein et al. [21] proposed a three-factor authentication scheme based on the blockchain platform for the IoT environment.

Amin and Biwas [22] first proposed a two-factor based multi-gateway authentication and key agreement protocol for WSNs that was able to ensure user anonymity while resisting password guessing attack, insider attack, stolen verifier attack, etc in. In the same year, Das et al. [23] pointed out that Amin and Biwas's scheme could not protect user anonymity. Additionally, it was vulnerable to password guessing attack, stolen smart card attack and identity guessing attack. After that, Das et al. proposed a three-factor authentication protocol based on the AES algorithm to solve these problems. However, Wu et al. [24] pointed out that the scheme in [23] also failed to resist the tracking attack and did not have a common session key for three parties. Gou et al. [25] found that the two-factor authentication scheme in [24] could not resist offline password guessing attack and identity guessing attack. In addition, the protocol in [24] was vulnerable to internal privilege attack, user tracing attack and sensor forgery attack. To address the security issues in Wu et al.'s protocol [24], Gou et al. [25] proposed a three-factor authentication protocol for multi-gateway WSNs. In 2017, Srinivas et al. [26] analyzed Amin and Biwas's scheme [22] in details and proposed an improved three-factor authentication protocol. Wang et al. [27] found that Srinivas et al.'s scheme [26] still suffered from smart card stolen attack, node capture attack, and tracking attack and could not guarantee forward secrecy. In 2019, Lee et al. [28] proposed a three-factor mutual authentication for multi-gateway protocols, in which they needed to register at all gateway nodes if users hoped to use all sensor nodes. In 2022, Dai and Xu [29] found the scheme in [25] was prone to single point of failure, and they proposed a novel elliptic curve cryptograph based three-factor authentication scheme for multi-gateway WSNs that realized smart card revocation, dynamic sensor node addition, and could withstand single point of failure. Zhao et al. [30] presented a novel three-factor authentication and key agreement protocol based on elliptic curve cryptography for IIoT environments, where their scheme can be used in single-gateway environments and can be extended to multi-gateway environments. In 2023, Chen et al. [31] proposed a two-factor multi-gateway authentication protocol based on password and smart card that could resist the joint password and identity guessing with the smart card loss attack.

A majority of schemes leverage ECC operations to ensure they can achieve more security attributes. However, ECC provides more security while producing more overhead. The Rabin mechanism is characterized by its property of computational asymmetry. The encryption performs a modular squaring operation, while the decryption performs a

module-square root operation. Rabin's encryption is significantly less than the ECC point multiple, which is the motivation that the proposed scheme leverages Rabin to build.

B. OUR CONTRIBUTIONS

1. We proposed a new three-factor user authentication and key agreement scheme using Rabin [32] for multi-gateway WSNs. Due to the lightweight computation of Rabin, it is suitable for devices with limited computational resources, such as sensor nodes. The proposed single-gateway and multi-gateway authentication schemes can also protect user privacy and achieve good forward secrecy due to the introduction of a public key system.

2. The designed authentication protocol is demonstrated to be secure using the random oracle model (ROM) [33]. As a result, the lightweight authentication scheme is secure against the Dolev-Yao adversary model.

3. Scyther [34], a verification tool, is utilized to simulate the security of the proposed protocols, and the results show that our protocols can achieve mutual authentication and can resist many attacks.

C. ORGANIZATION OF THE PAPER

The remaining sections are organized as follows: In Section II, we present the preliminary information, including the description of Rabin, the network model, and the threat model. In Section III, we propose a novel three-factor authentication and key agreement scheme for single-gateway and multi-gateway WSNs, respectively. In Section IV, we describe formal and informal security analysis and provable security. In Section V, we compare the security and efficiency of our scheme with those of related schemes. Finally, we summarize this article in Section VI.

II. PRELIMINARIES

A. RABIN

In 1979, Rabin [32] proposed a novel public-key cryptography mechanism. Selecting two different large primes p and q that satisfy $p \equiv q \equiv 3 \bmod 4$ as the private key and computing $N = pq$ as the public key. The user stores the private key and discloses the public key. Using a public key N to encrypt the message m can obtain ciphertext c, $c = m^2 \bmod N$. Decryption necessitates the use of the private keys p, q to determine $c \bmod p$ and $c \bmod q$, respectively. Subsequently, utilizing the Chinese remainder theorem (CRT), only the legitimate user in possession of the private key can calculate the decryption consequences.

The security of the Rabin cryptography mechanism is based on the intractable integer factorization problem. Since the intractable integer factorization problem (IFP) is considered computationally difficult under certain conditions, Rabin can be considered secure. In addition, Rabin is a typical asymmetric algorithm, and its encryption and decryption operations have different computational overheads. Rabin's encryption operations are lightweight, but the decryption operations consume a significant amount of resources. Therefore, encryption operations are usually performed on the side of sensor nodes with limited resources, and decryption operations can be performed on the side of gateway nodes with strong computing power.

B. NETWORK MODEL

According to literature [22], [24], [26], the network model used in the home region is a standard model for a single-gateway environment and consists of three types of entities: user nodes, gateway nodes, and sensor nodes, as shown in Figure 1. In this architecture, three types of entities can authenticate each other through two complete rounds of information interaction, which can be extended to a multi-gateway architecture. WSNs are typically unattended networks that cannot be physically changed on a large scale once they have been deployed. Due to the limited memory and computing power of sensor nodes, it is a great burden for edge sensors and gateway nodes to receive and send messages when the network size is too large or the distance is too long. Therefore, more gateways need to be extended to increase the network's capacity. Figure 2 shows us a multi-gateway architecture. In steps 1-4, the Home Gateway Node (*HGWN*) helps to establish the trust connection between the user and the Foreign Gateway Node (*FGWN*), and in steps 5-8, the user accesses the sensor via the *FGWN*.

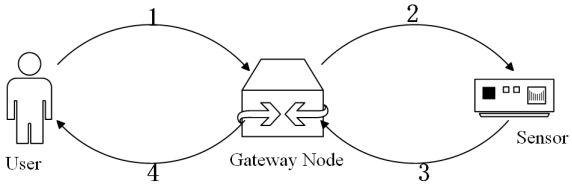


FIGURE 1. Network model of HGWN.

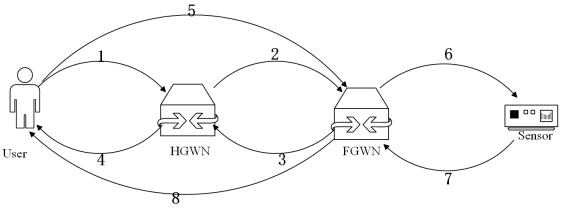


FIGURE 2. Network model of FGWN.

C. THREAT MODEL

Dolev-Yao threat model [35] is often used to formally analyze authentication protocols in communication networks, where the model assumes that two communication entities communicate over an insecure channel. WSNs can adopt a similar threat model where the channel is insecure and the terminal points cannot be generally trusted. Dolev-Yao threat model defines the precise mathematical model, and the basic assumptions listed are as follows:

1. In a perfect public key system:
   a. The one-way hash functions used are unbreakable.
   b. The public directory is secure and cannot be tampered with.
2. We will assume the following about an adversary $\mathscr{A}$:
   a. An adversary can obtain any message passing through the network.
   b. An adversary can be a legitimate user of a network and, in particular, can initiate a conversation with any other user.
   c. An adversary will be able to act as a receiver for any sender.

III. PROPOSED SCHEME

Our designed protocol is divided into seven phases: initialization phase, user registration phase, sensor registration phase, user login phase, authentication and key agreement in *HGWN*, authentication and key agreement in *FGWN*, and user key update.

A. INITIALIZATION PHASE

During the initialization phase, *SA* selects the identity SID_j for each sensor, sharing this global devices list with all gateways. After that, *SA* selects two different large random numbers p_h, q_h for home gateway node (*HGWN*), $p_h \equiv q_h \equiv 3 \bmod 4$ (p_h, q_h are congruent with module 4) and computes the public key $N_h = p_h q_h$. In the same way as above, the foreign gateway node (*FGWN*) gets a pair of private keys, which are two different large prime numbers p_f, q_f, where $p_f \equiv q_f \equiv 3 \bmod 4$ (p_f, q_f are congruent with module 4). *FGWN* computes the public key $N_f = p_f q_f$. Finally, two cryptographic collision-resistant one-way hash functions $h_1(\cdot) : \{0,1\}^* \rightarrow \{0,1\}^l$ and $h_2(\cdot) : \{0,1\}^* \rightarrow \{0,1\}^*$ are selected, where l is the output length of hash function. Table 1 shows the symbols used in this paper.

B. USER REGISTRATION PHASE

Biometric features are added in this scheme to improve the security of the system and can be well used in the setting of an authentication scheme because of their uniqueness. Compared with low-entropy passwords, biometric features also have the advantages of being difficult to forge and share and not being easy to lose. Figure 3 illustrates the registration procedure.

Step1: U_i inputs his/her own identity ID_i, password PW_i and the biometric information BIO_i. After that, the fuzzy extractor computes biometric key data σ_i and common parameter θ_i using $Gen(BIO_i) \rightarrow (\sigma_i, \theta_i)$. The common parameter θ_i is stored in SC_i. Next, U_i computes $HID_i = h_1(ID_i||\sigma_i)$ and $HPW_i = h_1(PW_i||\sigma_i)$, and transmits $\{HID_i, HPW_i\}$ to *HGWN* over a secure connection.

Step2: After getting the messages HID_i and HPW_i from U_i, *HGWN* creates a random number r_h, and computes $A_i = h_1(HID_i||p_h||q_h||r_h) \oplus HID_i$, $B_i = h_1(HID_i||r_h||HPW_i)$, and $C_i = HID_i \oplus r_h$. The long-term secret $\{HID_i, r_h\}$ is stored in *HGWN*'s memory. *HGWN* delivers a message $\{A_i, B_i, C_i\}$ to U_i over a secure connection.

TABLE 1. Symbol description.

