sensors

MDPI

Article

SLUA-WSN: Secure and Lightweight Three-Factor-Based User Authentication Protocol for Wireless Sensor Networks

SungJin Yu and YoungHo Park *

School of Electronics Engineering, Kyungpook National University, Daegu 41566, Korea; darkskiln@knu.ac.kr
* Correspondence: parkyh@knu.ac.kr; Tel.: +82-53-950-7842

Received: 11 June 2020; Accepted: 24 July 2020; Published: 25 July 2020

Abstract: Wireless sensor networks (WSN) are composed of multiple sensor nodes with limited storage, computation, power, and communication capabilities and are widely used in various fields such as banks, hospitals, institutes to national defense, research, and so on. However, useful services are susceptible to security threats because sensitive data in various fields are exchanged via a public channel. Thus, secure authentication protocols are indispensable to provide various services in WSN. In 2019, Mo and Chen presented a lightweight secure user authentication scheme in WSN. We discover that Mo and Chen's scheme suffers from various security flaws, such as session key exposure and masquerade attacks, and does not provide anonymity, untraceability, and mutual authentication. To resolve the security weaknesses of Mo and Chen's scheme, we propose a secure and lightweight three-factor-based user authentication protocol for WSN, called SLUA-WSN. The proposed SLUA-WSN can prevent security threats and ensure anonymity, untraceability, and mutual authentication. We analyze the security of SLUA-WSN through the informal and formal analysis, including Burrows–Abadi–Needham (BAN) logic, Real-or-Random (ROR) model, and Automated Verification of Internet Security Protocols and Applications (AVISPA) simulation. Moreover, we compare the performance of SLUA-WSN with some existing schemes. The proposed SLUA-WSN better ensures the security and efficiency than previous proposed scheme and is suitable for practical WSN applications.

Keywords: wireless sensor networks; authentication; BAN logic; ROR model; AVISPA simulation

1. Introduction

Wireless sensor networks (WSN) are widely exploited in terms of enormous applicability [1] and have been used in various fields such as smart homes, smart factories, healthcares, and environmental monitoring [2–8]. Generally, WSN consist of a gateway node (GWN), a user, and a sensor node (SN) which are resource-limited in smart devices (things, sensors, etc.) [9]. SNs are deployed in various fields and collect a large amount of real-time data. GWN manages data collected by deployed SNs to provide services for legitimate users.

One of the application areas of WSN is a smart home with sensor devices, which provides a better daily life for users [10,11]. A smart home provides various services for users such as automatic checking of the temperature and humidity of the house and controlling light bulbs. However, it may cause serious privacy problems [12,13] because the data collected by SNs are exchanged through a public channel. If data collected by SNs is exposed, a malicious adversary can obtain the private information of users such as daily routines and habits in the house, and also can use the information for criminal purposes. Furthermore, in these application scenarios, smart devices are resource-constrained in terms of computation, communication, and storage overheads, and it is not suitable to apply asymmetric cryptosystems that generate high computational overheads [14]. Therefore, secure and lightweight authentication and key agreement protocols are indispensable to provide secure services for legal users in WSN environments. The secure and lightweight authentication and key agreement protocols must consider the following security requirements.

- Three-factor security: The protocol must meet the three-factor security to protect the legitimate user's privacy.
- Preventing well-known attacks: The protocol for WSN must be secure against potential attacks, including smart card stolen, masquerade, privileged insider, man-in-the-middle (MITM) attacks, and so on.
- Preventing sensor node capture attack: Even if some sensors are captured by a malicious adversary, it is hard for an adversary to pretend to be other sensors.
- Preventing offline password guessing attack: The protocol must prevent the guessing of the legitimate user's real password if a malicious adversary either intercepts the transmitted messages or approaches smart card contents.
- Preventing smart card stolen attack: In this attack it is assumed that a malicious adversary can attain the stored secret parameters on the smart card, thus the knowledge of attained parameters should not be enough for the malicious adversary to attain useful information to masquerade a legal user.
- Preventing privileged insider attack: The protocol must be secure to privileged insider attacks where the insider having privileges in the database may access the secret credentials and misuse the contents.
- Anonymity and untraceability: A malicious adversary cannot reveal and trace the real identity of a legitimate user.
- User authentication and key agreement: The protocol must mutually authenticate among entities and successfully establish a secure session key.
- Confidentiality: All transmitted messages communicated between the participants must be safely transmitted using a secret credential so that only legal participants can verify the message.

In 2019, Mo and Chen [15] proposed an elliptic curve cryptosystem (ECC)-based user authentication scheme for WSN. Mo and Chen claimed that their scheme prevents various attacks and provides user anonymity, untraceability, and authentication. However, we prove that their scheme suffers from many drawbacks, including masquerade and replay and session key exposure attacks, and does not provide user anonymity, untraceability, and mutual authentication. In addition, their scheme is not suitable for WSN environments because it requires high communication and computation costs. Consequently, we propose a secure and lightweight three-factor authentication protocol for WSN (SLUA-WSN), considering the efficiency of smart devices and improving the security level of Mo and Chen's scheme [15].

1.1. Contributions and Motivations

The main contributions of our paper can be summarized as follows.

- We propose a secure and lightweight authentication protocol for WSN to resolve the security problems of Mo and Chen's scheme utilizing secret parameters and biometrics.
- We perform the Burrows–Abadi–Needham (BAN) logic analysis [16] to evaluate that SLUA-WSN ensures secure mutual authentication. We also perform formal security analysis utilizing the Real-or-Random (ROR) model [17] to prove session key security of SLUA-WSN.
- We carry out the simulation analysis using the automated verification of internet security protocols and applications (AVISPA) [18,19] to evaluate that SLUA-WSN prevents against replay and MITM attacks.

- According to the security and performance analysis, we show that the proposed SLUA-WSN achieves better security along with more features, and provides efficient computational, communication, and storage overheads as compared with related schemes.

The motivations of our paper can be summarized as follows.

- Authentication and key agreement protocols for WSN are susceptible to well-known attacks, including sensor node capture, masquerade, and replay attacks.
- Authentication and key agreement protocols for WSN should provide useful convenience for legitimate users and take into account the security requirements.
- Secure and efficient user authentication protocols are essential in WSN, which take into account limitations for resource-constrained smart devices in terms of memory and battery capacity.

We propose a secure and lightweight three-factor authentication protocol for WSN to resolve the security weaknesses of Mo and Chen's scheme [15]. The proposed SLUA-WSN presents several advantages compared with existing authentication schemes: SLUA-WSN prevents potential attacks, including sensor node capture, replay, privileged insider, and masquerade attacks, and also ensures secure untraceability, user anonymity, and mutual authentication. SLUA-WSN also uses the fuzzy extractor technique to improve the security level of the two-factor-based protocol. Even if two of the three factors are exposed, SLUA-WSN is still secure. Furthermore, SLUA-WSN provides better efficient computation and communication costs with existing schemes because it only uses the hash and XOR operations. Thus, SLUA-WSN is suitable for practical WSN environments because it is more secure and efficient than related schemes.

1.2. Organization

The rest of this article is organized as follows. We introduce the related works for WSN environments in Section 2, and present the preliminaries of this paper in Section 3. Section 4 reviews Mo and Chen's scheme and then Section 5 proves the security shortcomings of Mo and Chen's scheme. Section 6 presents a secure and lightweight user authentication protocol for WSN environments to enhance the security shortcomings of Mo and Chen's scheme. Section 7 evaluates the security analysis of SLUA-WSN by performing informal and formal analysis, including BAN logic, ROR model, and AVISPA simulation. Section 8 presents the results of the performance analysis of the SLUA-WSN compared with those of the related schemes. Finally, we conclude the paper in Section 9.

2. Related Works

In the last few decades, numerous authentication protocols have been proposed to provide user privacy in the WSN environment [20–25]. In 1981, Lamport [26] presented the password-based authentication protocol using a single factor to provide user privacy and anonymity. However, Lamport's scheme [26] was fragile to offline password guessing attacks because it relied solely on the security of the password. To improve these security problems, Das [27] presented a two-factor authentication scheme using smartcard and password. Das [27] claimed that their scheme is secure and efficient because it uses only hash functions and prevents various attacks. However, some researchers [28,29] pointed out that Das's scheme [27] has various security drawbacks. Nyang and Lee [28] showed that Das's scheme [27] is fragile to the sensor node capture and offline password guessing attacks. Nyang and Lee [28] presented a secure authentication scheme in WSN to enhance the security problems of Das's scheme. In 2010, He et al. [29] proposed a two-factor user authentication scheme for WSN. However, in 2011, Kumar and Lee [30] discovered that He et al.'s scheme [29] cannot provide mutual authentication and generate a session key between each entity. Therefore, these smartcard-based two-factor authentication protocols [27–29] were fragile to various attacks.

Numerous biometric-based three-factor authentication protocols have been proposed [31–33] to resolve the above-mentioned security issues. Compared with the existing two-factor authentication schemes using a password and smartcard, biometrics (palms, irises, and fingerprints) cannot be stolen or lost because they are very difficult to forget or lose, copy, distribute, guess, break, and forge. Thus, biometric-based three-factor authentication has a higher security level than two-factor authentication.

In recent years, many three-factor authenticated key agreement protocols have been proposed to provide various services in WSN environments [34–36]. In 2018, Wu et al. [37] presented a secure three-factor user authentication scheme for WSN. However, in 2019, Mo and Chen [15] demonstrated that if the user inputs an incorrect password at the login process in Wu et al.'s scheme [37], the smartcard does not check whether the password is verified, and the protocol will proceed until GWN finds that the login request of the user was invalid, so GWN performs unnecessary computational resources. In 2017, Wang et al. [38] presented an enhanced three-factor user authentication scheme using ECC for WSN. Unfortunately, Wang et al.'s scheme [38] is susceptible to insider attack because the random nonce for the legitimate user is stored in the database of GWN, and the insider can access and modify it so user login can result in failure. In 2018, Li et al. [39] presented a three-factor-based authentication scheme for WSN in Internet of Things (IoT) environments with adoption of fuzzy extractor to provide high security level. However, Mo and Chen [15] pointed out that Li et al.'s scheme [39] cannot provide three-factor security if the stolen/lost smartcard is obtained by the adversary. In addition, their scheme [39] is not as secure as they claimed because the biometric of the user is collected by the adversary without the awareness of the legitimate user. In 2019, Li et al. [40] presented a secure three-factor-based user authentication protocol for wireless medical sensor networks. However, Mo and Chen [15] demonstrated that their scheme [40] is vulnerable to replay attacks. In 2019, Lu et al. [41] proposed a three-factor authenticated key agreement for WSN using ECC. However, Mo and Chen [15] proved that Lu et al.'s protocol [41] cannot withstand known session-specific temporary information (KSSTI) attacks and cannot provide three-factor security along with session key security. To improve the security drawbacks of Lu et al.'s scheme, Mo and Chen [15] presented a lightweight secure user authenticated key agreement scheme for WSN using ECC. Mo and Chen [15] claimed that their scheme can prevent potential attacks and can ensure anonymity, untraceability, and authentication. However, we analyze that Mo and Chen's scheme suffers from various security threats, such as session key exposure and masquerade attacks, and cannot ensure anonymity, untraceability, and mutual authentication. In addition, Mo and Chen's scheme is not practical for WSN because ECC makes the computation and communication overheads burden very heavy. Therefore, we propose a secure and lightweight three-factor user authentication protocol in WSN, considering the efficiency of smart devices and improving security shortcomings of Mo and Chen's scheme.

3. Preliminaries

This section introduces the preliminaries to improve the readability of this paper.

3.1. Fuzzy Extractor

This section briefly discusses the concepts of a fuzzy extractor [42]. The fuzzy extractor is a cryptographic method utilizing biometrics to perform secure authentication and it comprises two operations—the generator (Gen) and reproduction (Rep)—which are presented below.

1. Gen : After users imprint the biometric input Bio, Gen generates a consistent random string $\rho \in \{0,1\}^l$ and a random auxiliary string $\sigma \in \{0,1\}^*$, which is a probabilistic function.
2. Rep : When a noisy biometric Bio_{new} is imprinted, Rep reproduces ρ using value σ, where σ is public reproduction value related with Bio.

3.2. Attacker Model

We present the well-known Dolev–Yao (DY) threat model [43] to examine the security of SLUA-WSN. In the DY model, the capabilities of the attacker are as follows.

- Referring to the DY model [43], an attacker can inject, delete, intercept, and eavesdrop the data exchanged over wireless networks.
- A malicious attacker can steal the smart card of legal users and can extract secret credentials stored in memory utilizing power-analysis [44].
- After obtaining the secret credentials of smart card, a malicious attacker may attempt various attacks, including the masquerade, offline password guessing, privileged insider, forward secrecy attacks, and so on [45,46].