Symbol	Description
SA	i-th user node
SN_j	j-th sensor node
SC_i	Smart card of U_i
$HGWN$	Home gateway node
$FGWN$	Foreign gateway node
ID_i	Identity of U_i
SID_j	Identity of SN_j
ID_{hg}	Identity of *HGWN*: $\{0,1\}^l$
ID_{fg}	Identity of *FGWN*: $\{0,1\}^l$
PW_i	Password of U_i
BIO_i	Biometric information of U_i
p_u, q_u	Large secret prime numbers for certain User
p_h, q_h	Large secret prime numbers for *HGWN*
p_f, q_f	Large secret prime numbers for *FGWN*
N_u	Public key of certain User $N_u = p_u q_u$
N_h	Public key of the *HGWN* $N_h = p_h q_h$
N_f	Public key of the *FGWN* $N_f = p_f q_f$
$r_u, r_h, r_{hg}\ r_s, r_f, r_{fn}$	Random numbers: $\{0,1\}^l$
$T_1, T_2, ..., T_8$	Timestamps
ΔT	Acceptable maximum transmission delay
SK	Session key
$h_1(\cdot)$	One-way hash function: $\{0,1\}^* \rightarrow \{0,1\}^l$
$h_2(\cdot)$	One-way hash function: $\{0,1\}^* \rightarrow \{0,1\}^*$
$\oplus$	Exclusive-or operation
$\|\|$	Concatenation operation

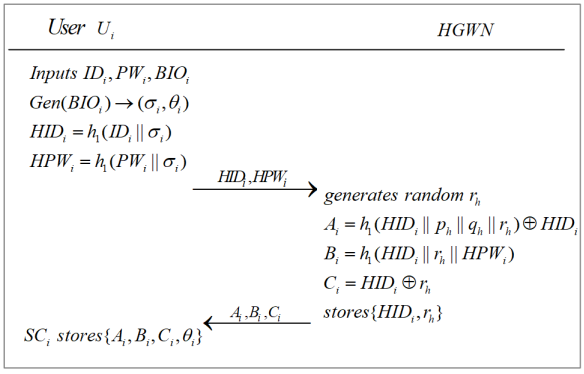


FIGURE 3. User registration phase.

Step3: After receiving the message $\{A_i, B_i, C_i\}$, long-term secret $\{A_i, B_i, C_i, \theta_i\}$ is preserved in U_i's SC_i.

C. SENSOR REGISTRATION PHASE

Each sensor node is given a distinct identity by *SA*. In order to register, SN_j transmits its own identity SID_j to the nearby *HGWN* over a secure channel. The registration procedure is described in Figure 4.

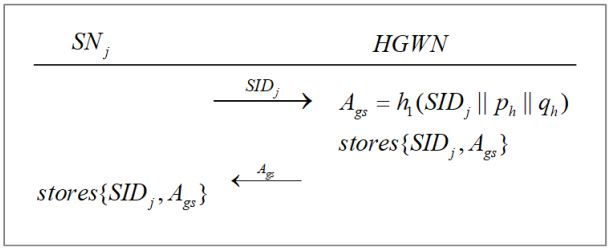


FIGURE 4. Sensor registration phase.

Step1: SN_j utilizes a secure connection to convey its own identity SID_j to the nearest *HGWN*.

Step2: After receiving the message from SN_j, *HGWN* computes $A_{gs} = h_1(SID_j||p_h||q_h)$ and maintains the long-term secret $\{SID_j, A_{gs}\}$ in its memory. Then *HGWN* transmits message A_{gs} to SN_j over a secure connection.

Step3: When SN_j receives the message A_{gs}, SN_j preserves the long-term secret $\{SID_j, A_{gs}\}$ in its own memory.

D. USER LOGIN PHASE

Before U_i enters his/her identity ID_i, password PW_i, and biometric information BIO_i, U_i inserts his/her smart card SC_i into a terminal. Next, the terminal utilizes a fuzzy extractor to recover the biometric key data σ_i, that is $Rep(BIO_i, \theta_i) \rightarrow \sigma_i$. After that, the terminal figures out $HID_i = h_1(ID_i||\sigma_i)$ and $HPW_i = h_1(PW_i||\sigma_i)$, and reads the secret parameters stored in its memory, computing $r_h^* = HID_i \oplus C_i$ and $B_i^* = h_1(HID_i||r_h^*||HPW_i)$. The terminal verifies $B_i^* \stackrel{?}{=} B_i$. When the verification is successful, U_i is able to log in. Otherwise, the login request will be rejected and no further procedure will be carried out.

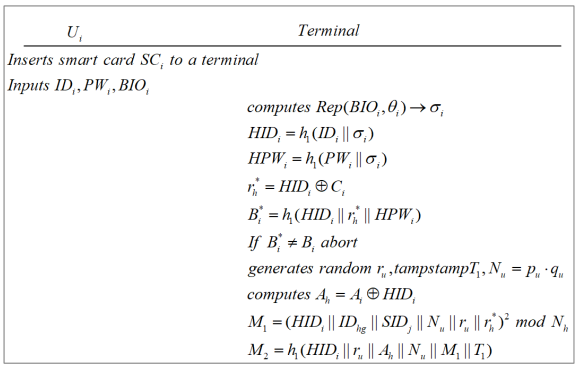


FIGURE 5. User login phase.

If user login is successful, generating a random number r_u, timestamp T_1, two large prime numbers p_u, q_u and public key $N_u = p_u \cdot q_u$, user calculates $A_h = A_i \oplus HID_i$, $M_1 = (HID_i||ID_{hg}||SID_j||N_u||r_u||r_h^*)^2 \ mod \ N_h$ and $M_2 = h_1(HID_i||r_u||A_h||N_u||M_1||T_1)$. Figure 5 demonstrates the entire procedure.

E. AUTHENTICATION AND KEY AGREEMENT PHASE IN HGWN

When U_i requires access to the SN_j in the home region where the SN_j's identity is in the registered devices database of *HGWN*, no foreign gateway is required. The mutual authentication between U_i and SN_j and the establishment of session keys can be performed through the *HGWN* only. The network model of the system is shown in Figure 6.

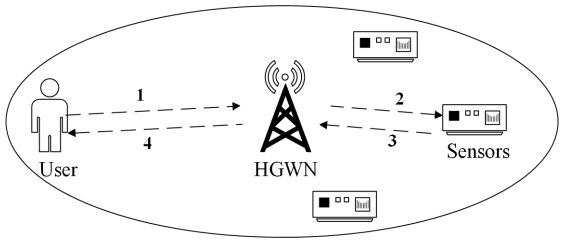


FIGURE 6. HGWN model.

Next, the authentication and key agreement phase in *HGWN* is described in the following steps, as shown in Figure 7.

Step1: U_i forwards the message $\{M_1, M_2, T_1\}$ to the closest *HGWN*.

Step2: When *HGWN* receives the message from U_i, *HGWN* obtains the current timestamp T_1^* and verifies T_1's validity, i.e., $|T_1^* - T_1| < \Delta T$. *HGWN* will refuse the current session if the timestamp T_1 is out of date. Otherwise, *HGWN* decrypts M_1 with its own private keys p_h, q_h and then obtains $HID_i, ID_{hg}, SID_j, N_u, r_u, r_h^*$. Then *HGWN* checks whether r_h^* is equal to r_h preserved in its own memory. If $r_h^* = r_h$, then it performs subsequent calculations, otherwise, it aborts the current session. *HGWN* computes $A_h^* = h_1(HID_i||p_h||q_h||r_h^*)$ and $M_2^* = h_1(HID_i||r_u||A_h^*||N_u||M_1||T_1)$ and verifies whether M_2^* is equal to the received M_2. If $M_2^* \neq M_2$, the current session is terminated. Otherwise, *HGWN* creates a random number r_{hg}, a new timestamp T_2, and computes $A_{gs} = h_1(SID_j||p_h||q_h)$, $M_3 = (HID_i||r_u||r_{hg}||N_u) \oplus h_2(A_{gs}||T_2||SID_j)$ and $M_4 = h_1(HID_i||SID_j||r_{hg}||N_u||A_{gs}||T_2)$. Finally, *HGWN* transmits the message $\{M_3, M_4, T_2\}$ to SN_j.

Step3: After receiving the message *HGWN* sent, SN_j acquires the current timestamp T_2^* and checks T_2's validity, i.e., $|T_2^* - T_2| < \Delta T$. If the timestamp T_2 is out of date, SN_j rejects the current session. Otherwise, SN_j computes $h_2(A_{gs}||T_2||SID_j)$ and calculates $h_2(A_{gs}||T_2||SID_j) \oplus M_3$ to obtain $(HID_i||r_u||r_{hg}||N_u)$. SN_j computes $M_4^* = h_1(HID_i||SID_j \ ||r_{hg}||N_u||A_{gs}||T_2)$ and checks whether M_4^* is equal to the received M_4. If $M_4^* \neq M_4$, SN_j aborts the current session. Otherwise, SN_j creates a random number r_s, a new timestamp T_3, and computes $M_5 = (HID_i||SID_j||A_{gs}||r_s)^2 \mod N_u$, $SK = h_1(HID_i||SID_j||r_u||r_s)$, $M_6 = (h_1(SK||A_{gs}))^2 \mod N_h$, $M_7 = h_1(HID_i||SID_j||r_{hg}||A_{gs}||M_5||M_6||T_3)$ and $M_8 = h_1(HID_i||SID_j||SK||r_u||r_s||M_5)$. Finally, SN_j sends the message $\{M_5, M_6, M_7, M_8, T_3\}$ to *HGWN*.

Step4: After receiving the message from SN_j, *HGWN* first obtains the current timestamp T_3^* and checks T_3's validity, i.e., $|T_3^* - T_3| < \Delta T$. *HGWN* cancels the current session if the timestamp T_3 is not current. Otherwise, *HGWN* computes $M_7^* = h_1(HID_i||SID_j||r_{hg}\ ||A_{gs}||M_5||M_6||T_3)$, and inspects $M_7^* \stackrel{?}{=} M_7$. If $M_7^* \neq M_7$, the current session is terminated by *HGWN*. Otherwise, *HGWN* creates a new timestamp T_4 and figures out $M_9 = h_1(M_5||M_8||r_u||A_h||T_4)$, and finally sends the message $\{M_5, M_8, M_9, T_4\}$ to U_i.

Step5: After getting the message from *HGWN*, U_i first acquires the current timestamp T_4^* and confirms T_4's validity, i.e., $|T_4^* - T_4| < \Delta T$. If the timestamp T_4 is not fresh, U_i aborts the current session. Otherwise, U_i calculates $M_9^* = h_1(M_5||M_8||r_u||A_h||T_4)$ and verifies whether M_9^* is equal to the received M_9. If $M_9^* \neq M_9$, U_i rejects the current session. Otherwise, U_i decrypts M_5 with its own private keys p_u, q_u and then obtains $HID_i, SID_j, A_{gs}, r_s$ and calculates $SK^* = h_1(HID_i||SID_j||r_u||r_s)$ and $M_8^* = h_1(HID_i||SID_j||SK^*||r_u||r_s||M_5)$, and inspects whether M_8^* is equal to the received M_8. If $M_8^* \neq M_8$, the current session is rejected by U_i. Otherwise, U_i calculates $M_6^* = h_1(SK^*||A_{gs})$ and sends M_6^* to *HGWN*.

Step6: *HGWN* decrypts M_6 using its own private keys p_h, q_h to obtain $h_1(SK||A_{gs})$. *HGWN* checks $M_6^* \stackrel{?}{=} h_1(SK||A_{gs})$. If $M_6^* \neq h_1(SK||A_{gs})$, *HGWN* aborts the current session. Otherwise, the authentication is successful. U_i and SN_j share a session key SK.

F. AUTHENTICATION AND KEY AGREEMENT PHASE IN FGWN

If U_i needs access to SN_j in the foreign region, where SID_j is not in *HGWN*'s database of registered devices, it is necessary to use a foreign gateway to authenticate between U_i and SN_j. In this case, the network model of the system is shown in Figure 8.

The authentication and key agreement phase in *FGWN* is described in the following steps, as shown in Figure 9 and Figure 10.

Step1: U_i generates the message $\{M_1, M_2, T_1\}$ as in user login phase, and transmits it to the nearest *HGWN*.

Step2: When *HGWN* receives the message from U_i, *HGWN* acquires the current timestamp T_1^* and checks T_1's availability, i.e., $|T_1^* - T_1| < \Delta T$. If the timestamp T_1 is not fresh, *HGWN* aborts the current session. Otherwise, *HGWN* decrypts M_1 with its own private keys p_h, q_h and then obtains $HID_i, ID_{hg}, SID_j, N_u, r_u, r_h^*$. Then *HGWN* verifies whether r_h^* is equal to r_h in its own memory. If $r_h^* = r_h$, then it performs subsequent calculations, otherwise, it aborts the current session. *HGWN* computes $A_h^* = h_1(HID_i||p_h||q_h||r_h^*)$ and $M_2^* = h_1(HID_i||r_u||A_h^*||N_u||M_1||T_1)$, and verifies whether M_2^* is equal to the received M_2. If $M_2^* \neq M_2$, *HGWN* terminates the current session. In WSNs, *HGWN* broadcasts the destination sensor identity SID_j to the remaining gateway nodes. If any *FGWN* detects SID_j in its database, it will respond to *HGWN*. At this point, the *FGWN* generates a random number r_{fn} and computes $M_3 = (SID_j||N_f||r_{fn})^2 \bmod N_h$, where N_h is the public key of the *HGWN*. *FGWN* sends M_3 as the reaction to the *HGWN*. Then *HGWN* will contact this *FGWN* in the following steps. *HGWN* decrypts M_3 using its own private keys and obtains SID_j, N_f, r_{fn}. *HGWN* creates a new timestamp T_2 and calculates $M_4 = (HID_i||SID_j||A_h||r_h||T_2)^2 \bmod N_f$, where N_f is the public key of the above *FGWN*. *HGWN* computes $M_5 = h_1(HID_i||r_h||A_h||M_4||r_{fn}||T_2)$ In the end, *HGWN* sends the message $\{M_4, M_5\}$ to *FGWN*.

Step3: After acquiring the message from *HGWN*, *FGWN* decrypts M_4 with its own private keys p_f, q_f to obtain $HID_i, SID_j, A_h, r_h, T_2$. *FGWN* gets the current timestamp T_2^* and checks T_2's validity, i.e., $|T_2^* - T_2| < \Delta T$. The current session is rejected by *FGWN* if the timestamp T_2 is out of date. Otherwise, *FGWN* figures out $M_5^* = h_1(HID_i||r_h||A_h||M_4||r_{fn}||T_2)$, and checks whether M_5^* is equal to the received M_5. If $M_5^* \neq M_5$, *FGWN* aborts the current session. Otherwise, *FGWN* constructs a random number r_f and a new timestamp T_3. After that, *FGWN* computes $A_f = h_1(HID_i||p_f||q_f||r_f)$, $M_6 = (HID_i||ID_{fg}||A_f||r_f||r_h||T_3)^2 \bmod N_h$. Finally, *FGWN* computes $M_7 = h_1(HID_i||A_f||M_6||r_f||r_h||T_3)$ and then sends the message $\{M_6, M_7\}$ to *HGWN*.

Step4: After receiving the message from *FGWN*, *HGWN* decrypts M_6 with its own private keys p_h, q_h to obtain $HID_i, ID_{fg}, A_f, r_h, r_f, T_3$. *HGWN* obtains the current timestamp T_3^* and checks T_3's validity, i.e., $|T_3^* - T_3| < \Delta T$. If the timestamp T_3 is not fresh, *HGWN* aborts the current session. *HGWN* checks whether r_h^* is equal to the received r_h. If $r_h^* = r_h$, then it performs the subsequent calculations, otherwise, it aborts the current session. *HGWN* computes $M_7^* = h_1(HID_i||A_f||M_6||r_f||r_h||T_3)$ and verifies whether M_7^* is equal to the received M_7. If $M_7^* \neq M_7$, *HGWN* aborts the current session. Otherwise, *HGWN* generates a new timestamp T_4 and computes $M_8 = A_f \oplus A_h$, $M_9 = h_1(HID_i||SID_j||M_8||A_f||T_4)$ and $R_1 = h_1(SID_j||HID_i||A_h||A_f||T_4) \oplus r_f$. Finally, *HGWN* sends the message $\{M_8, M_9, T_4, R_1\}$ to U_i.

Step5: When U_i receives the message *HGWN* transmitted, U_i first obtains the current timestamp T_4^* and verifies T_4's validity, i.e., $|T_4^* - T_4| < \Delta T$. If the timestamp T_4 is not fresh, U_i aborts the current session. Otherwise, U_i calculates $A_f^* = M_8 \oplus A_h$, $r_f^* = R_1 \oplus h_1(SID_j||HID_i||A_h||A_f^*||T_4)$ and $M_9^* = h_1(HID_i||SID_j||M_8||A_f^*||T_4)$. U_i inspects whether M_9^* is equal to the received M_9. If $M_9^* \neq M_9$, U_i aborts the current session. Otherwise, U_i creates a new random number r_u, a new timestamp T_5, and figures out $M_{10} = (HID_i||SID_j||N_u||r_u||r_f^*)^2 \bmod N_f$ and $M_{11} = h_1(HID_i||r_u||A_f^*||N_u||M_{10}||T_5)$. U_i sends message $\{M_{10}, M_{11}, T_5\}$ to *FGWN*.

Step6: Upon getting the message U_i sent, *FGWN* acquires the current timestamp T_5^* and verifies T_5's validity, i.e., $|T_5^* - T_5| < \Delta T$. If the timestamp T_5 is not fresh, the current session is terminated by *FGWN*. Otherwise, *FGWN* decrypts M_{10} with its own private keys p_f, q_f, and then obtains

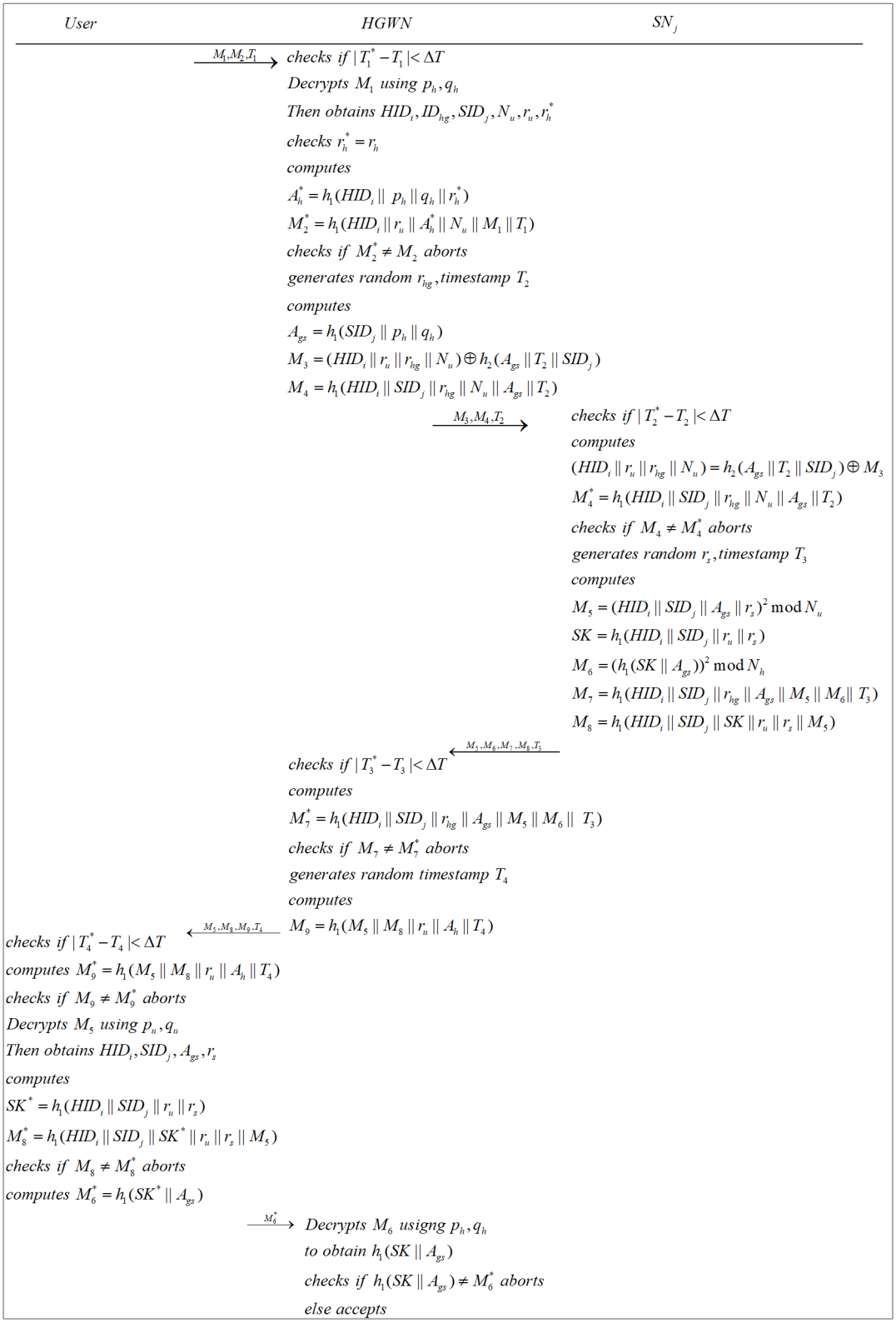


FIGURE 7. Authentication and key agreement in HGWN.