3.3. System Model

In 2013, Xue et al.'s scheme [47] introduced the five basic authentication mechanism models for WSN. We adopt the first authentication mechanism model presented by Xue et al.'s scheme [47]. This authentication model for WSN consists of three entities: the user, the SN, and the GWN, as shown in Figure 1. Initially, the user contacts GWN to initiate the key agreement between them and the SN. In contrast, the SN checks whether the legitimate user and performs mutual authentication through a GWN. As a result, this model enables mutual authentication between all entities and establishes key agreement between users and corresponding sensor nodes.

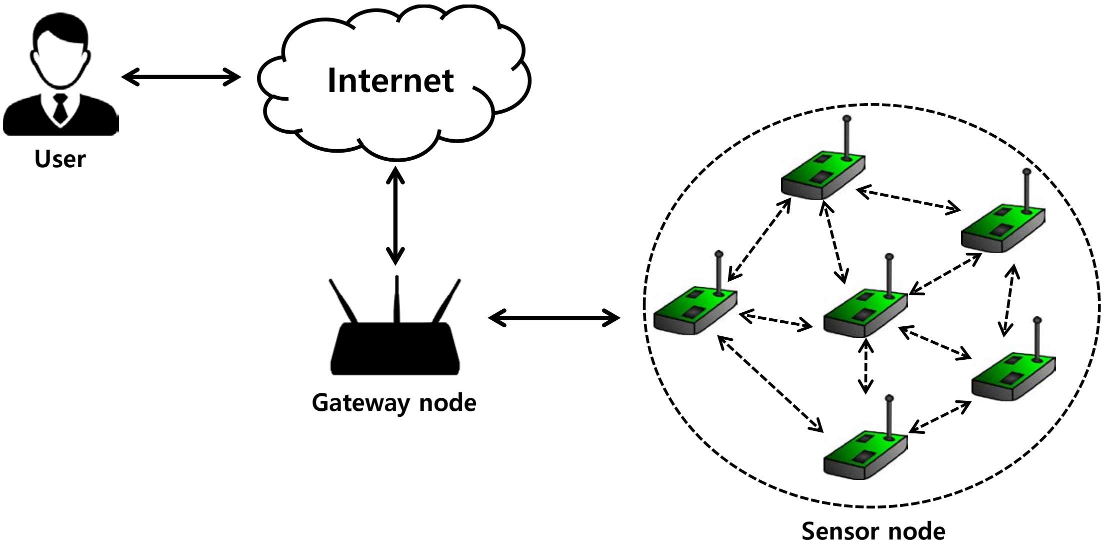


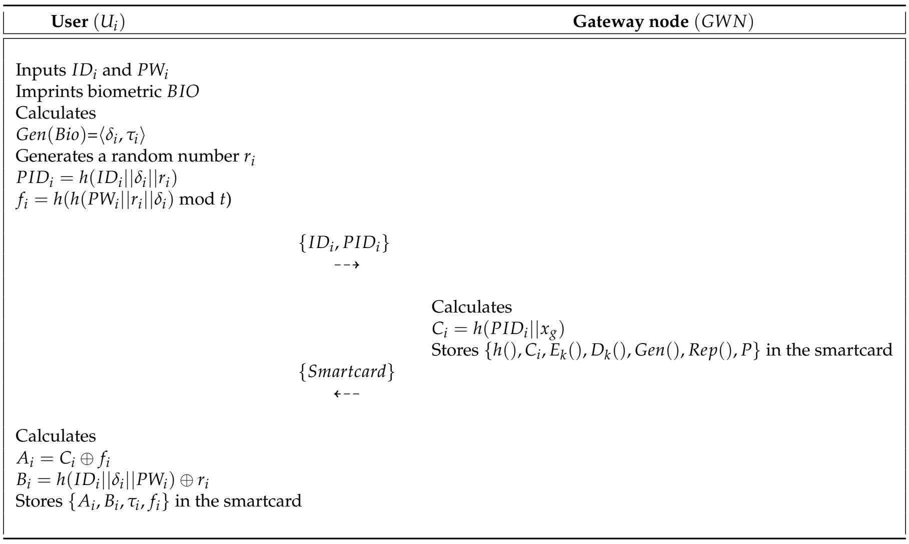
Figure 1. Authentication model in wireless sensor network.

4. Review of Mo and Chen's Scheme

Mo and Chen's scheme [15] presented a secure authentication protocol to provide useful services in WSN. This protocol comprises three entities: the user, the SN, and the GWN. Mo and Chen's scheme has four processes: pre-deployment, user registration, authentication, and password update. In the pre-deployment process, the gateway node (GWN) selects a unique identity SID_j for each sensor (S_j) and computes $K_j = h(SID_j||X_{GWN})$. Then, GWN sends $\{SID_j, K_j, P\}$ to S_j through a secure channel. Finally, S_j stores $\{SID_j, K_j, P\}$ in memory. During the user registration process, the GWN issues a smartcard to the legal user who wants to request registration through a secure channel and then helps the agreement of the session key between the S_j and the user. They presented a password update process to maintain a high level of security. Figure 2 shows the registration process of Mo and Chen's scheme, and also the detailed steps involved in the authentication and key agreement process of Mo and Chen's scheme are as shown in Figure 3. Furthermore, the password update process is described in the following subsections. Table 1 presents the notations used in this paper.

Table 1. Notations.

Notation	Description
U_i	User
GWN	Gateway node
S_j	Sensor node
ID_i	U_i's identity
PW_i	U_i's password
SID_j	S_j's identity
K_{GWN}	Master key of GWN
X_{pub}	Public key of GWN
X_j	Secret key of S_j
E/F_p	Elliptic curve E defined on the finite field F_p with order p
G	A group for an elliptic curve
P	The generator of G
E_k/D_k	Symmetric key encryption/decryption
SK	Session key
T_i	Timestamp
BIO	Biometric of U_i
$h(\cdot)$	Hash function
$\oplus$	XOR operation
$\|\|$	Concatenation operation



User (U_i)

Gateway node (GWN)

Inputs ID_i and PW_i
Imprints biometric BIO
Calculates
$Gen(Bio)=\langle \delta_i, \tau_i \rangle$
Generates a random number r_i
$PID_i = h(ID_i||\delta_i||r_i)$
$f_i = h(h(PW_i||r_i||\delta_i) \text{ mod } t)$

$\{ID_i, PID_i\}$
$\dashrightarrow$

Calculates
$C_i = h(PID_i||x_g)$
Stores $\{h(), C_i, E_k(), D_k(), Gen(), Rep(), P\}$ in the smartcard

$\{Smartcard\}$
$\dashleftarrow$

Calculates
$A_i = C_i \oplus f_i$
$B_i = h(ID_i||\delta_i||PW_i) \oplus r_i$
Stores $\{A_i, B_i, \tau_i, f_i\}$ in the smartcard

Figure 2. Registration process of Mo and Chen's scheme.

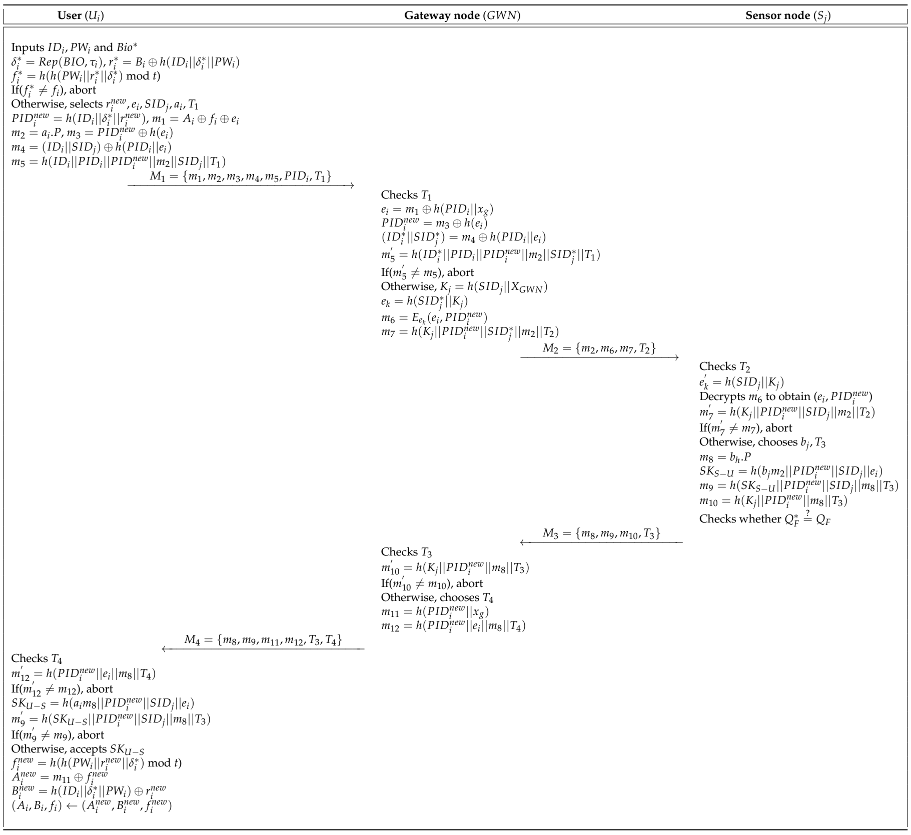

User (U_i)
Gateway node (GWN)
Sensor node (S_j)
Inputs ID_i, PW_i and Bio^*
$\delta_i^* = Rep(BIO, \tau_i), r_i^* = B_i \oplus h(ID_i||\delta_i^*||PW_i)$
$f_i^* = h(h(PW_i||r_i^*||\delta_i^*)$ mod $t)$
If($f_i^* \neq f_i$), abort
Otherwise, selects $r_i^{new}, e_i, SID_j, a_i, T_1$
$PID_i^{new} = h(ID_i||\delta_i^*||r_i^{new}), m_1 = A_i \oplus f_i \oplus e_i$
$m_2 = a_i.P, m_3 = PID_i^{new} \oplus h(e_i)$
$m_4 = (ID_i||SID_j) \oplus h(PID_i||e_i)$
$m_5 = h(ID_i||PID_i||PID_i^{new}||m_2||SID_j||T_1)$
$M_1 = \{m_1, m_2, m_3, m_4, m_5, PID_i, T_1\}$
Checks T_1
$e_i = m_1 \oplus h(PID_i||x_g)$
$PID_i^{new} = m_3 \oplus h(e_i)$
$(ID_i^*||SID_j^*) = m_4 \oplus h(PID_i||e_i)$
$m_5^{'} = h(ID_i^*||PID_i||PID_i^{new}||m_2||SID_j^*||T_1)$
If($m_5^{'} \neq m_5$), abort
Otherwise, $K_j = h(SID_j||X_{GWN})$
$e_k = h(SID_j^*||K_j)$
$m_6 = E_{e_k}(e_i, PID_i^{new})$
$m_7 = h(K_j||PID_i^{new}||SID_j^*||m_2||T_2)$
$M_2 = \{m_2, m_6, m_7, T_2\}$
Checks T_2
$e_k^{'} = h(SID_j||K_j)$
Decrypts m_6 to obtain (e_i, PID_i^{new})
$m_7^{'} = h(K_j||PID_i^{new}||SID_j||m_2||T_2)$
If($m_7^{'} \neq m_7$), abort
Otherwise, chooses b_j, T_3
$m_8 = b_h.P$
$SK_{S-U} = h(b_j m_2||PID_i^{new}||SID_j||e_i)$
$m_9 = h(SK_{S-U}||PID_i^{new}||SID_j||m_8||T_3)$
$m_{10} = h(K_j||PID_i^{new}||m_8||T_3)$
Checks whether $Q_F^* \overset{?}{=} Q_F$
$M_3 = \{m_8, m_9, m_{10}, T_3\}$
Checks T_3
$m_{10}^{'} = h(K_j||PID_i^{new}||m_8||T_3)$
If($m_{10}^{'} \neq m_{10}$), abort
Otherwise, chooses T_4
$m_{11} = h(PID_i^{new}||x_g)$
$m_{12} = h(PID_i^{new}||e_i||m_8||T_4)$
$M_4 = \{m_8, m_9, m_{11}, m_{12}, T_3, T_4\}$
Checks T_4
$m_{12}^{'} = h(PID_i^{new}||e_i||m_8||T_4)$
If($m_{12}^{'} \neq m_{12}$), abort
$SK_{U-S} = h(a_i m_8||PID_i^{new}||SID_j||e_i)$
$m_9^{'} = h(SK_{U-S}||PID_i^{new}||SID_j||m_8||T_3)$
If($m_9^{'} \neq m_9$), abort
Otherwise, accepts SK_{U-S}
$f_i^{new} = h(h(PW_i||r_i^{new}||\delta_i^*)$ mod $t)$
$A_i^{new} = m_{11} \oplus f_i^{new}$
$B_i^{new} = h(ID_i||\delta_i^*||PW_i) \oplus r_i^{new}$
$(A_i, B_i, f_i) \leftarrow (A_i^{new}, B_i^{new}, f_i^{new})$

Figure 3. Authentication process of Mo and Chen's scheme.