$HID_i, ID_{fg}, SID_j, N_u, r_u, r_f^*$. Then *FGWN* checks whether r_f^* is equal to r_f in its own memory. If $r_f^* = r_f$, then it performs subsequent calculations. Otherwise, it terminates the current session. *FGWN* calculates $A_f^* = h_1(HID_i||p_f||q_f||r_f^*)$ and $M_{11}^* = h_1(HID_i||r_u||A_f^*||N_u||M_{10}||T_5)$. *FGWN* inspects whether M_{11}^* is equal to the received M_{11}.

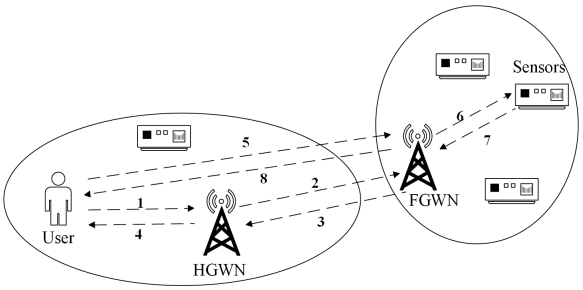


FIGURE 8. FGWN model.

If $M_{11}^* \neq M_{11}$, *FGWN* aborts the current session. Otherwise, *FGWN* produces a random number r_{fg}, a new timestamp T_6, and computes $A_{fs} = h_1(SID_j||p_f||q_f)$, $M_{12} = (HID_i||r_u||r_{fg}||N_u) \oplus h_2(A_{fs}||T_6||SID_j)$, and $M_{13} = h_1(HID_i||SID_j||r_{fg}||N_u||A_{fs}||T_6)$. Finally, *FGWN* delivers the message $\{M_{12}, M_{13}, T_6\}$ to SN_j.

Step7: After receiving the message *FGWN* transmitted, SN_j acquires the current timestamp T_6^* and checks T_6's validity, i.e., $|T_6^* - T_6| < \Delta T$. The current session is refused by SN_j if the timestamp T_6 is not fresh. Otherwise, SN_j calculates $h_2(A_{fs}||T_6||SID_j)$ and calculates $h_2(A_{fs}||T_6||SID_j) \oplus M_{12}$ to obtain $(HID_i||r_u||r_{fg}||N_u)$. SN_j computes $M_{13}^* = h_1(HID_i||SID_j||r_{fg}||N_u||A_{fs}||T_6)$ and checks whether M_{13}^* is equal to the received M_{13}. If $M_{13}^* \neq M_{13}$, SN_j aborts the current session. Otherwise, SN_j creates a random number r_s, a new timestamp T_7, and computes $M_{14} = (HID_i||SID_j||A_{fs}||r_s)^2 \bmod N_u$, $SK = h_1(HID_i||SID_j||r_u||r_s)$. SN_j computes $M_{15} = (h_1(SK||A_{fs}))^2 \bmod N_f$, $M_{16} = h_1(HID_i||\ SID_j||r_{fg}||A_{fs}||M_{14}||M_{15}||T_7)$ and $M_{17} = h_1(HID_i||SID_j||SK||r_u||r_s||M_{14})$. Finally, SN_j sends the message $\{M_{14}, M_{15}, M_{16}, M_{17}, T_7\}$ to *FGWN*.

Step8: On receiving the message SN_j sent, *FGWN* first gets the current timestamp T_7^* and inspects T_7's validity, i.e., $|T_7^* - T_7| < \Delta T$. If the timestamp T_7 is not fresh, *FGWN* terminates the current session. Otherwise, *FGWN* figures out $M_{16}^* = h_1(HID_i||SID_j||r_{fg}\ ||A_{fs}||M_{14}||M_{15}||T_7)$, and inspects whether M_{16}^* is equal to the received M_{16}. If $M_{16}^* \neq M_{16}$, *FGWN* rejects the current session. Otherwise, *FGWN* creates a new timestamp T_8 and computes $M_{18} = h_1(M_{14}||M_{17}||r_u||A_f||T_8)$, and finally sends the message $\{M_{14}, M_{17}, M_{18}, T_8\}$ to U_i.

Step9: When U_i receives the message *FGWN* transmitted, U_i first acquires the current timestamp T_8^* and verifies T_8's validity, i.e., $|T_8^* - T_8| < \Delta T$. If the timestamp T_8 is not fresh, U_i aborts the current session. Otherwise, U_i calculates $M_{18}^* = h_1(M_{14}||M_{17}||r_u||A_f||T_8)$, and verifies whether M_{18}^* is equal to the received M_{18}. If $M_{18}^* \neq M_{18}$, U_i terminates the current session. Otherwise, U_i decrypts M_{14} with its own private keys p_u, q_u, and then obtains $HID_i, SID_j, A_{fs}, r_s$. U_i calculates $SK^* = h_1(HID_i||SID_j||r_u||r_s)$, $M_{17}^* = h_1(HID_i||SID_j||SK^*||r_u||r_s||M_{14})$, and inspects whether M_{17}^* is equal to the received M_{17}. If $M_{17}^* \neq M_{17}$, U_i refuses the current session. Otherwise, U_i computes $M_{15}^* = h_1(SK^*||A_{fs})$ and sends M_{15}^* to *FGWN*.

Step10: *FGWN* decrypts M_{15} using its own private keys p_f, q_f to obtain $h_1(SK||A_{fs})$. *FGWN* checks $M_{15}^* \stackrel{?}{=} h_1(SK||A_{fs})$. If $M_{15}^* \neq h_1(SK||A_{fs})$, *FGWN* aborts the current session. Otherwise, the authentication is successful. U_i and SN_j share a session key SK.

G. USER PASSWORD UPDATE PHASE

The user password update phase does not need the help of the gateway. When a user needs to update his/her password information, the user needs to input his/her identity ID_i, old password PW_i^{old}, and biometric information BIO_i. Then the terminal regenerates the secret data σ_i, i.e., $Rep(BIO_i, \theta_i) \rightarrow \sigma_i$. After that, the terminal computes $HID_i = h_1(ID_i||\sigma_i)$ and $HPW_i^{old} = h_1(PW_i^{old}||\sigma_i)$, and reads the secret parameters stored in SC_i to compute $r_h^* = HID_i \oplus C_i$ and $B_i^* = h_1(HID_i||r_h^*||HPW_i^{old})$. The terminal verifies $B_i^* \stackrel{?}{=} B_i$. If $B_i^* = B_i$, U_i login is successful. After that, U_i inputs his/her new password PW_i^{new}. The terminal calculates $HPW_i^{new} = h_1(PW_i^{new}||\sigma_i)$ and updates $B_i^{new} = h_1(HID_i||r_h||HPW_i^{new})$ stored in SC_i. Then a password update is completed.

IV. SECURITY ANALYSIS OF THE PROPOSED SCHEME

A. FORMAL VERIFICATION

This subsection focuses on the security of the proposed authentication scheme, utilizing the Dolev-Yao model as the foundation and the Scyther [34] formal analysis tool in order to more comprehensively and systematically assess the security of our proposed scheme. Since the authentication and key agreement phase is the core of this scheme and this phase runs on an insecure wireless public channel, this subsection focuses on the security simulation of the authentication and key agreement.

Figure 11 shows the result of the formal security analysis of the *HGWN*'s authentication and key agreement. Similarly, the analysis of *FGWN*'s result is shown in Figure 12. From the simulation results, it can be seen that the scheme successfully passes the security check of Scyther, which verifies its security and functionality.

B. INFORMAL SECURITY ANALYSIS

1) MUTUAL AUTHENTICATION

Our scheme ensures mutual authentication between user nodes, gateway nodes, and wireless sensor nodes. In the home region, *HGWN* verifies U_i by utilizing M_2, and U_i can check the legitimacy of *HGWN* by relying on M_9. At the same time, SN_j and *HGWN* can achieve mutual authentication using M_4, M_7. U_i can check the legitimacy of SN_j utilizing M_8. *HGWN* uses M_6 to help SN_j check the legitimacy of U_i and confirm the shared SK.

There is a identical procedure in the foreign region. U_i and *HGWN* can achieve mutual authentication by relying on M_2, M_9. *HGWN* and *FGWN* can achieve mutual

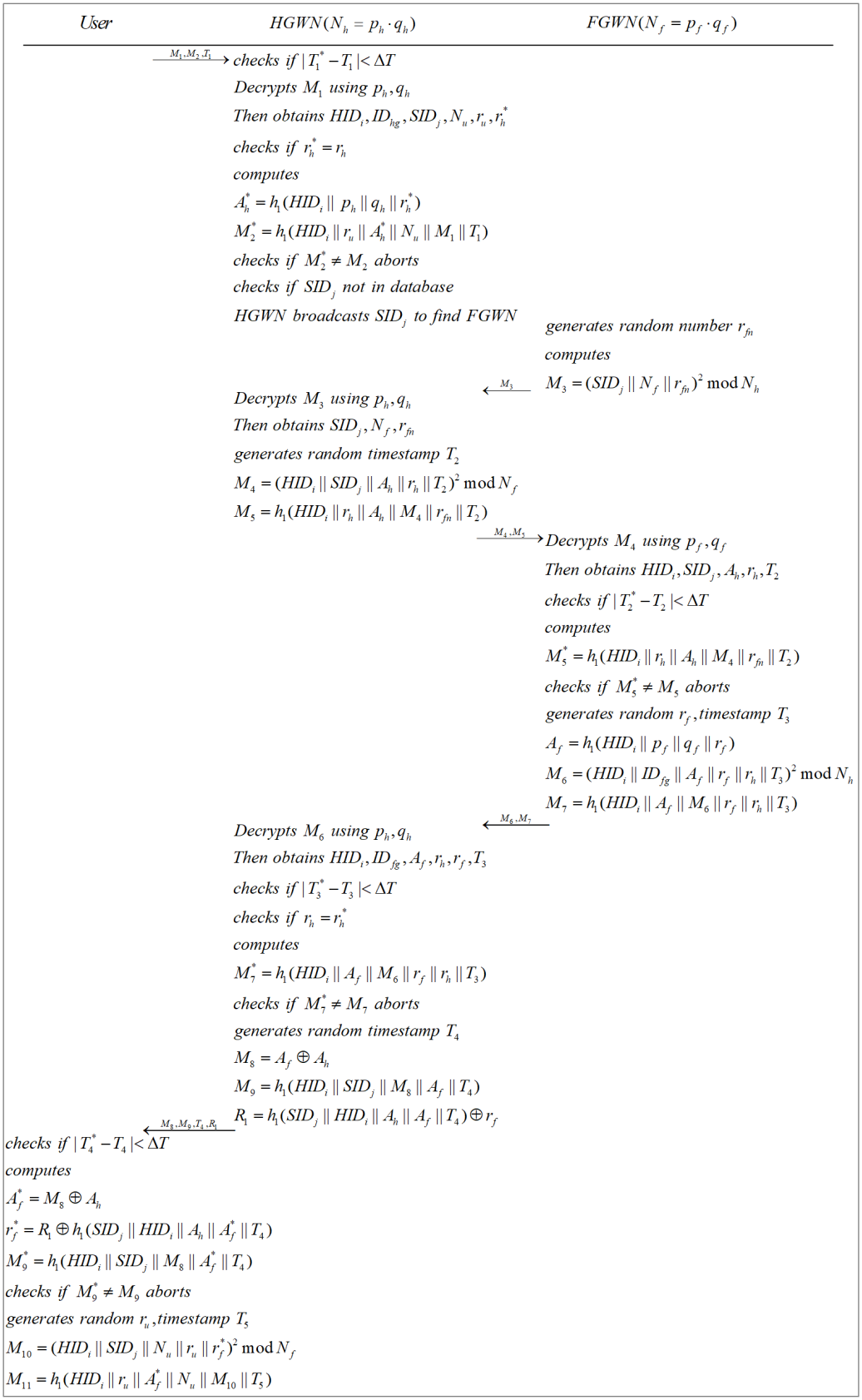


FIGURE 9. Authentication and key agreement phase 1 in FGWN.

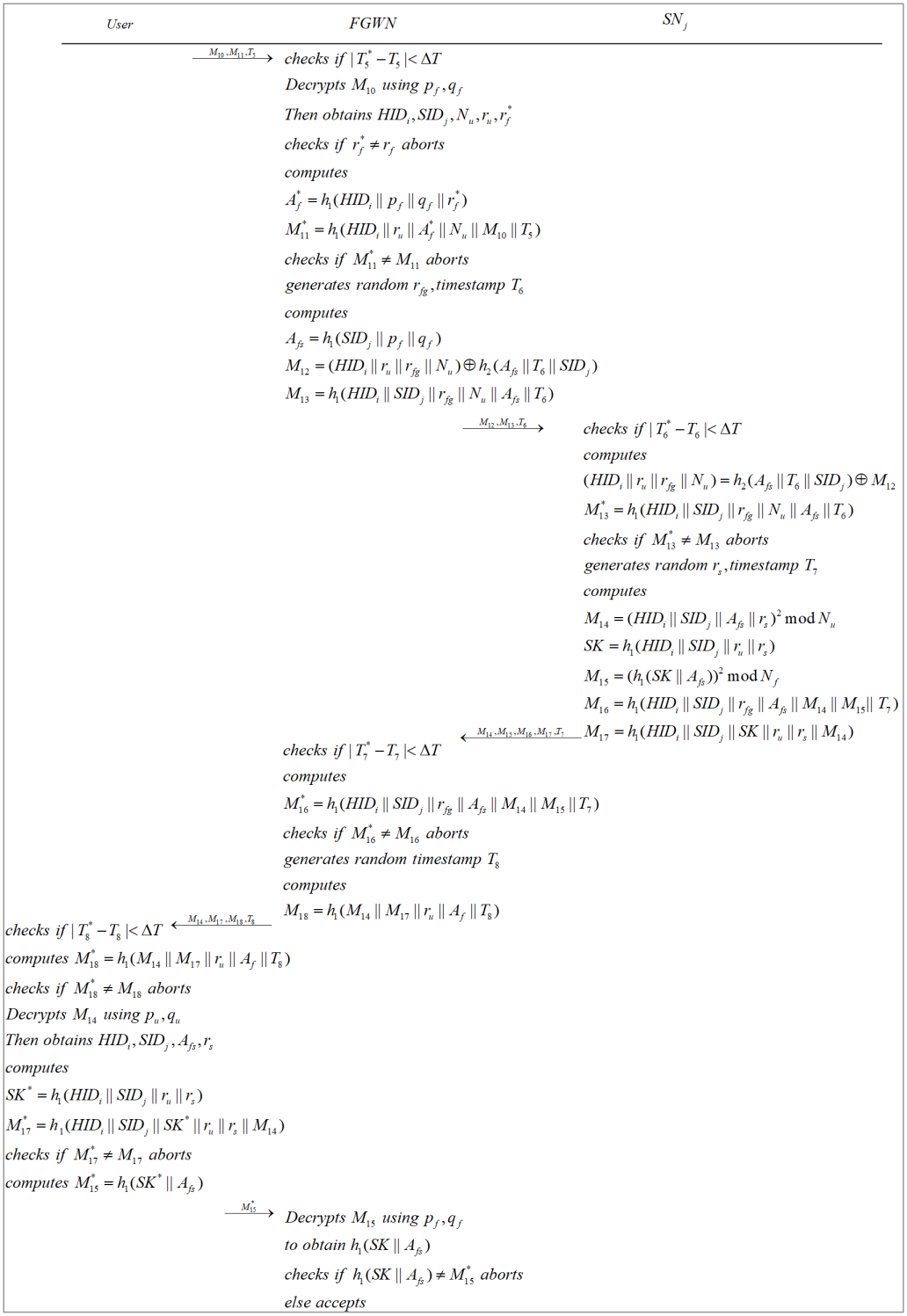
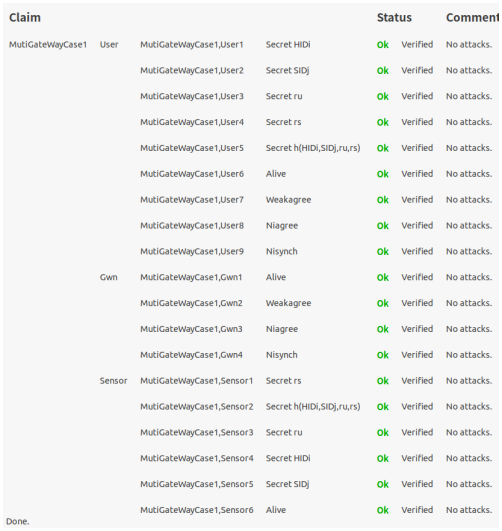


FIGURE 10. Authentication and key agreement phase 2 in FGWN.

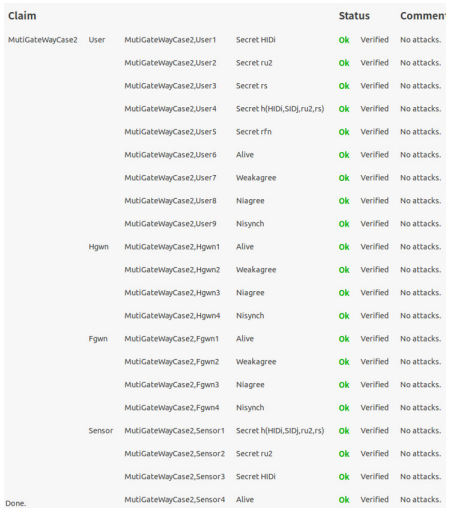
authentication using M_5, M_7. *FGWN* and U_i can achieve mutual authentication using M_{11}, M_{18}. Meanwhile, SN_j and *FGWN* can achieve mutual authentication utilizing M_{13}, M_{16}. U_i can check the legitimacy of SN_j by relying on M_{17} and *FGWN* helps SN_j to check the legitimacy of U_i and confirm the sharing of SK utilizing M_{15}.



Claim				Status		Comment
MutiGateWayCase1	User	MutiGateWayCase1,User1	Secret HIDi	Ok	Verified	No attacks.
		MutiGateWayCase1,User2	Secret SIDj	Ok	Verified	No attacks.
		MutiGateWayCase1,User3	Secret ru	Ok	Verified	No attacks.
		MutiGateWayCase1,User4	Secret rs	Ok	Verified	No attacks.
		MutiGateWayCase1,User5	Secret h(HIDi,SIDj,ru,rs)	Ok	Verified	No attacks.
		MutiGateWayCase1,User6	Alive	Ok	Verified	No attacks.
		MutiGateWayCase1,User7	Weakagree	Ok	Verified	No attacks.
		MutiGateWayCase1,User8	Niagree	Ok	Verified	No attacks.
		MutiGateWayCase1,User9	Nisynch	Ok	Verified	No attacks.
	Gwn	MutiGateWayCase1,Gwn1	Alive	Ok	Verified	No attacks.
		MutiGateWayCase1,Gwn2	Weakagree	Ok	Verified	No attacks.
		MutiGateWayCase1,Gwn3	Niagree	Ok	Verified	No attacks.
		MutiGateWayCase1,Gwn4	Nisynch	Ok	Verified	No attacks.
	Sensor	MutiGateWayCase1,Sensor1	Secret rs	Ok	Verified	No attacks.
		MutiGateWayCase1,Sensor2	Secret h(HIDi,SIDj,ru,rs)	Ok	Verified	No attacks.
		MutiGateWayCase1,Sensor3	Secret ru	Ok	Verified	No attacks.
		MutiGateWayCase1,Sensor4	Secret HIDi	Ok	Verified	No attacks.
		MutiGateWayCase1,Sensor5	Secret SIDj	Ok	Verified	No attacks.
		MutiGateWayCase1,Sensor6	Alive	Ok	Verified	No attacks.

Done.

FIGURE 11. Formal verification result in HGWN.