Password Update Process

If the authorized user requests a new password, Mo and Chen's scheme can update the password from the gateway as follows.

Step 1: U_i inputs ID_i and the old PW_i and imprints Bio^*, and inserts the smartcard (SC) in the reader. After that, the SC calculates $Gen(Bio^*) = (\delta_i^*, \tau_i^*)$, $r_i^* = B_i \oplus h(ID_i||\delta_i^*||PW_i)$, and $f_i^{'} = h(h(ID_i||r_i^*||PW_i)$ mod t and checks whether $f_i^{'} \overset{?}{=} f_i$ holds. If the condition is false, the communication is aborted.

Step 2: U_i inputs a new PW_i^{new}, computes $f_i^{new} = h(h(PW_i^{new}||r_i^*||\delta_i^*)$ mod t, $A_i^{new} = C_i \oplus f_i^{new}$, $B_i^{new} = h(ID_i||\delta_i^*||PW_i^{new}) \oplus r_i^*$ and replaces (A_i, B_i, f_i) with $(A_i^{new}, B_i^{new}, f_i^{new})$.

5. Security Flaws of Mo and Chen's Scheme

We discuss the security flaws of Mo and Chen's scheme, including session key exposure and masquerade attacks. Furthermore, we discover that Mo and Chen's scheme cannot ensure user anonymity, untraceability, and mutual authentication.

5.1. Masquerade Attack

In this attack, a malicious attacker (MA) may attempt to impersonate legal users through stolen smartcard. According to Section 3.2, we assume that MA is able to extract the secret credentials $\{A_i, B_i, \tau_i, f_i\}$ stored in the smart card. Furthermore, MA can intercept the messages exchanged over the wireless network. Therefore, MA can perform the masquerade attack as shown in the following detailed steps.

Step 1: A MA first calculates $e_i = m_1 \oplus A_i \oplus f_i$, $PID_i^{new} = m_3 \oplus h(e_i)$, $(ID_i||SID_j) = m_4 \oplus h(PID_i||e_i)$, and $m_5 = h(ID_i||PID_i||PID_i^{new}||m_2||SID_j||T_1)$. After that, the MA generates the two random numbers e_{MA}, a_{MA} and computes $m_{1MA} = A_i \oplus f_i \oplus e_{MA}$, $m_{2MA} = a_{MA}P$, $m_{3MA} = PID_i^{new} \oplus h(e_{MA})$, $m_{4MA} = (ID_i||SID_j) \oplus h(PID_i||e_{MA})$ and $m_{5MA} = h(ID_i||PID_i||PID_i^{new}||m_2||SID_j||T_1)$. The MA sends $M_1 = \{m_{1MA}, m_{2MA}, m_{3MA}, m_{4MA}, m_{5MA}, PID_i, T_1\}$ to the GWN over wireless networks.

Step 2: Upon getting the M_1, the GWN verifies the validity of T_1. If it is equal, the GWN computes $e_{MA} = m_{1MA} \oplus h(PID_i||x_g)$, $PID_i^{new} = m_{3MA} \oplus h(e_{MA})$, $(ID_i^{'}||SID_j^{'}) = m_{4MA} \oplus h(PID_i||e_{MA})$, and $m_{5MA} = h(ID_i^{'}||PID_i||PID_i^{new}||m_{2MA}||SID_j^{'}||T_1)$. Then, the GWN checks $m_{5MA}^{'} \stackrel{?}{=} m_{5MA}$. If it is correct, the GWN computes $e_k = h(SID_j^{'}||K_j)$, $m_6 = E_{e_k}(e_{MA}, PID_i^{new})$ and $m_7 = h(K_j||PID_i^{new}||SID_j^{'}||m_2||T_2)$. Next, the GWN sends $M_2 = \{m_{2MA}, m_6, m_7, T_2\}$ to the S_j.

Step 3: After getting the M_2, the S_j verifies the T_2. If it is equal, the S_j calculates $e_k^{'} = h(SID_j||K_j)$ and decrypts m_6 to get (e_{MA}, PID_i^{new}). After that, the S_j calculates $m_7^{'} = h(K_j||PID_i^{new}||SID_j||m_2||T_2)$ and then checks $m_7^{'} \stackrel{?}{=} m_7$. If the condition is equal, the S_j selects a random number b_j and timestamp T_3. Then, S_j computes $m_8 = b_jP$, $SK_{S-MA} = h(b_jm_{2MA}||PID_i^{new}||SID_j||e_{MA})$, $m_{9MA} = h(SK_{S-MA}||PID_i^{new}||SID_j||m_8||T_3)$ and $m_{10} = h(K_j||PID_i^{new}||m_8||T_3)$. Finally, S_j sends $M_3 = \{m_8, m_{9MA}, m_{10}, T_3\}$ to the GWN.

Step 4: Upon getting the M_3, the GWN verifies the validity of T_3. If the condition is equal, the GWN calculates $m_{10}^{'} = h(K_j||PID_i^{new}||m_8||T_3)$ and verifies $m_{10}^{'} \stackrel{?}{=} m_{10}$. If the condition is valid, the GWN selects T_4 and calculates $m_{11} = h(PID_i^{new}||x_g)$ and $m_{12MA} = h(PID_i^{new}||e_{MA}||m_8||T_4)$. Finally, GWN sends $M_4 = \{m_8, m_{9MA}, m_{11}, m_{12MA}, T_3, T_4\}$ to the U_i.

Step 5: After getting the M_4, the MA checks the T_4 and calculates $m_{12MA}^{'} = h(PID_i^{new}||e_{MA}||m_8||T_4)$ and checks $m_{12MA}^{'} \stackrel{?}{=} m_{12MA}$. If it is equal, the MA computes $SK_{MA-S} = h(a_{MA}||m_8||PID_i^{new}||SID_j||e_{MA})$ and $m_9^{'} = h(SK_{MA-S}||PID_i^{new}||SID_j||m_8||T_3)$.

As a result, Mo and Chen's scheme cannot prevent the masquerade attack because the MA can impersonate an legitimate user successfully.

5.2. Session Key Exposure Attack

In Mo and Chen's scheme, they claimed that their scheme could prevent to session key exposure attack because a MA could not obtain the secret credentials. However, according to Section 5.1, we prove that MA is able to impersonate legal users U_i and calculates the session key SK as follows. Referring to Section 3.2, the MA can extract secret credentials $\{A_i, B_i, \tau_i, f_i\}$ stored in the smartcard. Then, the MA is able to intercept the exchanged messages between U_i, GWN, and S_j via wireless networks. If so, the MA can calculate e_i, PID_i^{new} and $(ID_i||SID_j)$. After that, the MA selects random numbers e_{MA}, a_{MA} and can successfully generate new messages $\{m_{1MA}, m_{2MA}, m_{3MA}, m_{4MA}, m_{5MA}\}$ by utilizing e_{MA} and a_{MA}. Consequently, the MA can successfully perform the session key exposure attack by calculating $SK_{MA-S} = h(a_{MA}||m_8||PID_i^{new}||SID_j||e_{MA})$ and disguise as legitimate users.

5.3. Anonymity and Untraceability

Referring to Section 5.1, the MA can trace a legitimate user U_i and can obtain the real identities $\{ID_i, SID_j\}$ of U_i and S_j. The MA computes $e_i = m_1 \oplus A_i \oplus f_i$ utilizing secret credentials $\{A_i, f_i\}$ stored in the smart card. After that, the MA can compute $(ID_i||SID_j) = m_4 \oplus h(PID_i||e_i)$, $PID_i^{new} = m_3 \oplus h(e_i)$, and $m_5 = h(ID_i||PID_i||PID_i^{new}||m_2||SID_j||T_1)$ successfully. Thus, Mo and Chen's scheme does not ensure user anonymity and untraceability.

5.4. Mutual Authentication

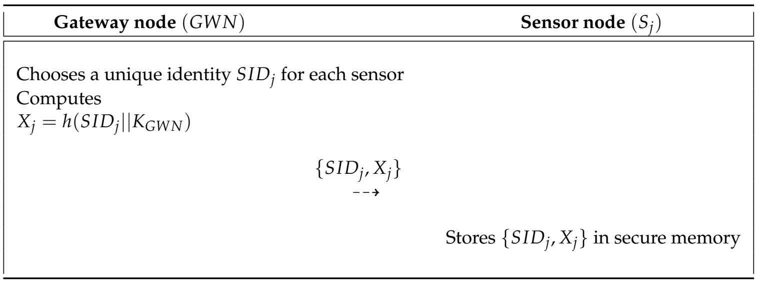
Mo and Chen's scheme asserted that their scheme provides secure mutual authentication among the U_i, GWN, and S_j. However, referring to Section 5.1, the MA can generate authentication request message $m_{5MA} = h(ID_i||PID_i||PID_i^{new}||m_2||SID_j||T_1)$, response message $m_{12MA} = h(PID_i^{new}||e_{MA}||m_8||T_4)$, and then can calculate session key $SK_{MA-S} = h(a_{MA}||m_8||PID_i^{new}||SID_j||e_{MA})$. As a result, we prove that their scheme cannot provide correct mutual authentication among U_i, GWN, and S_j.

6. Proposed Scheme

We present a secure and lightweight user authentication protocol in WSN to improve the security flaws of [15]. The proposed SLUA-WSN comprises the same process as that Mo and Chen's scheme. The details of the four processes are shown below.

6.1. Pre-Deployment Process

This process is similar to the pre-deployment process given in Mo and Chen's scheme [15]. In Figure 4, we show the user registration process of SLUA-WSN and the detailed steps are below.



Gateway node (GWN) **Sensor node** (S_j)

Chooses a unique identity SID_j for each sensor
Computes
$X_j = h(SID_j||K_{GWN})$

$\{SID_j, X_j\}$ $\dashrightarrow$

Stores $\{SID_j, X_j\}$ in secure memory

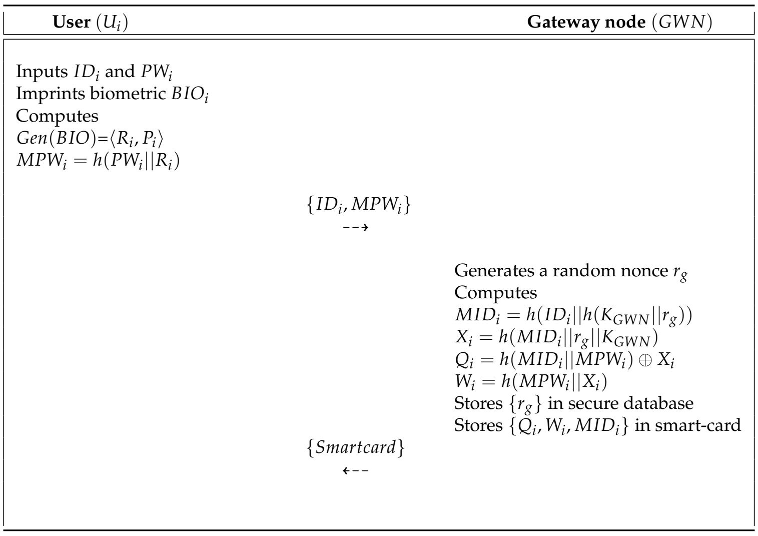
Figure 4. Pre-deployment process of the proposed scheme.

Step 1: GWN selects a unique identity SID_j for sensors and computes $X_j = h(SID_j||K_{GWN})$. Finally, GWN sends $\{SID_j, X_j\}$ to the S_j over a secure communication.

Step 2: Upon receiving the messages, the S_j stores them in secure memory.

6.2. User Registration Process

The U_i must register within GWN to access various services. In Figure 5, we show the user registration process of SLUA-WSN and the detailed steps are below.



User (U_i) **Gateway node** (GWN)

Inputs ID_i and PW_i
Imprints biometric BIO_i
Computes
$Gen(BIO) = \langle R_i, P_i \rangle$
$MPW_i = h(PW_i||R_i)$

$\{ID_i, MPW_i\}$ $\dashrightarrow$

Generates a random nonce r_g
Computes
$MID_i = h(ID_i||h(K_{GWN}||r_g))$
$X_i = h(MID_i||r_g||K_{GWN})$
$Q_i = h(MID_i||MPW_i) \oplus X_i$
$W_i = h(MPW_i||X_i)$
Stores $\{r_g\}$ in secure database
Stores $\{Q_i, W_i, MID_i\}$ in smart-card

$\{Smartcard\}$ $\dashleftarrow$

Figure 5. User registration process of our scheme.