Claim				Status		Comment
MutiGateWayCase2	User	MutiGateWayCase2,User1	Secret HIDi	Ok	Verified	No attacks.
		MutiGateWayCase2,User2	Secret ru2	Ok	Verified	No attacks.
		MutiGateWayCase2,User3	Secret rs	Ok	Verified	No attacks.
		MutiGateWayCase2,User4	Secret h(HIDi,SIDj,ru2,rs)	Ok	Verified	No attacks.
		MutiGateWayCase2,User5	Secret rfn	Ok	Verified	No attacks.
		MutiGateWayCase2,User6	Alive	Ok	Verified	No attacks.
		MutiGateWayCase2,User7	Weakagree	Ok	Verified	No attacks.
		MutiGateWayCase2,User8	Niagree	Ok	Verified	No attacks.
		MutiGateWayCase2,User9	Nisynch	Ok	Verified	No attacks.
	Hgwn	MutiGateWayCase2,Hgwn1	Alive	Ok	Verified	No attacks.
		MutiGateWayCase2,Hgwn2	Weakagree	Ok	Verified	No attacks.
		MutiGateWayCase2,Hgwn3	Niagree	Ok	Verified	No attacks.
		MutiGateWayCase2,Hgwn4	Nisynch	Ok	Verified	No attacks.
	Fgwn	MutiGateWayCase2,Fgwn1	Alive	Ok	Verified	No attacks.
		MutiGateWayCase2,Fgwn2	Weakagree	Ok	Verified	No attacks.
		MutiGateWayCase2,Fgwn3	Niagree	Ok	Verified	No attacks.
		MutiGateWayCase2,Fgwn4	Nisynch	Ok	Verified	No attacks.
	Sensor	MutiGateWayCase2,Sensor1	Secret h(HIDi,SIDj,ru2,rs)	Ok	Verified	No attacks.
		MutiGateWayCase2,Sensor2	Secret ru2	Ok	Verified	No attacks.
		MutiGateWayCase2,Sensor3	Secret HIDi	Ok	Verified	No attacks.
		MutiGateWayCase2,Sensor4	Alive	Ok	Verified	No attacks.

Done.

FIGURE 12. Formal verification result in FGWN.

2) SESSION KEY AGREEMENT

In the home region, U_i and SN_j establish a same symmetric session key $SK = h_1(HID_i||SID_j||r_u||r_s)$ with the help of $HGWN$. In the foreign region, a similar symmetric session key SK is shared with the help of $HGWN$ and $FGWN$.

3) USER ANONYMITY AND UNTRACEABILITY

In our authentication scheme, $\mathscr{A}$ cannot access the true identity ID_i through the transmitted messages. Only the authorized gateway node in possession of private keys p_h, q_h is able to decrypt M_1 to obtain U_i's pseudo-identity HID_i. Similarly, $\mathscr{A}$ who does not know U_i's private keys p_u, q_u and SID_j's secret parameter A_{gs} cannot decrypt M_5 and M_3. Because of the one-way nature of hash function, $\mathscr{A}$ cannot obtain U_i's pseudo-identity HID_i from M_2, M_4, M_7, M_8. In foreign region, $\mathscr{A}$ who does not know U_i's private keys p_u, q_u and SID_j's secret parameter A_{fs} is unable to decrypt M_{14} and M_{12}. Meanwhile, because of the one-way nature of hash function, $\mathscr{A}$ cannot obtain U_i's pseudo-identity HID_i from $M_{11}, M_{13}, M_{16}, M_{17}$. Due to the use of random numbers in each round of interaction, the untraceability of the user is guaranteed.

4) RESISTANCE TO SMART CARD ATTACK

Through the side channel attack, $\mathscr{A}$ can obtain the secret parameters stored in SC_i and use them. In our scheme, when the adversary obtains U_i's $\{A_i, B_i, C_i, \theta_i\}$ stored in SC_i, he/she can also not obtain the legitimate user's identity, password, and biometric information. The first step of user login is that U_i inputs his/her ID_i, PW_i, BIO_i, then the terminal computes σ_i using BIO_i and θ_i. $\mathscr{A}$ cannot calculate the true σ_i on account of having no correct BIO_i. Therefore, the request for a login will be rejected in this step. Furthermore, even if the adversary can get through this step, he/she cannot compute M_1, M_2 without the correct HID_i.

5) RESISTANCE TO REPLAY ATTACK

Timestamps are adopted in our scheme to resist the replay attack. Meanwhile, random numbers are taken during the interaction of the protocol to ensure the freshness and independence of the messages. As a result, the proposed scheme is resistant to replay attack.

6) RESISTANCE TO PRIVILEGED INSIDER ATTACK

Assuming that the adversary is a malicious privileged node, he/she can obtain a user's login request information $\{HID_i, HPW_i\}$, but due to the one-way nature of the hash function and the biometric key data σ_i, $\mathscr{A}$ cannot obtain the password PW_i through HPW_i. Thus the proposed scheme can resist the privilege insider attack.

7) ILLEGAL LOGIN DETECTION

When a user inputs an incorrect identity, password, or biometric information, the mobile device can quickly detect an illegal login and abort the session. In our scheme, when a user inputs incorrect information, the correct verification parameter B_i cannot be generated, and the mobile device will reject the user's login request. The mechanism described above can reduce communication and computation costs.

8) FORWARD SECRECY

Rabin is leveraged to ensure forward secrecy in the proposed scheme. In both home and foreign regions, U_i and SN_j can establish a common symmetric session key $SK = h_1(HID_i||SID_j||r_u||r_s)$ where r_u is a random number created by U_i and r_s is a random number generated by SN_j and updated at each session round. Even if the long-term secret values are compromised, the previous session key cannot be corrupted because the adversary has to resolve an intractable IFP in order to obtain $\{r_u, r_s\}$. The specific conditions are described in the following. If an adversary obtains $\{A_i, B_i, C_i, \theta_i\}$ that are stored in user's SC_i, he/she still cannot get $\{r_u, r_s\}$ resulting in failing to compromise

the previous session key. Similarly, there are identical results on gateway and sensor. Furthermore, even though $\mathscr{A}$ gets three tuples of long-term secrets $\{A_i, B_i, C_i, \theta_i\}$, $\{HID_i, r_h\}$, $\{SID_j, A_{gs}\}$ simultaneously, the consequence is the same as above. Therefore, this scheme achieves good forward secrecy.

9) RESISTANCE TO DESYNCHRONIZATION ATTACK

In our scheme, users and gateway nodes do not store any identical secret parameters, and all entities involved in the session do not need to update any information at the end of this session. Therefore, this scheme is resistant to desynchronization attack.

10) RESISTANCE TO IMPERSONATION ATTACK

Assuming that the adversary tries to participate in the session by impersonating as a legitimate user node, the legitimate information M_1, M_2, T_1 needs to be generated. However, since the parameters HID_i, r_h^*, A_h cannot be obtained, $\mathscr{A}$ fails to generate M_1, M_2. $\mathscr{A}$ cannot forge a legitimate authentication request in polynomial time. Moreover, $\mathscr{A}$ cannot forge legitimate information without the private key of the gateway node, so this scheme can resist the gateway node impersonation attack. Similarly, $\mathscr{A}$ needs A_{gs} to generate valid messages M_5, M_6, M_7, M_8 when simulating a legitimate sensor node, so this scheme can resist the sensor node impersonation attack.

C. PROVABLE SECURITY

1) BASIC KNOWLEDGE OF PROVABLE SECURITY

Based on the security models of previous work in literature [31], [36], we verifies the security of the proposed scheme utilizing ROM.

There are three types of communication entities in the proposed scheme, i.e., user U_i, gateway GWN, and sensor S_j. Each entity has an independent number and can be considered an oracle. Supposing that the oracles $\prod_{U_i}^u$, $\prod_{GWN}^v$, $\prod_{SN_k}^t$ for U_i,GWN, and S_j respectively, where u, v, t are instances of U_i, GWN, S_j respectively, all the above oracles can output three states $\{accept, reject, \perp\}$. When the last expected message is received, $\prod^t$ becomes $accept$; otherwise, $\prod^t$ becomes $reject$. When instances $\prod^{t_1}, \prod^{t_2}$ satisfy the three conditions below, they are called partnerships.

1. Both $\prod^{t_1}$ and $\prod^{t_2}$ are $accept$.
2. $\prod^{t_1}, \prod^{t_2}$ are mutual authentication and have identical session identifiers.
3. $\prod^{t_1}, \prod^{t_2}$ cooperate with each other, and if the session key SK shared by the user node and the sensor node has not been asked for a $Reveal$ query, $\prod_{U_i}^u$, $\prod_{SN_k}^t$ can be considered as fresh.

In the ROM, $\mathscr{A}$ can compromise the security of the authentication information and session key by using queries from oracles. The adversary model is defined as shown below, and this scheme assumes that $\mathscr{A}$ has the following capabilities:

$Execute(\prod^u, \prod^v, \prod^t)$: This query simulates a passive eavesdropping attack. If $\prod_{U_i}^u$, $\prod_{GWN}^v$, $\prod_{SN_k}^t$ meets the execution rules, the oracle executes the protocol and sends the transcript of all transmitted messages to $\mathscr{A}$.

$Send(\prod^t, m)$: This query simulates an active attack between $\mathscr{A}$ and instance $\prod_{SN_k}^t$, where $\mathscr{A}$ sends messages to $\prod_{SN_k}^t$ and $\prod_{SN_k}^t$ returns the processing result of the message to $\mathscr{A}$.

$Reveal(\prod^t)$: This query can help $\mathscr{A}$ to obtain the session key SK generated by user $\prod^t$.

$Corrupt(\prod_{U_i}^u, a)$: This query can help $\mathscr{A}$ obtain the information stored on the user's mobile device or password PW_i. The notice is that $\mathscr{A}$ is unable to obtain two types of authentication information at the same time. Otherwise, $\mathscr{A}$ will be indistinguishable from a legitimate user. There are listed as follows:

1. a = 0, $\mathscr{A}$ acquires password via this query.
2. a = 1, $\mathscr{A}$ acquires all values in the mobile device via this query.

$Test(\prod^t)$: This query simulates an active attack and can measure the semantic security of the session key. Envision a challenger who flips a coin to define a bit b. If the oracle cannot meet $accept$, it returns an empty symbol $\perp$. Otherwise, if $b = 1$, then the response is the session key at that instance, but if $b = 0$, then the response is a completely random string of the same length as the session key. The adversary's final output is a bit b' which is its own guess at the value of b. Then we say that $\mathscr{A}$ wins the security game if and only if $b' = b$.

2) PROCEDURE OF PROVABLE SECURITY

Definition 1: P is used to represent the security authentication scheme described in this section, A is used to represent $\mathscr{A}$ who can break our scheme in polynomial time, and D is used to represent the uniformly distributed password dictionary. q_{hash}, $|hash|$, q_{send}, $|D|$ and Adv_p^{IFP} are respectively used to represent the number of one-way hash queries, the space of one-way hash functions, the number of queries, the size of D and the advantage of $\mathscr{A}$ corrupt IFP. Hash function is modeled as a random oracle. There are

$$Adv_p^{AKA}(\mathscr{A}) \leq \frac{(q_{hash})^2}{|Hash|} + \frac{2q_{send}}{D} + 2Adv_p^{IFP} \tag{1}$$

Proof: *Game 0∼4* are defined to describe the entire process. For each *Game*, defining the event WG_0 represents $\mathscr{A}$ performing $Test(\prod^t)$ query and successfully guessing the bit b to win the game.

Game 0: This game simulates a real attack on the protocol by the adversary $\mathscr{A}$. At the start, choosing bit b, according to the above definition, there is:

$$Adv_p^{AKA}(\mathscr{A}) = |2Pr|WG_0| - 1| \tag{2}$$

Game 1: In this game, $\mathscr{A}$ is able to perform $Execute(\prod^u, \prod^v, \prod^t)$ query to simulate an eavesdropping attack. By performing $Test$ query, $\mathscr{A}$ can determine its return value as a session key or a random string. Analyzing the session key generates, where the user generates HID_i, r_u, M_1, the sensor node generates SID_j, r_s, M_5. $\mathscr{A}$ cannot compute

the session key without the above secret values. Accordingly,

$$Pr|WG_1| = Pr|WG_0| \tag{3}$$

Game 2: *Game 2* adds the oracles *Send* and *Hash* to the foundation of *Game 1*. In this game, the active attack is mainly simulated, where the attack tries to forge legitimate information by repeatedly querying the random oracle *Hash* to generate collisions. However, since all the messages transmitted in the channel contain random numbers and entity identifiers, the oracle *Send* will not generate collisions. According to the birthday paradox, it is obtained that:

$$Pr|WG_2| - Pr|WG_1| \leq \frac{q_{hash}^2}{2|Hash|} \tag{4}$$

Game 3: *Game 3* adds the oracle *CorruptMobileDevice* $(\prod_{U_i}^{u}, a)$ to the foundation of *Game 2*. This game mainly simulates a user's mobile device theft attack combined with a dictionary attack, where $\mathscr{A}$ tries to obtain the user's password. Suppose that the times that $\mathscr{A}$ enters the wrong password are limited by the system, thus it is obtained that:

$$Pr|WG_3| - Pr|WG_2| \leq \frac{q_{send}}{|D|} \tag{5}$$

Game 4: *Game 4* is the final game. In order to get the session key SK, $\mathscr{A}$ needs to get HID_i, SID_j and random numbers r_u, r_s. Assume that $\mathscr{A}$ can obtain the secret information stored in mobile device, and is able to obtain the information in the channel through eavesdropping attacks. However, due to the irreversibility and collision resistance of the one-way hash function, $\mathscr{A}$ cannot extract useful information from $M_2, M_3, M_4, M_7, M_8, M_9$. Similarly, in order to extract HID_i, SID_j, r_u, r_s from the messages M_1, M_5, $\mathscr{A}$ needs to have the ability to solve the *IFP*. As a result,

$$Pr|WG_4| - Pr|WG_3| \leq Adv_P^{IFP} \tag{6}$$

In addition, $\mathscr{A}$ executes all the oracles, but it does not have the advantage of correctly guessing the bit b, therefore, we have

$$Pr|WG_4| = \frac{1}{2} \tag{7}$$

According to equations (2) and (3), we can get:

$$\frac{1}{2}Adv_p^{AKA}(\mathcal{A}) = |Pr|WG_0| - \frac{1}{2}| = |Pr|WG_1| - \frac{1}{2}| \tag{8}$$

In conjunction with equations (3), (4), (5), and (6), there will be:

$$\begin{aligned}
&|Pr|WG_1| - |Pr|WG_4| \\
&\quad \leq |Pr|WG_1| - Pr|WG_2| + Pr|WG_2| - Pr|WG_4|| \\
&\quad \leq |Pr|WG_1| - Pr|WG_2| + Pr|WG_2| \\
&\qquad -Pr|WG_3| + Pr|WG_3| - Pr|WG_4|| \\
&\quad \leq |Pr|WG_1| - Pr|WG_2|| + |Pr|WG_2| \\
&\qquad -Pr|WG_3|| + |Pr|WG_3| - Pr|WG_4|| \\
&\quad \leq \frac{q_{hash}^2}{2|Hash|} + \frac{q_{send}}{|D|} + Adv_P^{IFP}
\end{aligned} \tag{9}$$

As a result, we can obtain the following consequence:

$$Adv_p^{AKA}(\mathscr{A}) \leq \frac{(q_{hash})^2}{|Hash|} + \frac{2q_{send}}{D} + 2Adv_p^{IFP} \tag{10}$$

Hence the safety and validity of our scheme are proved.

V. THE PERFORMANCE COMPARISON

This section describes the performance comparison between the proposed and other corresponding schemes. We compare the security attribute that each scheme can achieve, the computational cost, and the communication cost, respectively, where Case 1 signifies the scheme applied in the home region and Case 2 denotes the protocol created in the foreign region.

A. SECURITY ATTRIBUTE COMPARISON

Table 2 describes the security features our scheme and other pertinent schemes can achieve, where "✓" means that the security property is satisfied, "×" means that it is not, and "−" means that the security property is not mentioned in their schemes. From Table 2, none of these literature [17], [22], [23], [24], [26], [28] can achieve forward secrecy. Literature [29] can achieve forward secrecy because their scheme takes advantage of ECC operations, which produce more overhead than our scheme. It can be seen that only our scheme can achieve all the security attributes listed in Table 2. Therefore, it can be concluded that the proposed protocol has better security attributes compared with other schemes.

B. COMPARISON OF COMPUTATIONAL COST

The computational costs of authentication protocols are evaluated according to all the required calculations. To demonstrate the superiority of our scheme, the computational costs of several protocols are shown in Table 3, according to the results of experiments in [17] and [37]. "fe" denotes the fuzzy extractor, and one fe computation roughly costs 0.0171 s (s denotes second). "ed" denotes the symmetric encryption or decryption operation (using AES-128), and one ed computation roughly costs 0.0056 s. "h" denotes a hash function, and one h computation roughly costs 0.00032 s. "b" represents the run time of a BioHash operation. "m" denotes the modular squaring operation (the encryption operation in Rabin), and one m computation roughly costs 0.00088 s (when we set the length of modular |N| = 512). "qr" denotes a module-square root operation (the decryption operation in Rabin), and one qr computation roughly costs 0.0192 s. "ecm" signifies ECC point multiplication, and one ecm computation roughly costs 0.0171 s. "eca" signifies ECC point addition, and one eca computation roughly costs 0.0044 s.

Since only the computational resources of sensor nodes are usually limited in WSNs, this scheme primarily focuses on computational costs on the sensor node side. Table 3 compares the computational costs for *HGWN* and *FGWN*. Compared with [22], [23], [24], and [26], the computational costs on the sensor side increased slightly, but the performance improved. However, compared with [17] and [28] that cannot support multi-gateway access, our protocol has a much

TABLE 2. Security comparison.

Security Properties	[17]	[22]	[23]	[24]	[26]	[28]	[29]	Ours
Mutual authentication	✓	✓	✓	✓	✓	✓	✓	✓
Session key agreement	✓	✓	✓	✓	×	-	✓	✓
User anonymity	✓	×	×	×	×	✓	✓	✓
Trace attack	✓	×	✓	✓	×	✓	✓	✓
Stolen smart card attack	✓	×	✓	✓	✓	✓	✓	✓
Replay attack	✓	✓	✓	✓	✓	✓	✓	✓
Insider attack	✓	✓	✓	×	✓	-	✓	✓
Quick login detection	✓	✓	✓	×	✓	-	-	✓
Forward secrecy	×	×	×	×	×	×	✓	✓
Desynchronization Attack	×	×	✓	✓	×	✓	-	✓
Impersonation attack	✓	×	✓	×	✓	✓	✓	✓

TABLE 3. Computational cost comparison.

Protocols		User	HGWN	FGWN	Sensor
[17]	Case 1	$17h + 1m$	$9h + 1qr$	-	$12h$
[22]	Case 1	$7h$	$8h$	-	$5h$
	Case 2	$8h$	$1h$	$7h$	$5h$
[23]	Case 1	$9h + 1fe + 1ed$	$5h + 2ed$	-	$3h + 1ed$
	Case 2	$10h + 1fe + 2ed$	-	$5h + 2ed$	$4h + 1ed$
[24]	Case 1	$9h$	$11h$	-	$4h$
	Case 2	$11h$	$7h$	$7h$	$4h$
[26]	Case 1	$10h$	$14h$	-	$7h$
	Case 2	$14h$	$6h$	$17h$	$6h$
[28]	Case 1	$1fe + 14h$	$5h$	-	$9h$
[29]	Case 1	$9h + 1b + 3ecm$	$12h + 1ecm$	-	$5h + 2ecm$
	Case 2	$13h + 1b + 4ecm$	$12h + 1ecm$	$13h + 2ecm$	$5h + 2ecm$
[30]	Case 1	$9h + 1fe + 3ecm$	$8h + 2ecm$	-	$5h + 3ecm$
	Case 2	$12h + 1fe + 4ecm$	$8h + 6ecm + 2eca$	$10h + 7ecm + 2eca$	$5h + 3ecm$
Ours	Case 1	$8h + 1qr + 1fe + 1m$	$7h + 2qr$	-	$6h + 2m$
	Case 2	$11h + 1qr + 1fe + 2m$	$6h + 3qr + 1m$	$10h + 2qr + 2m$	$6h + 2m$

TABLE 4. Communication cost comparison.

Scheme		Number of Messages	Communication Cost (bits)
[17]	Case-1	3	2720
[22]	Case-1	4	2528
	Case-2	5	3008
[23]	Case-1	3	2784
	Case-2	6	4704
[24]	Case-1	4	2688
	Case-2	8	4480
[26]	Case-1	4	2368
	Case-2	7	3904
[28]	Case-1	4	2400
[29]	Case-1	4	2432
	Case-2	8	4704
[30]	Case-1	4	2848
	Case-2	8	4416
Ours	Case-1	5	5504
	Case-2	9	9600

smaller overhead on the sensor side in *HGWN*. Furthermore, the computational cost on the sensor side is significantly less than that in literatures [29], [30]. In summary, the proposed protocol achieves better security attributes with lower computational costs.