Step 1: U_i inputs the ID_i and PW_i and imprints biometric BIO_i. Then, the U_i computes $Gen(BIO)$=$\langle R_i, P_i\rangle$ and $MPW_i = h(PW_i||R_i)$, and sends $\{ID_i, MPW_i\}$ to the GWN over a secure communication.

Step 2: After reception of messages, the GWN generates a random nonce r_g and calculates $MID_i = h(ID_i||h(K_{GWN}||r_g))$, $X_i = h(MID_i||r_g||K_{GWN})$, $Q_i = h(MID_i||MPW_i) \oplus X_i$ and $W_i = h(MPW_i||X_i)$, and then stores $\{r_g\}$ in secure database. After that, the GWN stores $\{Q_i, W_i, MID_i\}$ in the smart card and issues it to the U_i.

6.3. Authentication Process

After performing the registration process, the registered U_i requests authentication to the GWN in order to establish the session key. In Figure 6, we show the authentication process of SLUA-WSN and the detailed steps are below.

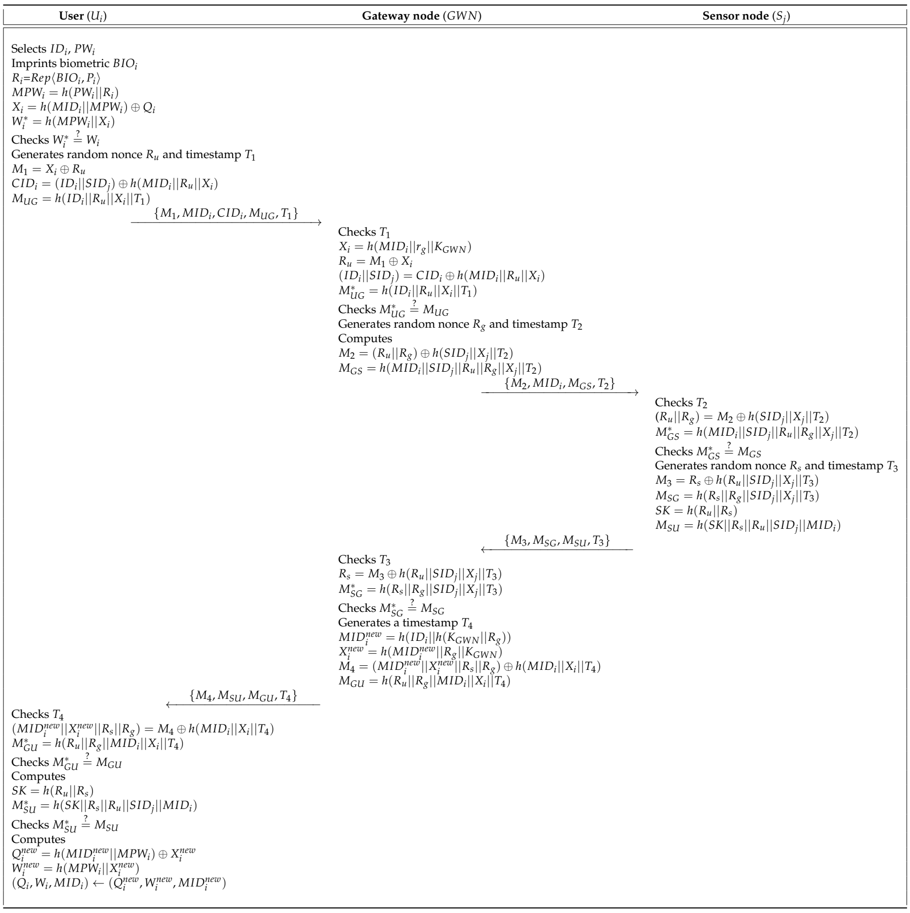

User (U_i)
Gateway node (GWN)
Sensor node (S_j)
Selects ID_i, PW_i
Imprints biometric BIO_i
R_i=$Rep\langle BIO_i, P_i\rangle$
$MPW_i = h(PW_i||R_i)$
$X_i = h(MID_i||MPW_i) \oplus Q_i$
$W_i^* = h(MPW_i||X_i)$
Checks $W_i^* \overset{?}{=} W_i$
Generates random nonce R_u and timestamp T_1
$M_1 = X_i \oplus R_u$
$CID_i = (ID_i||SID_j) \oplus h(MID_i||R_u||X_i)$
$M_{UG} = h(ID_i||R_u||X_i||T_1)$
$\xrightarrow{\{M_1, MID_i, CID_i, M_{UG}, T_1\}}$
Checks T_1
$X_i = h(MID_i||r_g||K_{GWN})$
$R_u = M_1 \oplus X_i$
$(ID_i||SID_j) = CID_i \oplus h(MID_i||R_u||X_i)$
$M_{UG}^* = h(ID_i||R_u||X_i||T_1)$
Checks $M_{UG}^* \overset{?}{=} M_{UG}$
Generates random nonce R_g and timestamp T_2
Computes
$M_2 = (R_u||R_g) \oplus h(SID_j||X_j||T_2)$
$M_{GS} = h(MID_i||SID_j||R_u||R_g||X_j||T_2)$
$\xrightarrow{\{M_2, MID_i, M_{GS}, T_2\}}$
Checks T_2
$(R_u||R_g) = M_2 \oplus h(SID_j||X_j||T_2)$
$M_{GS}^* = h(MID_i||SID_j||R_u||R_g||X_j||T_2)$
Checks $M_{GS}^* \overset{?}{=} M_{GS}$
Generates random nonce R_s and timestamp T_3
$M_3 = R_s \oplus h(R_u||SID_j||X_j||T_3)$
$M_{SG} = h(R_s||R_g||SID_j||X_j||T_3)$
$SK = h(R_u||R_s)$
$M_{SU} = h(SK||R_s||R_u||SID_j||MID_i)$
$\xleftarrow{\{M_3, M_{SG}, M_{SU}, T_3\}}$
Checks T_3
$R_s = M_3 \oplus h(R_u||SID_j||X_j||T_3)$
$M_{SG}^* = h(R_s||R_g||SID_j||X_j||T_3)$
Checks $M_{SG}^* \overset{?}{=} M_{SG}$
Generates a timestamp T_4
$MID_i^{new} = h(ID_i||h(K_{GWN}||R_g))$
$X_i^{new} = h(MID_i^{new}||R_g||K_{GWN})$
$M_4 = (MID_i^{new}||X_i^{new}||R_s||R_g) \oplus h(MID_i||X_i||T_4)$
$M_{GU} = h(R_u||R_g||MID_i||X_i||T_4)$
$\xleftarrow{\{M_4, M_{SU}, M_{GU}, T_4\}}$
Checks T_4
$(MID_i^{new}||X_i^{new}||R_s||R_g) = M_4 \oplus h(MID_i||X_i||T_4)$
$M_{GU}^* = h(R_u||R_g||MID_i||X_i||T_4)$
Checks $M_{GU}^* \overset{?}{=} M_{GU}$
Computes
$SK = h(R_u||R_s)$
$M_{SU}^* = h(SK||R_s||R_u||SID_j||MID_i)$
Checks $M_{SU}^* \overset{?}{=} M_{SU}$
Computes
$Q_i^{new} = h(MID_i^{new}||MPW_i) \oplus X_i^{new}$
$W_i^{new} = h(MPW_i||X_i^{new})$
$(Q_i, W_i, MID_i) \leftarrow (Q_i^{new}, W_i^{new}, MID_i^{new})$

Figure 6. Authentication process of our scheme.

Step 1: U_i first inserts the smart card and inputs ID_i and PW_i. Then, the U_i imprints BIO_i and computes R_i=$Rep\langle BIO_i, P_i\rangle$, $MPW_i = h(PW_i||R_i)$, $X_i = h(MID_i||MPW_i) \oplus Q_i$, and $W_i^* = h(MPW_i||X_i)$, and then checks $W_i^*? = W_i$. If the condition is valid, the U_i generates a random nonce R_u and a timestamp T_1. The U_i computes $M_1 = X_i \oplus R_u$, $CID_i = (ID_i||SID_j) \oplus h(MID_i||R_u||X_i)$, and $M_{UG} = h(ID_i||R_u||X_i||T_1)$, and sends $\{M_1, MID_i, CID_i, M_{UG}, T_1\}$ to the GWN over an insecure channel.

Step 2: Upon reception of messages, the GWN checks the validity of T_1 and calculates $X_i = h(MID_i||r_g||K_{GWN})$, $R_u = M_1 \oplus X_i$, $(ID_i||SID_j) = CID_i \oplus h(MID_i||R_u||X_i)$ and $M_{UG}^* = h(ID_i||R_u||X_i||T_1)$ and then, checks $M_{UG}^* \overset{?}{=} M_{UG}$. If the condition is correct, the GWN calculates $M_2 = (R_u||R_g) \oplus h(SID_j||X_j||T_2)$ and $M_{GS} = h(MID_i||SID_j||R_u||R_g||X_j||T_2)$, and sends $\{M_2, MID_i, M_{GS}, T_2\}$ to the S_j.

Step 3: After reception of messages, the S_j checks the validity of T_2 and computes $(R_u||R_g) = M_2 \oplus h(SID_j||X_j||T_2)$ and $M^*_{GS} = h(MID_i||SID_j||R_u||R_g||X_j||T_2)$ and checks $M^*_{GS} \overset{?}{=} M_{GS}$. If it is valid, the S_j generates a random nonce R_s and timestamp T_3 and calculates $M_3 = R_s \oplus h(R_u||SID_j||X_j||T_3)$, $M_{SG} = h(R_s||R_g||SID_j||X_j||T_3)$, $SK = h(R_u||R_s)$, and $M_{SU} = h(SK||R_s||R_u||SID_j||MID_i)$, and then sends $\{M_3, M_{SG}, M_{SU}, T_3\}$ to the GWN over an insecure channel.

Step 4: Upon reception of messages, the GWN checks the validity of T_3 and calculates $R_s = M_3 \oplus h(R_u||SID_j||X_j||T_3)$ and $M^*_{SG} = h(R_s||R_g||SID_j||X_j||T_3)$, and checks $M^*_{SG} \overset{?}{=} M_{SG}$. If it is valid, the GWN generates a timestamp T_4 and computes $MID_i^{new} = h(ID_i||h(K_{GWN}||R_g))$, $X_i^{new} = h(MID_i^{new}||R_g||K_{GWN})$, $M_4 = (MID_i^{new}||X_i^{new}||R_s||R_g) \oplus h(MID_i||X_i||T_4)$, and $M_{GU} = h(R_u||R_g||MID_i||X_i||T_4)$ and sends $\{M_4, M_{SU}, M_{GU}, T_4\}$ to the U_i.

Step 5: After reception of messages, the U_i checks the validity of T_4 and computes $(MID_i^{new}||X_i^{new}||R_s||R_g) = M_4 \oplus h(MID_i||X_i||T_4)$ and $M^*_{GU} = h(R_u||R_g||MID_i||X_i||T_4)$, and then checks $M^*_{GU} \overset{?}{=} M_{GU}$. If the condition is valid, the U_i computes $SK = h(R_u||R_s)$ and $M^*_{SU} = h(SK||R_s||R_u||SID_j||MID_i)$, and checks $M^*_{SU} \overset{?}{=} M_{SU}$. If the condition is correct, the U_i computes $Q_i^{new} = h(MID_i^{MPW}||MPW_i) \oplus X_i^{new}$, and $W_i^{new} = h(MPW_i||X_i^{new})$ and replaces $\{Q_i, W_i, MID_i\}$ with $\{Q_i^{new}, W_i^{new}, MID_i^{new}\}$. Consequently, the U_i, the GWN and S_j are mutually authenticated successfully.

6.4. Password Change Process

In SLUA-WSN, an authorized U_i can freely update their password. The detailed steps of the password change process are below.

Step 1: U_i inputs $ID_i^{'}$ and $PW_i^{'}$ and imprints biometric $BIO_i^{'}$. After that, the U_i computes $Gen(BIO^{'})$=$\langle R_i^{'}, P_i^{'}\rangle$ and $MPW_i^{'} = h(PW_i^{'}||R_i^{'})$ and then sends $\{ID_i^{'}, MPW_i^{'}\}$ to the SC over a secure communication.

Step 2: Upon reception of messages, the SC calculates $X_i^{'} = Q_i^{'} \oplus h(MID_i^{'}||MPW_i^{'})$ and $W_i^{'} = h(MPW_i^{'}||X_i^{'})$ and sends authentication message to the U_i.

Step 3: After reception of messages, the U_i chooses a new PW_i^{new} and imprints a new BIO^{new}. Then, the U_i calculates $Gen(BIO^{new})$=$\langle R_i^{new}, P_i^{new}\rangle$ and $MPW_i^{new} = h(PW_i^{new}||R_i^{new})$ and sends $\{MPW_i^{new}\}$ to the SC over a secure channel.

Step 4: Upon reception of messages, the SC calculates $Q_i^{new} = h(MID_i^{'}||MPW_i^{new}) \oplus X_i^{'}$ and $W_i^{new} = h(MPW_i^{new}||X_i^{'})$ and then replaces $\{Q_i^{'}, W_i^{'}\}$ with $\{Q_i^{new}, W_i^{new}\}$ successfully.

7. Security Analysis

This section assessed the security of SLUA-WSN by using informal and formal security analysis such as BAN logic, ROR model, and AVISPA simulation, which are widely known security models.

7.1. Informal Security Analysis

The security of SLUA-WSN is assessed by performing an informal security analysis. We show that SLUA-WSN can resist potential security threats, including masquerade, sensor node capture, replay, and privileged insider attacks, and ensure secure authentication and anonymity.

7.1.1. Masquerade Attack

In this attack, the MA attempts to masquerade a legitimate user by intercepting messages transmitted over an insecure channel. However, the MA cannot generate the request messages

$\{M_1, MID_i, CID_i, M_{UG}\}$ in the proposed SLUA-WSN correctly. The MA cannot compute the request messages because MA cannot get U_i's real identity ID_i, the biometric BIO, and the random nonce R_u. As a result, SLUA-WSN resists masquerade attacks.

7.1.2. Replay Attack

Assuming that the MA attempts the replay attack utilizing previously exchanged data over an insecure channel, even if the MA intercepts the request message $\{M_1, MID_i, CID_i, M_{UG}, T_1\}$ in the previous session, the proposed SLUA-WSN verifies the freshness of the timestamp. In addition, the request messages are protected with secret parameter X_i and random nonce R_u. Thus, SLUA-WSN prevents replay attacks.

7.1.3. Sensor Node Capture Attack

As sensor nodes are typically placed in unmanned or hostile areas, the MA can easily capture sensor nodes. However, each S_j has a unique SID_j and a secret parameter X_j. Even if some sensor nodes are captured by the MA, it is difficult to impersonate that the MA is another sensor. Therefore, the MA does not have any ability to compromise other SK established between the U_i and non-compromised S_j. Thus, SLUA-WSN prevents sensor node capture attacks.

7.1.4. Privileged Insider Attack

In this attack, the privileged insider is able to access the password of the user stored in GWN and disguises the user to log in to other systems. However, the user in the proposed SLUA-WSN only sends $\{ID_i, MPW_i\}$ to the GWN during the registration process. Consequently, SLUA-WSN prevents privileged insider attacks because the privileged insider cannot obtain the real password of the legitimate user.

7.1.5. Anonymity and Untraceability

We assume that the MA can extract secret credentials stored in a smartcard and is able to eavesdrop the message exchanged in each session. However, the MA cannot trace a legal user U_i because all exchanged messages are updated every session, and also $\{Q_i, W_i, MID_i\}$ messages in the proposed SLUA-WSN update with $\{Q_i^{new}, W_i^{new}, MID_i^{new}\}$. Moreover, the MA cannot obtain the real ID_i of U_i because it is masked with XOR and hash functions. Thus, SLUA-WSN provides anonymity and untraceability because the MA cannot retrieve ID_i without knowing a secret parameter X_i and a random nonce R_u.

7.1.6. Mutual Authentication

In SLUA-WSN, each entity performs mutual authentication successfully. Upon getting the authentication request messages $\{M_1, MID_i, CID_i, M_{UG}\}$ from the U_i, the GWN verifies $M_{UG}^* \overset{?}{=} M_{UG}$. If the condition is correct, the GWN authenticates the U_i. After getting the messages $\{M_2, MID_i, M_{GS}, T_2\}$ from the GWN, the S_j checks $M_{GS}^* \overset{?}{=} M_{GS}$. If it is valid, the S_j authenticates the GWN. After receiving the messages $\{M_3, M_{SG}, M_{SU}, T_3\}$ from the S_j, the GWN verifies $M_{SG}^* \overset{?}{=} M_{SG}$. If the condition is correct, the GWN authenticates the S_j. After obtaining the response messages $\{M_4, M_{SU}, M_{GU}, T_4\}$ from the GWN, the U_i authenticates the GWN. As a result, the U_i, the S_j and the GWN are mutually authenticated because the MA cannot generate exchanged messages $\{M_{UG}, M_{GS}, M_{SG}, M_{SU}\}$ successfully.

7.2. Security Properties

We present the security properties of SLUA-WSN compared to those of the existing schemes [15,37–41]. Table 2 tabulates the security and functionality features of the proposed SLUA-WSN and other existing schemes. According to Table 2, previous schemes [15,37–41] suffer from various attacks, and also their schemes cannot ensure anonymity, untraceability, and mutual authentication. In contrast, SLUA-WSN ensures mutual authentication, anonymity, and untraceability and prevents various attacks. Thus, the proposed SLUA-WSN offers superior security and more functionality features compared with existing schemes.

Table 2. Security property comparison.

Security Properties	Wu et al. [37]	Wang et al. [38]	Li et al. [39]	Li et al. [40]	Lu et al. [41]	Mo and Chen [15]	Ours
Three-factor security	×	○	×	○	×	○	○
Masquerade attack	×	○	×	×	×	×	○
Replay attack	×	○	×	×	○	○	○
Privileged insider attack	○	×	○	×	○	○	○
Sensor node capture attack	○	○	○	○	○	○	○
Man-in-the-middle attack	○	○	×	×	○	○	○
User anonymity	○	○	○	○	○	×	○
Untraceability	○	○	○	○	○	×	○
Mutual authentication	○	○	○	○	○	×	○

○: it supports security properties; ×: it does not support security properties;

7.3. *Formal Security Analysis Using Ban Logic*

We perform the BAN logic to demonstrate the mutual authentication of SLUA-WSN. We present notations utilized for BAN logic in Table 3.

Table 3. Notations used for BAN logic.

Notation	Description
$N| \equiv M$	N **believes** M
$\#M$	M is updated and **fresh**
$N \lhd M$	N **sees** M
$N| \sim M$	N once **said** M
$N \Rightarrow M$	N **controls** that M
$< M >_W$	M is **combined** with W
$\{M\}_K$	M is **encrypted** utilizing symmetric key K
$N \overset{K}{\leftrightarrow} P$	N and P share a **shared secret key** K
SK	**Session key** used in communication session

7.3.1. Rules of Ban Logic

In the following, the rules of BAN logic are summarized.

1. Message meaning rule:

$$\frac{N \Big| \equiv N \overset{K}{\leftrightarrow} P, \quad N \lhd \{M\}_K}{N \ |\equiv P \ | \sim M}$$

2. Nonce verification rule:

$$\frac{N \mid\equiv \#(M), \quad N \mid \equiv P \Big| \sim M}{N \mid\equiv P \mid \equiv M}$$

3. Jurisdiction rule:

$$\frac{N \mid\equiv P \mid \Longrightarrow M, \quad N \mid\equiv P \mid \equiv M}{N \Big| \equiv M}$$

4. Freshness rule:

$$\frac{N \Big| \equiv \#(M)}{N \Big| \equiv \#\,(M, W)}$$

5. Belief rule:

$$\frac{N \Big| \equiv (M, W)}{N \Big| \equiv M}$$

7.3.2. Goals

We define the following security goals to prove that the proposed SLUA-WSN is capable of performing secure mutual authentication.

Goal 1: $U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$

Goal 2: $S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$

Goal 3: $U_i \mid\equiv S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$

Goal 4: $S_j \mid\equiv U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$

7.3.3. Idealized Forms

The idealized form messages of SLUA-WSN are as below.

Msg_1**:** $U_i \rightarrow GWN$: $(ID_i, MID_i, R_u, T_1)_{X_i}$

Msg_2**:** $GWN \rightarrow S_j$: $(MID_i, SID_j, R_u, R_g, T_2)_{X_j}$

Msg_3**:** $S_j \rightarrow GWN$: $(MID_i, SID_j, R_u, R_s, T_3)_{X_j}$

Msg_4**:** $GWN \rightarrow U_i$: $(ID_{MU}, R_g, R_s, T_4)_{X_i}$

7.3.4. Assumptions

In the following, the assumptions used in BAN logic are summarized.

A_1**:** $GWN \mid\equiv \#(T_1)$

A_2**:** $GWN \mid\equiv \#(T_3)$

A_3: $S_j \mid\equiv \#(T_2)$

A_4: $U_i \mid\equiv \#(T_4)$

A_5: $GWN \mid\equiv (GWN \overset{X_j}{\longleftrightarrow} S_j)$

A_6: $S_j \mid\equiv (GWN \overset{X_j}{\longleftrightarrow} S_j)$

A_7: $U_i \mid\equiv (U_i \overset{X_i}{\longleftrightarrow} GWN)$

A_8: $GWN \mid\equiv (U_i \overset{X_i}{\longleftrightarrow} GWN)$

A_9: $U_i \mid\equiv S_j \Rightarrow (U_i \overset{SK}{\longleftrightarrow} S_j)$

A_{10}: $S_j \mid\equiv U_i \Rightarrow (U_i \overset{SK}{\longleftrightarrow} S_j)$

7.3.5. Proof Using Ban Logic

The BAN logic proof then proceeds as below.

Step 1: According to Msg_1, we could get the following,

$$(S_1) : GWN \lhd (ID_i, MID_i, R_u, T_1)_{X_i}$$

Step 2: Using S_1 and A_8 with "message meaning rule", the following is obtained,

$$(S_2) : GWN \mid\equiv MU \mid\sim (ID_i, MID_i, R_u, T_1)_{X_i}$$

Step 3: Using S_2 and A_1 with "freshness rule", the following is obtained,

$$(S_3) : GWN \mid\equiv \#(ID_i, MID_i, R_u, T_1)_{X_i}$$

Step 4: From S_2 and S_3 with "nonce verification rule", we could get

$$(S_4) : GWN \mid\equiv U_i \mid\equiv (ID_i, MID_i, R_u, T_1)_{X_i}$$

Step 5: According to Msg_2, we could get

$$(S_5) : S_j \lhd (MID_i, SID_j, R_u, R_g, T_2)_{X_j}$$

Step 6: Using the S_5 and A_6 with "message meaning rule", the following is obtained,

$$(S_6) : S_j \mid\equiv GWN \mid\sim (MID_i, SID_j, R_u, R_g, T_2)_{X_j}$$

Step 7: Now, using S_6 and A_3 with "freshness rule", we could get

$$(S_7) : S_j \mid\equiv \#(MID_i, SID_j, R_u, R_g, T_2)_{X_j}$$

Step 8: Utilizing S_6 and S_7 with "nonce verification rule", the following is obtained,

$$(S_8) : S_j \mid\equiv GWN \mid\equiv (MID_i, SID_j, R_u, R_g, T_2)_{X_j}$$

Step 9: According to Msg_3, we could get the following,

$$(S_9) : GWN \lhd (MID_i, SID_j, R_u, R_s, T_3)_{X_j}$$

Step 10: Using S_9 and A_5 with "message meaning rule", the following is obtained,

$$(S_{10}) : GWN \mid\equiv S_j \mid\sim (MID_i, SID_j, R_u, R_s, T_3)_{X_j}$$

Step 11: Using S_{10} and A_2 with "freshness rule", the following is obtained,

$$(S_{11}) : GWN \mid\equiv \#(MID_i, SID_j, R_u, R_s, T_3)_{X_j}$$

Step 12: From S_{10} and S_{11} with "nonce verification rule", we could get

$$(S_{12}) : GWN \mid\equiv U_i \mid\equiv (MID_i, SID_j, R_u, R_s, T_3)_{X_j}$$

Step 13: According to Msg_4, we could get the following,

$$(S_{13}) : U_i \lhd (ID_{MU}, R_g, R_s, T_4)_{X_i}$$

Step 14: Using S_{13} and A_7 with "message meaning rule", the following is obtained,

$$(S_{14}) : U_i \mid\equiv GWN \mid\sim (ID_{MU}, R_g, R_s, T_4)_{X_i}$$

Step 15: Using S_{14} and A_4 with "freshness rule", the following is obtained,

$$(S_{15}) : U_i \mid\equiv \#(ID_{MU}, R_g, R_s, T_4)_{X_i}$$

Step 16: From S_{14} and S_{15} with "nonce verification rule", we could get

$$(S_{16}) : U_i \mid\equiv GWN \mid\equiv (ID_{MU}, R_g, R_s, T_4)_{X_i}$$

Step 17: Because $SK = h(R_u||R_s)$, according to S_{12} and S_{16}, the following is obtained,

$$(S_{17}) : U_i \mid\equiv S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j) \qquad \textbf{(Goal 3)}$$

Step 18: Because $SK = h(R_u||R_s)$, according to S_4 and S_8, we could get

$$(S_{18}) : S_j \mid\equiv U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j) \qquad \textbf{(Goal 4)}$$

Step 19: From A_9 and S_{17}, the following is obtained,

$$(S_{19}) : U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j) \qquad \textbf{(Goal 1)}$$

Step 20: Using A_{10} and S_{18}, the following is obtained,

$$(S_{20}) : S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j) \qquad \textbf{(Goal 2)}$$

According to Goals 1–4, we prove that the proposed SLUA-WSN ensures secure mutual authentication among U_i, GWN, and S_j.