C. COMPARISON OF COMMUNICATION COST

According to literature [17], [38], the following assumptions are used to calculate the communication cost. The bit sizes required for random nonce, timestamp, hash output and ECC point are, respectively, 160 bits, 32 bits, 160 bits, and 320 bits. Additionally, modular exponentiation and inversion operations are performed using 1024-bit modulus to guarantee security. In the proposed scheme, the transmitted messages in the home region during the login and authentication phase $\{M_1, M_2, T_1\}$, $\{M_3, M_4, T_2\}$, $\{M_5, M_6, M_7, M_8, T_3\}$, $\{M_5, M_8, M_9, T_4\}$, and M_6^* require $(1024 + 160 + 32) = 1216$ bits, $(160+160+32) = 352$ bits, $(1024+1024+160+160+32) = 2400$ bits, $(1024+160+160+32) = 1376$ bits, and 160 bits, respectively.

Furthermore, the transmitted messages in the foreign region during the login and authentication phase $\{M_1, M_2, T_1\}$, $\{M_4, M_5\}$, $\{M_6, M_7\}$, $\{M_8, M_9, T_4, R_1\}$, $\{M_{10}, M_{11}, T_5\}$, $\{M_{12}, M_{13}, T_6\}$, $\{M_{14}, M_{15}, M_{16}, M_{17}, T_7\}$, $\{M_{14}, M_{17}, M_{18}, T_8\}$, and M_{15}^* require $(1024 + 160 + 32) = 1216$ bits, $(1024 + 160) = 1184$ bits, $(1024 + 160) = 1184$ bits, $(160 + 160 + 32 + 160) = 512$ bits, $(1024 + 160 + 32) = 1216$ bits, $(160 + 160 + 32) = 352$ bits, $(1024 + 1034 + 160 + 160 + 32) = 2400$ bits, $(1024 + 160 + 160 + 32) = 1376$ bits and 160 bits, respectively. An identical method is used to calculate the relative schemes' communication cost, and their results are listed in Table 4. The communication of our protocol is slight higher because we transmitted the encryption message over the open channel. We take advantage of Rabin's ability to encrypt messages to ensure the scheme's security, which produces slightly higher communication costs while achieving more security attributes.

VI. CONCLUSION

We propose a lightweight user authentication and key agreement scheme for multi-gateway based wireless sensor networks using the Rabin cryptosystem. In the proposed protocol, we used lightweight cryptographic primitives such as the hash function, the encryption of Rabin, and the XOR operation to reduce the overhead. This protocol is proven to have good authentication and confidentiality through formal analysis, informal analysis, and provable security. Furthermore, because of the introduction of the public key cryptosystem, this scheme achieves forward secrecy. Finally, compared with the corresponding schemes, the proposed scheme provides more security features while requiring less computational overhead.

REFERENCES

[1] M. El-Hajj, M. Chamoun, A. Fadlallah, and A. Serhrouchni, "Taxonomy of authentication techniques in Internet of Things (IoT)," in *Proc. IEEE 15th Student Conf. Res. Develop. (SCOReD)*, Dec. 2017, pp. 67–71.

[2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

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

[4] R. J. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: Securing sensor networks with public key technology," in *Proc. 2nd ACM Workshop Secur. Ad Hoc Sensor Netw. (SASN)*, S. Setia and V. Swarup, Eds., Washington, DC, USA, Oct. 2004, pp. 59–64.

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

[6] K. H. M. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," in *Proc. IEEE Int. Conf. Sensor Netw., Ubiquitous, Trustworthy Comput. (SUTC)*. Taiwan: IEEE Computer Society, Jun. 2006, pp. 244–251.

[7] H.-R. Tseng, R.-H. Jan, and W. Yang, "An improved dynamic user authentication scheme for wireless sensor networks," in *Proc. IEEE Global Telecommun. Conf. (IEEE GLOBECOM)*, Washington, DC, USA, Nov. 2007, pp. 986–990.

[8] D. Nyang and M. Lee, "Improvement of Das's two-factor authentication protocol in wireless sensor networks," *IACR Cryptol. ePrint Arch.*, p. 631, 2009.

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

[10] O. Cheikhrouhou, A. Koubaa, M. Boujelben, and M. Abid, "A lightweight user authentication scheme for wireless sensor networks," in *Proc. 8th ACS/IEEE Int. Conf. Comput. Syst. Appl. (AICCSA)*. Hammamet, Tunisia: IEEE Computer Society, May 2010, pp. 1–7.

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

[12] W. Han, "Weakness of a secured authentication protocol for wireless sensor networks using elliptic curves cryptography," *IACR Cryptol. ePrint Arch.*, p. 293, 2011.

[13] W. Shi and P. Gong, "A new user authentication protocol for wireless sensor networks using elliptic curves cryptography," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 4, Apr. 2013, Art. no. 730831.

[14] Y. Choi, D. Lee, J. Kim, J. Jung, J. Nam, and D. Won, "Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 14, no. 6, pp. 10081–10106, Jun. 2014.

[15] N. Dinarvand and H. Barati, "A survey and comparing RFID authentication protocols based on elliptic curve cryptography," *Majlesi J. Telecommun. Devices*, vol. 5, no. 1, pp. 1–5, 2016.

[16] N. Dinarvand and H. Barati, "An efficient and secure RFID authentication protocol using elliptic curve cryptography," *Wireless Netw.*, vol. 25, no. 1, pp. 415–428, Jan. 2019.

[17] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "Cloud centric authentication for wearable healthcare monitoring system," *IEEE Trans. Depend. Sec. Comput.*, vol. 17, no. 5, pp. 942–956, Sep. 2020.

[18] W. Yuanbing, L. Wanrong, and L. Bin, "An improved authentication protocol for smart healthcare system using wireless medical sensor network," *IEEE Access*, vol. 9, pp. 105101–105117, 2021.

[19] H. Hayouni, "AuthenIoT: A lightweight authentication protocol for the Internet of Things based wireless sensor networks," *EAI Endorsed Trans. Cloud Syst.*, vol. 7, no. 21, Mar. 2022, Art. no. 171321.

[20] M. A. Nezhad, H. Barati, and A. Barati, "An authentication-based secure data aggregation method in Internet of Things," *J. Grid Comput.*, vol. 20, no. 3, p. 29, Sep. 2022.

[21] A. Ghafouri Mirsaraei, A. Barati, and H. Barati, "A secure three-factor authentication scheme for IoT environments," *J. Parallel Distrib. Comput.*, vol. 169, pp. 87–105, Nov. 2022.

[22] R. Amin and G. P. Biswas, "A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Netw.*, vol. 36, pp. 58–80, Jan. 2016.

[23] A. K. Das, A. K. Sutrala, S. Kumari, V. Odelu, M. Wazid, and X. Li, "An efficient multi-gateway-based three-factor user authentication and key agreement scheme in hierarchical wireless sensor networks," *Secur. Commun. Netw.*, vol. 9, no. 13, pp. 2070–2092, Sep. 2016.

[24] F. Wu, L. Xu, S. Kumari, X. Li, J. Shen, K.-K.-R. Choo, M. Wazid, and A. K. Das, "An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment," *J. Netw. Comput. Appl.*, vol. 89, pp. 72–85, Jul. 2017.

[25] 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 Netw.*, vol. 95, Dec. 2019, Art. no. 101965.

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

[27] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4081–4092, Sep. 2018.

[28] J. Lee, S. Yu, K. Park, Y. Park, and Y. Park, "Secure three-factor authentication protocol for multi-gateway IoT environments," *Sensors*, vol. 19, no. 10, p. 2358, May 2019.

[29] C. Dai and Z. Xu, "A secure three-factor authentication scheme for multi-gateway wireless sensor networks based on elliptic curve cryptography," *Ad Hoc Netw.*, vol. 127, Mar. 2022, Art. no. 102768.

[30] X. Zhao, D. Li, and H. Li, "Practical three-factor authentication protocol based on elliptic curve cryptography for industrial Internet of Things," *Sensors*, vol. 22, no. 19, p. 7510, Oct. 2022.

[31] C. Chen, H. Guo, Y. Wu, Y. Gao, and J. Liu, "A novel two-factor multi-gateway authentication protocol for WSNs," *Ad Hoc Netw.*, vol. 141, Mar. 2023, Art. no. 103089.

[32] M. O. Rabin., "Digitalized signatures and public-key functions as intractable as factorization," Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. TR-212, 1979.

[33] M. Abdalla, P. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Public Key Cryptography—PKC 2005* (Lecture Notes in Computer Science), vol. 3386, S. Vaudenay, Ed. Les Diablerets, Switzerland: Springer, Jan. 2005, pp. 65–84.
[34] C. J. F. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols," in *Computer Aided Verification* (Lecture Notes in Computer Science), vol. 5123, A. Gupta and S. Malik, Eds. Princeton, NJ, USA: Springer, Jul. 2008, pp. 414–418.
[35] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–207, Mar. 1983.
[36] Y. Lu, G. Xu, L. Li, and Y. Yang, "Anonymous three-factor authenticated key agreement for wireless sensor networks," *Wireless Netw.*, vol. 25, no. 4, pp. 1461–1475, May 2019.
[37] S. Challa, M. Wazid, A. K. Das, and M. K. Khan, "Authentication protocols for implantable medical devices: Taxonomy, analysis and future directions," *IEEE Consum. Electron. Mag.*, vol. 7, no. 1, pp. 57–65, Jan. 2018.
[38] M. Shuai, L. Xiong, C. Wang, and N. Yu, "A secure authentication scheme with forward secrecy for industrial Internet of Things using Rabin cryptosystem," *Comput. Commun.*, vol. 160, pp. 215–227, Jul. 2020.

XINGWEN ZHAO received the B.S. and M.S. degrees from the School of Telecommunications Engineering, Xidian University, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree from the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China, in 2011.

He is currently an Associate Professor with the School of Cyber Engineering, Xidian University. His research interests include machine learning (or artificial intelligent)-based network security, multi-party data sharing, and anonymous authentication.

DEXIN LI received the B.S. degree from the School of Data Science and Technology, Heilongjiang University, Harbin, China, in 2019. She is currently pursuing the Ph.D. degree with the School of Cyber Engineering, Xidian University.

Her research interest includes authentication protocols in wireless sensor networks.