7.4. Formal Security Analysis Using Ror Model

We perform the ROR model [17] to evaluate the session key (SK) security of SLUA-WSN from the malicious attacker MA. Initially, we introduce the ROR model [17] before performing the analysis of SK security for SLUA-WSN.

In the ROR model, the malicious attacker MA interacts with the P^t_{MA}, the t^{th} instance of the executing participant. Furthermore, there are three participants—the user $P^{t_1}_{U_i}$, gateway $P^{t_2}_{GWN}$, and sensor $P^{t_3}_{S_j}$—where $P^{t_1}_{U_i}$, $P^{t_2}_{GWN}$, and $P^{t_3}_{S_j}$ are instances t^{th}_1 of U_i, t^{th}_2 of GWN, and t^{th}_3 of S_j, respectively. In Table 4, we define various queries for ROR model to evaluate security analysis such as $Execute$, $CorruptSC$, $Reveal$, $Send$, and $Test$. Furthermore, an one-way hash function $h(\cdot)$ is modeled as a random oracle $Hash$. We utilize Zipf's law [48] to evaluate SK security of SLUA-WSN.

Table 4. Queries of the Real-or-Random (ROR) model.

Query	**Description**
$Execute(P^{t_1}_{U_i}, P^{t_2}_{GWN}, P^{t_3}_{S_j})$	$Execute$ denotes that MA performs the passive attack by eavesdropping transmitted messages between legitimate participants over an insecure channel.
$CorruptSC(P^{t_1}_{U_i})$	$CorruptSC$ is modeled that the smartcard stolen attack, in which the MA can extract the secret credentials stored in the smartcard.
$Send(P^t, M)$	Using this query, the MA can transmit a message M to the instance P^t and also can receive accordingly.
$Test(P^t)$	$Test$ corresponds to the semantic security of the SK between U_i and S_j following the indistinguishability style in the ROR model [17]. In this query, an unbiased coin c is flipped prior to the starting of the experiment. If the MA performs $Test$ query and the corresponding SK is fresh, and then P^t returns SK when $c = 1$ after running $Test$ query, SK is new or a random number when $c = 0$; otherwise, it delivers a null value ($\perp$).
$Reveal(P^t)$	Using this query, the MA reveals the current SK generated by its partner to an adversary MA.

Theorem 1. *If Adv_{MA} denotes the advantage function of the MA in violating SK security of SLUA-WSN. After that, we can derive the following.*

$$Adv_{MA} \leq \frac{q_h^2}{|Hash|} + 2\{C \cdot q^s_{send}, \frac{q_s}{2^{l_b}}\}$$

where q_h, $|Hash|$, and q_{send} are the number of Hash, the range space of Hash, and the number of Send queries, respectively. Furthermore, C, l_b, and s are parameters used in Zipf's laws [48].

Proof 1. We define the following four games, namely, G_i ($i \in [0, 3]$). We indicate that $Succ_i$ is the probability of MA winning the G_i. All G_i are described in detail as shown below.

- **Game** G_0: The first game G_0 is considered as an passive attack executed from the MA in the proposed protocol P, as the bit C is guessed randomly at the beginning of G_0. According to this game, the following is obtained.

$$Adv_{MA} = |2 \cdot Pr[Succ_0] - 1| \quad (1)$$

- **Game** G_1: This G_1 considers the scenario where MA simulates the eavesdropping attack in which the transmitted messages are intercepted during the authentication process using the $Execute$ query. After eavesdropping transmitted messages, the MA performs the $Reveal$ and $Test$ queries to verify whether it is the SK or a random number. The MA needs the secret parameters, such as R_u, R_s, X_i, and X_j, to derive $SK = h(R_u||R_s)$. Thus, the MA does not at all help in increasing the

G_1's winning probability by eavesdropping on the transmitted messages. According to this game, the following is obtained.

$$Pr[Succ_1] = Pr[Succ_0] \tag{2}$$

- **Game** G_2: G_2 is modeled as an active attack, where the simulations of the *Send* and *Hash* oracles are included. In G_2, the MA can eavesdrop all exchanged messages $\{M_1, MID_i, CID_i, M_{UG}, T_1\}$, $\{M_2, MID_i, M_{GS}, T_2\}$, $\{M_3, M_{SG}, M_{SU}, T_3\}$, and $\{M_4, M_{SU}, M_{GU}, T_4\}$ during the authentication and key agreement process. However, all exchanged messages are safeguarded using the hash function $h(\cdot)$. Furthermore, the random numbers R_u and R_s are not derived from the intercepted exchanged messages because the random numbers are protected by hash function $h(\cdot)$. By applying the birthday paradox [49], we can derive the following.

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{q_h^2}{2|Hash|} \tag{3}$$

- **Game** G_3: G_3 is simulated using $CorruptSC$ query. In this game, the MA is able to extract the secret credentials $\{Q_i, W_i, MID_i\}$ from a smartcard's memory using the power analysis attack. Generally, a user utilizes the low-entropy password. Using SC's stored secret credentials $\{Q_i, W_i, MID_i\}$, the MA may try to extract the password PW_i by performing a password guessing attack. However, in the proposed protocol, the MA cannot obtain password PW_i of the legitimate user correctly through the *Send* query without GWN's master key K_{GWN} and secret parameter X_i. Furthermore, the probability of guessing the biometric secret key b_i of l_b bits by the MA is approximately $\frac{1}{2^{l_b}}$. Thus, the G_2 and G_3 are indistinguishable if biometric/password guessing attacks are not present. Consequently, by applying Zipf's law [48], the following is obtained.

$$|Pr[Succ_3] - Pr[Succ_2]| \leq max\{C \cdot q_{send}^s, \frac{q_s}{2^{l_b}}\} \tag{4}$$

When all the games are executed, the MA should guess the correct bit c. Consequently, we can obtain the following result.

$$Pr[Succ_3] = \frac{1}{2} \tag{5}$$

By applying Equations (1), (2), and (5), the following result is obtained.

$$\begin{aligned}\frac{1}{2}Adv_{MA} &= |Pr[Succ_0] - \frac{1}{2}| \\ &= |Pr[Succ_1] - \frac{1}{2}| \\ &= |Pr[Succ_1] - Pr[Succ_3]|\end{aligned} \tag{6}$$

By applying Equations (4)–(6), the following result is obtained, utilizing the triangular inequality.

$$\begin{aligned}\frac{1}{2}Adv_{U_A} &= |Pr[Succ_1] - Pr[Succ_3]| \\ &\leq |Pr[Succ_1] - Pr[Succ_2]| \\ &\quad + |Pr[Succ_2] - Pr[Succ_3]| \\ &\leq \frac{q_h^2}{2|Hash|} + max\{C \cdot q_{send}^s, \frac{q_s}{2^{l_b}}\}\end{aligned} \tag{7}$$

As a result, multiplying both sides of Equation (7) by a factor of two, the following result is obtained.

$$Adv_{MA} \leq \frac{q_h^2}{|Hash|} + 2max\{C \cdot q_{send}^s, \frac{q_s}{2^{l_b}}\}$$

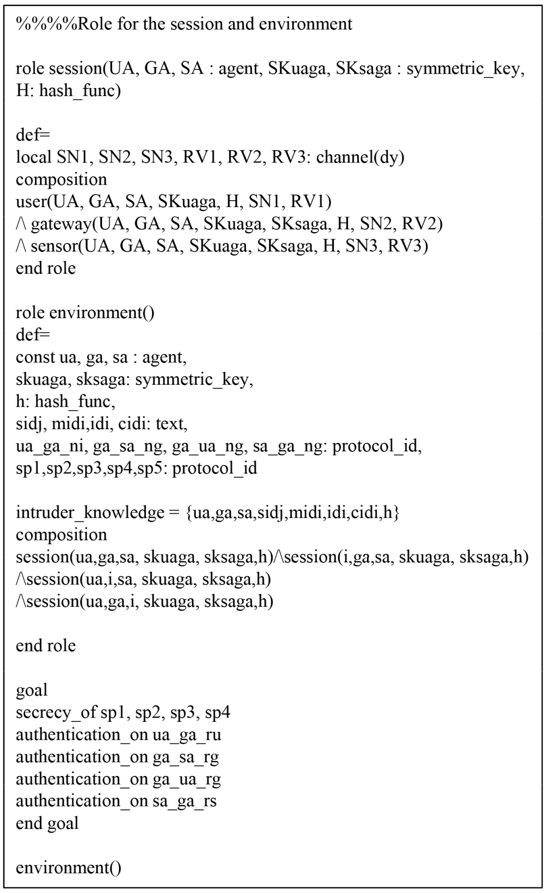
□

7.5. AVISPA Simulation

We perform the AVISPA simulation tool [18,19] to prove the security of SLUA-WSN against MITM and replay attacks. To perform the AVISPA simulation, the environment and session of the protocol must be implemented utilizing the High-Level Protocols Specification Language (HLPSL) [50].

7.5.1. HLPSL Specification

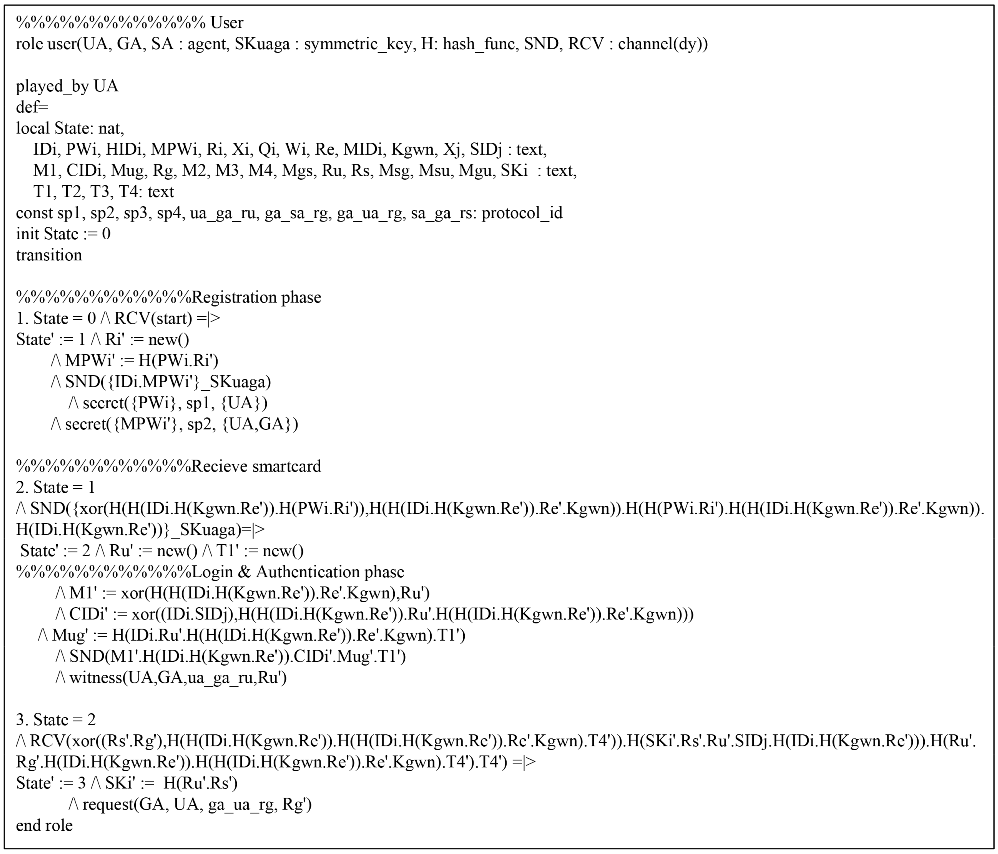
Referring to HLPSL, we consider three roles: the U_i, the GWN, and the S_j. We present the environment and session using HLPSL in Figure 7, which consists of the security goals.



```
%%%%Role for the session and environment

role session(UA, GA, SA : agent, SKuaga, SKsaga : symmetric_key,
H: hash_func)

def=
local SN1, SN2, SN3, RV1, RV2, RV3: channel(dy)
composition
user(UA, GA, SA, SKuaga, H, SN1, RV1)
/\ gateway(UA, GA, SA, SKuaga, SKsaga, H, SN2, RV2)
/\ sensor(UA, GA, SA, SKuaga, SKsaga, H, SN3, RV3)
end role

role environment()
def=
const ua, ga, sa : agent,
skuaga, sksaga: symmetric_key,
h: hash_func,
sidj, midi,idi, cidi: text,
ua_ga_ni, ga_sa_ng, ga_ua_ng, sa_ga_ng: protocol_id,
sp1,sp2,sp3,sp4,sp5: protocol_id

intruder_knowledge = {ua,ga,sa,sidj,midi,idi,cidi,h}
composition
session(ua,ga,sa, skuaga, sksaga,h)/\session(i,ga,sa, skuaga, sksaga,h)
/\session(ua,i,sa, skuaga, sksaga,h)
/\session(ua,ga,i, skuaga, sksaga,h)

end role

goal
secrecy_of sp1, sp2, sp3, sp4
authentication_on ua_ga_ru
authentication_on ga_sa_rg
authentication_on ga_ua_rg
authentication_on sa_ga_rs
end goal

environment()
```

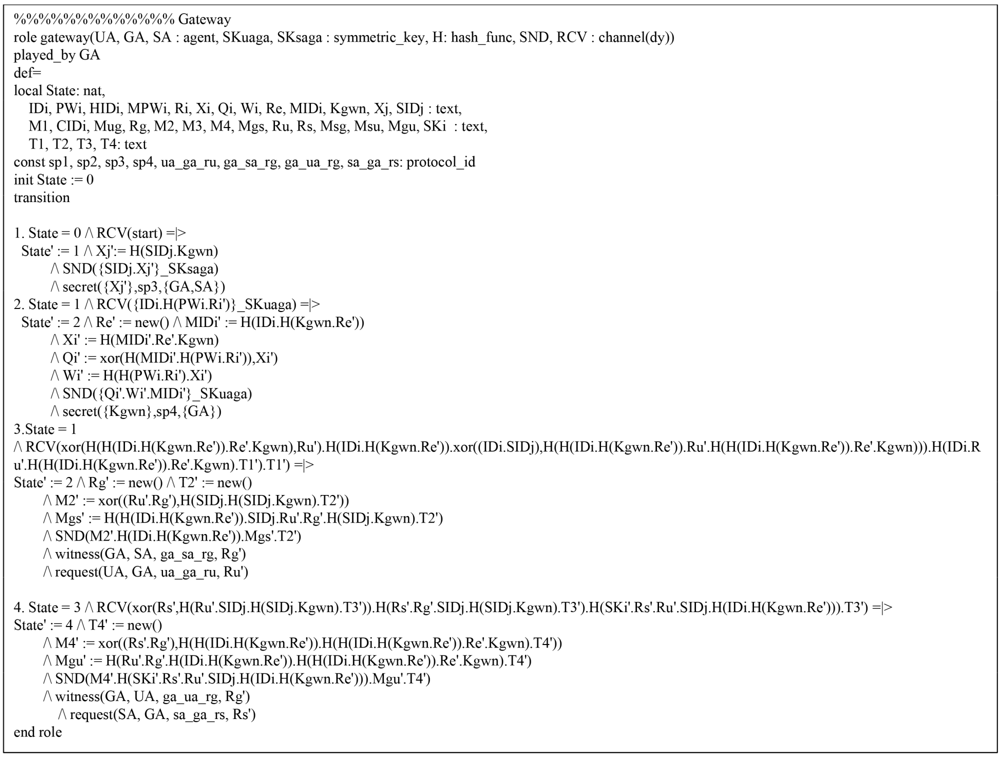
Figure 7. High-Level Protocols Specification Language (HLPSL) syntax for session and environment.

In Figure 8, the U_i initially receives the message and updates the state value from 1 to 2. After that, U_i transmits the registration request message $\{ID_i, MPW_i\}$ to GWN over a secure channel. Then, U_i receives the $\{smartcard\}$ from GWN and U_i changes the state value from 1 to 2. In the authentication process, the U_i should send an authentication request message $\{M_1, MID_i, CID_i, M_{UG}, T_1\}$ to GWN over a public channel. Thus, the U_i declares $witness(UA, GA, ua\_ga\_ru, RU')$ from the GWN, and then changes the state value from 2 to 3. Then, U_i receives the authentication response messages $\{M_4, M_{SU}, M_{GU}, T_4\}$ from the GWN. Finally, U_i checks $M^*_{GU} \overset{?}{=} M_{GU}$ and $M^*_{SU} \overset{?}{=} M_{SU}$. If it is correct, the U_i, GWN, and S_j are mutually authenticated successfully. In addition, the HLPSL specification roles of GWN and S_j are similarly defined. Figures 9 and 10 show the role specification of the GWN and S_j.



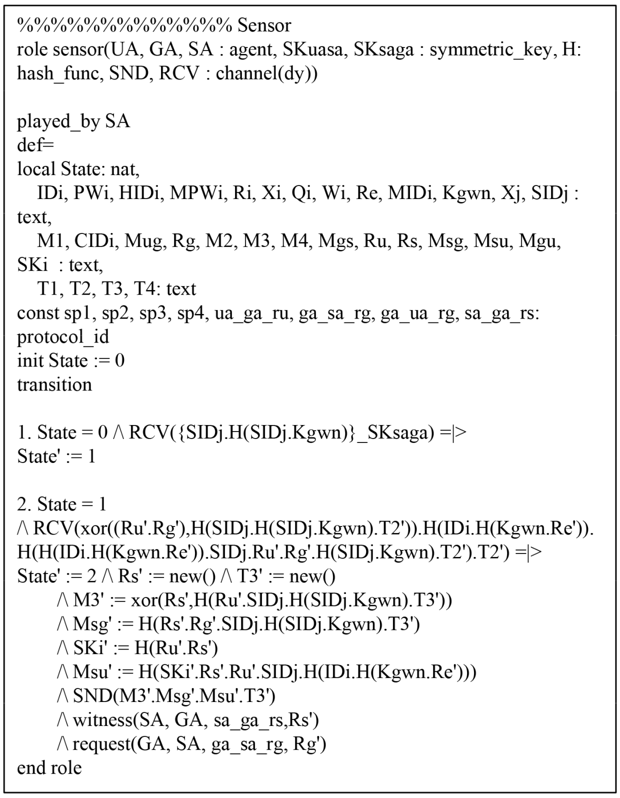
```
%%%%%%%%%%%%% User
role user(UA, GA, SA : agent, SKuaga : symmetric_key, H: hash_func, SND, RCV : channel(dy))

played_by UA
def=
local State: nat,
   IDi, PWi, HIDi, MPWi, Ri, Xi, Qi, Wi, Re, MIDi, Kgwn, Xj, SIDj : text,
   M1, CIDi, Mug, Rg, M2, M3, M4, Mgs, Ru, Rs, Msg, Msu, Mgu, SKi  : text,
   T1, T2, T3, T4: text
const sp1, sp2, sp3, sp4, ua_ga_ru, ga_sa_rg, ga_ua_rg, sa_ga_rs: protocol_id
init State := 0
transition

%%%%%%%%%%%%Registration phase
1. State = 0 /\ RCV(start) =|>
State' := 1 /\ Ri' := new()
      /\ MPWi' := H(PWi.Ri')
      /\ SND({IDi.MPWi'}_SKuaga)
         /\ secret({PWi}, sp1, {UA})
      /\ secret({MPWi'}, sp2, {UA,GA})

%%%%%%%%%%%%Recieve smartcard
2. State = 1
/\ SND({xor(H(H(IDi.H(Kgwn.Re')).H(PWi.Ri')),H(H(IDi.H(Kgwn.Re')).Re'.Kgwn)).H(H(PWi.Ri').H(H(IDi.H(Kgwn.Re')).Re'.Kgwn)).
H(IDi.H(Kgwn.Re'))}_SKuaga)=|>
 State' := 2 /\ Ru' := new() /\ T1' := new()
%%%%%%%%%%%%%Login & Authentication phase
       /\ M1' := xor(H(H(IDi.H(Kgwn.Re')).Re'.Kgwn),Ru')
       /\ CIDi' := xor((IDi.SIDj),H(H(IDi.H(Kgwn.Re')).Ru'.H(H(IDi.H(Kgwn.Re')).Re'.Kgwn)))
     /\ Mug' := H(IDi.Ru'.H(H(IDi.H(Kgwn.Re')).Re'.Kgwn).T1')
       /\ SND(M1'.H(IDi.H(Kgwn.Re')).CIDi'.Mug'.T1')
       /\ witness(UA,GA,ua_ga_ru,Ru')

3. State = 2
/\ RCV(xor((Rs'.Rg'),H(H(IDi.H(Kgwn.Re')).H(H(IDi.H(Kgwn.Re')).Re'.Kgwn).T4')).H(SKi'.Rs'.Ru'.SIDj.H(IDi.H(Kgwn.Re'))).H(Ru'.
Rg'.H(IDi.H(Kgwn.Re')).H(H(IDi.H(Kgwn.Re')).Re'.Kgwn).T4').T4') =|>
State' := 3 /\ SKi' :=  H(Ru'.Rs')
         /\ request(GA, UA, ga_ua_rg, Rg')
end role
```

Figure 8. HLPSL syntax for U_i.



```
%%%%%%%%%%%%% Gateway
role gateway(UA, GA, SA : agent, SKuaga, SKsaga : symmetric_key, H: hash_func, SND, RCV : channel(dy))
played_by GA
def=
local State: nat,
   IDi, PWi, HIDi, MPWi, Ri, Xi, Qi, Wi, Re, MIDi, Kgwn, Xj, SIDj : text,
   M1, CIDi, Mug, Rg, M2, M3, M4, Mgs, Ru, Rs, Msg, Msu, Mgu, SKi  : text,
   T1, T2, T3, T4: text
const sp1, sp2, sp3, sp4, ua_ga_ru, ga_sa_rg, ga_ua_rg, sa_ga_rs: protocol_id
init State := 0
transition

1. State = 0 /\ RCV(start) =|>
 State' := 1 /\ Xj':= H(SIDj.Kgwn)
       /\ SND({SIDj.Xj'}_SKsaga)
       /\ secret({Xj'},sp3,{GA,SA})
2. State = 1 /\ RCV({IDi.H(PWi.Ri')}_SKuaga) =|>
 State' := 2 /\ Re' := new() /\ MIDi' := H(IDi.H(Kgwn.Re'))
       /\ Xi' := H(MIDi'.Re'.Kgwn)
       /\ Qi' := xor(H(MIDi'.H(PWi.Ri')),Xi')
       /\ Wi' := H(H(PWi.Ri').Xi')
       /\ SND({Qi'.Wi'.MIDi'}_SKuaga)
       /\ secret({Kgwn},sp4,{GA})
3.State = 1
/\ RCV(xor(H(H(IDi.H(Kgwn.Re')).Re'.Kgwn),Ru').H(IDi.H(Kgwn.Re')).xor((IDi.SIDj),H(H(IDi.H(Kgwn.Re')).Ru'.H(H(IDi.H(Kgwn.Re')).Re'.Kgwn))).H(IDi.R
u'.H(H(IDi.H(Kgwn.Re')).Re'.Kgwn).T1').T1') =|>
State' := 2 /\ Rg' := new() /\ T2' := new()
      /\ M2' := xor((Ru'.Rg'),H(SIDj.H(SIDj.Kgwn).T2'))
      /\ Mgs' := H(H(IDi.H(Kgwn.Re')).SIDj.Ru'.Rg'.H(SIDj.Kgwn).T2')
      /\ SND(M2'.H(IDi.H(Kgwn.Re')).Mgs'.T2')
      /\ witness(GA, SA, ga_sa_rg, Rg')
      /\ request(UA, GA, ua_ga_ru, Ru')

4. State = 3 /\ RCV(xor(Rs',H(Ru'.SIDj.H(SIDj.Kgwn).T3')).H(Rs'.Rg'.SIDj.H(SIDj.Kgwn).T3').H(SKi'.Rs'.Ru'.SIDj.H(IDi.H(Kgwn.Re'))).T3') =|>
State' := 4 /\ T4' := new()
      /\ M4' := xor((Rs'.Rg'),H(H(IDi.H(Kgwn.Re')).H(H(IDi.H(Kgwn.Re')).Re'.Kgwn).T4'))
      /\ Mgu' := H(Ru'.Rg'.H(IDi.H(Kgwn.Re')).H(H(IDi.H(Kgwn.Re')).Re'.Kgwn).T4')
      /\ SND(M4'.H(SKi'.Rs'.Ru'.SIDj.H(IDi.H(Kgwn.Re'))).Mgu'.T4')
      /\ witness(GA, UA, ga_ua_rg, Rg')
         /\ request(SA, GA, sa_ga_rs, Rs')
end role
```

Figure 9. HLPSL syntax for GWN.

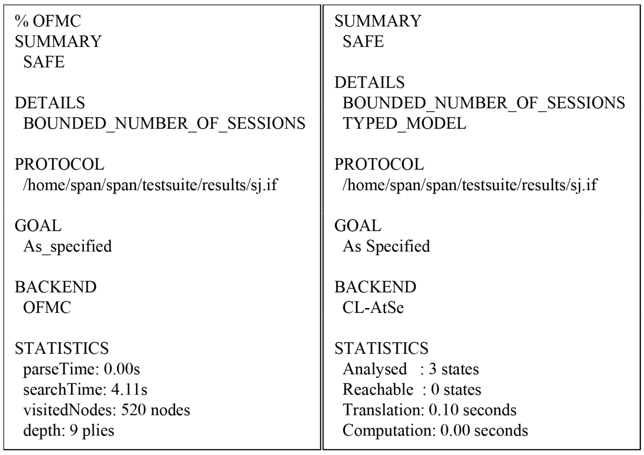


```
%%%%%%%%%%%%% Sensor
role sensor(UA, GA, SA : agent, SKuasa, SKsaga : symmetric_key, H:
hash_func, SND, RCV : channel(dy))

played_by SA
def=
local State: nat,
   IDi, PWi, HIDi, MPWi, Ri, Xi, Qi, Wi, Re, MIDi, Kgwn, Xj, SIDj :
text,
   M1, CIDi, Mug, Rg, M2, M3, M4, Mgs, Ru, Rs, Msg, Msu, Mgu,
SKi  : text,
   T1, T2, T3, T4: text
const sp1, sp2, sp3, sp4, ua_ga_ru, ga_sa_rg, ga_ua_rg, sa_ga_rs:
protocol_id
init State := 0
transition

1. State = 0 /\ RCV({SIDj.H(SIDj.Kgwn)}_SKsaga) =|>
State' := 1

2. State = 1
/\ RCV(xor((Ru'.Rg'),H(SIDj.H(SIDj.Kgwn).T2')).H(IDi.H(Kgwn.Re')).
H(H(IDi.H(Kgwn.Re')).SIDj.Ru'.Rg'.H(SIDj.Kgwn).T2').T2') =|>
State' := 2 /\ Rs' := new() /\ T3' := new()
        /\ M3' := xor(Rs',H(Ru'.SIDj.H(SIDj.Kgwn).T3'))
        /\ Msg' := H(Rs'.Rg'.SIDj.H(SIDj.Kgwn).T3')
        /\ SKi' := H(Ru'.Rs')
        /\ Msu' := H(SKi'.Rs'.Ru'.SIDj.H(IDi.H(Kgwn.Re')))
        /\ SND(M3'.Msg'.Msu'.T3')
        /\ witness(SA, GA, sa_ga_rs,Rs')
        /\ request(GA, SA, ga_sa_rg, Rg')
end role
```

Figure 10. HLPSL syntax for S_j.

7.5.2. AVISPA Simulation Result

We present the AVISPA simulation result to demonstrate the security of the SLUA-WSN utilizing On-the-Fly Model Checker (OFMC) and Constraint-Logic-based ATtack SEarcher (CL-AtSe) back-ends. The OFMC and CL-AtSe back-ends verify whether a legitimate entity is able to execute the protocol by searching for a passive attacker. In addition, CL-AtSe and OFMC back-ends check that the SLUA-WSN is secure against the replay and MITM attacks based on the DY model. According to Figure 11, the proposed SLUA-WSN is secure against MITM and replay attacks. Moreover, the result of OFMC validation shows that the search time was 4.11 s for visiting 520 nodes, and the result of the CL-AtSe validation analyzed three states and the translation time was 0.10 s. We provide similar AVISPA simulation results as adopted in [51–55].



```
% OFMC
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS

PROTOCOL
  /home/span/span/testsuite/results/sj.if

GOAL
  As_specified

BACKEND
  OFMC

STATISTICS
  parseTime: 0.00s
  searchTime: 4.11s
  visitedNodes: 520 nodes
  depth: 9 plies
```

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/sj.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS
  Analysed   : 3 states
  Reachable  : 0 states
  Translation: 0.10 seconds
  Computation: 0.00 seconds
```

Figure 11. AVISPA simulation results using On-the-Fly Model Checker (OFMC) and Constraint-Logic-based ATtack SEarcher (CL-AtSe).

8. Performance Analysis

We evaluate the performance of SLUA-WSN in terms of the computation, communication, and storage overheads. We also compare SLUA-WSN with other existing schemes [15,37–41].

8.1. Computation Overheads

This section compares the computation overhead associated with the SLUA-WSN to those of related schemes [15,37–41] during the authentication process. We analyzed utilizing the following parameters to evaluate the computation overhead. Referring to the work in [15], T_m, T_R, T_S, and T_h denote the execution time for point multiplication ($\approx$ 7.3529 ms), rep operation ($\approx$ 7.3529 ms), symmetric encryption/decryption ($\approx$ 0.1303 ms), and hash function ($\approx$ 0.0004 ms), respectively. The execution time of XOR operation is not included because it is negligible. In Table 5, we show the results of the computation overhead comparison. Consequently, SLUA-WSN provides a more efficient computation cost compared with the other existing schemes [15,37–41].

Table 5. Computation overheads comparison.

Schemes	User	Gateway	Sensor node	Total	Computation overhead
Wu et al. [37]	$11T_h + T_R + 2T_m$	$10T_h$	$3T_h + 2T_m$	$24T_h + T_R + 4T_m$	36.77 ms
Wang et al. [38]	$10T_h + T_R + 3T_m$	$13T_h + T_m$	$6T_h + 2T_m$	$29T_h + T_R + 6T_m$	51.48 ms
Li et al. [39]	$8T_h + T_R + 2T_m$	$9T_h + T_m$	$4T_h$	$21T_h + T_R + 3T_m$	29.42 ms
Li et al. [40]	$12T_h + 3T_m$	$8T_h + T_m$	$4T_h + 2T_m$	$24T_h + 6T_m$	44.13 ms
Lu et al. [41]	$7T_h + T_R + 3T_m + T_S$	$6T_h + T_m + T_S$	$2T_h + 2T_m + 2T_S$	$15T_h + T_R + 6T_m + 4T_S$	51.99 ms
Mo and Chen [15]	$12T_h + T_R + 2T_m$	$10T_h + T_S$	$5T_h + 2T_m + T_S$	$27T_h + T_R + 4T_m + 2T_S$	37.03 ms
Ours	$11T_h + T_R$	$11T_h$	$6T_h$	$28T_h + T_R$	7.36 ms

8.2. Communication Overheads

We compare the communication cost with the related schemes [15,37–41]. Referring to the work in [15], we assume that the hash function, a timestamp, an identity, a random nonce, and a prime p are 160 bits, 32 bits, 32 bits, 128 bits, and 160 bits, respectively. In addition, we consider that an ECC of 160 bits has a security level equivalent to that of the 1024-bit RSA [56]. The block size of plaintext/ciphertext for the AES algorithm is 128 bits [57]. In the authentication process of SLUA-WSN, the exchanged messages $\{M_1, MID_i, CID_i, M_{UG}, T_1\}$, $\{M_2, MID_i, M_{GS}, T_2\}$, $\{M_3, M_{SG}, M_{SU}, T_3\}$, and $\{M_4, M_{SU}, M_{GU}, T_4\}$ require (160 + 160 + 160 + 160 + 32 = 672 bits), (160 + 160 + 160 + 32 = 512 bits), (160 + 160 + 160 + 32 = 512 bits), and (160 + 160 + 160 + 32 = 512 bits), respectively. In Table 6, we present the results of the communication overhead comparison. Thus, SLUA-WSN has a more efficient communication cost compared with other related schemes [15,37–41].

Table 6. Communication overheads comparison.

Schemes	Communication Overhead	Number of Messages
Wu et al. [37]	3072 bits	4 messages
Wang et al. [38]	2368 bits	4 messages
Li et al. [39]	2496 bits	4 messages
Li et al. [40]	2880 bits	4 messages
Lu et al. [41]	2880 bits	3 messages
Mo and Chen [15]	3328 bits	4 messages
Ours	2208 bits	4 messages

8.3. Storage Overheads

We compare the storage costs with the related schemes [15,37–41]. We first define that the hash, identity, timestamp, random nonce, ECC algorithm, RSA algorithm, and AES algorithm are 20, 4, 4, 16, 20, 128, and 16 bytes, respectively, and the prime p in $E_p(a, b)$ is 20 bytes. In the proposed SLUA-WSN, stored messages $\{Q_i, W_i, MID_i\}$ and $\{r_g\}$ require (20 + 20 + 20 = 60 bytes) and (20 bytes), respectively. Although the storage costs of the proposed SLUA-WSN are somewhat higher than Mo and Chen's scheme [15], it provides better security and efficiency than the other related schemes [15,37–41]. Table 7 shows the analysis results of storage overhead compared to related schemes.

Table 7. Storage overheads comparison.

Schemes	Stored Message (Smart Card/mobile Device)	Stored Message (Gateway Node)
Wu et al. [37]	$B_1, B_2, P_{bi} \approx$ 56 bytes	$ID_i \approx$ 4 bytes
Wang et al. [38]	$A_i, B_i, n_0, Y, P \approx$ 100 bytes	$ID_i, r_i \approx$ 20 bytes
Li et al. [39]	$\alpha, \delta, A_i, B_i, X \approx$ 92 bytes	$ID_i \approx$ 4 bytes
Li et al. [40]	$A_i, B_i, E_i, X, f, n_0, r \approx$ 108 bytes	$ID_i, k_i \approx$ 20 bytes
Lu et al. [41]	$RPW_i, f_i, v_i \approx$ 56 bytes	$K_j \approx$ 20 bytes
Mo and Chen [15]	$RID_i, f_i, \tau \approx$ 56 bytes	$K_j \approx$ 20 bytes
Ours	$Q_i, W_i, MID_i \approx$ 60 bytes	$r_g \approx$ 20 bytes

9. Conclusions

In this paper, we proved that Mo and Chen's scheme suffers from various security flaws, such as session key exposure and masquerade attacks, and does not provide anonymity, untraceability, and authentication. We proposed a secure and lightweight user authentication protocol in WSN environments utilizing biometric and secret parameters to resolve the security drawbacks of Mo and Chen's protocol. SLUA-WSN prevents various attacks, including sensor node capture, masquerade, and privileged insider attacks. We demonstrated that the proposed SLUA-WSN ensures secure mutual authentication between U_i, GWN, and S_j by performing BAN logic. We also proved the security of SLUA-WSN by performing the formal security analysis such as the ROR model and AVISPA simulation. We compared the performance of SLUA-WSN in terms of computation, communication, and storage overheads with existing schemes. Consequently, the proposed SLUA-WSN provided a great improvement in terms of the security level compared with three-factor-based related schemes and also preserved the low computation and communication overheads using only hash and XOR operations. Therefore, the proposed SLUA-WSN provides superior security and efficiency than related schemes and is suitable for practical WSN environments.

Author Contributions: Conceptualization, S.Y. and Y.P.; methodology, S.Y. and Y.P.; software, S.Y. and Y.P.; validation, S.Y. and Y.P.; formal analysis, S.Y. and Y.P.; writing–original draft preparation, S.Y. and Y.P.; writing–review and editing, S.Y. and Y.P.; supervision, Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1I1A3058605, and in part by the BK21 Plus Project, Ministry of Education, South Korea, under Grant 21A20131600011.

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

References

1. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. A survey on Sensor networks. *IEEE Commun. Mag.* **2002**, *40*, 102–114.
2. Park, Y.H.; Lee, S.Y.; Kim, C.K.; Park, Y.H. Secure biometric-based authentication scheme with smart card revocation/reissue for wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 1–11.
3. Chen, C.M.; Wang, K.H.; Yeh, K.H.; Wu, T.Y. Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 3133–3142.
4. Chen, C.M.; Xiang, B.; Wu, T.Y.; Wang, K.H. An anonymous mutual authenticated key agreement scheme for wearable sensors in wireless body area networks. *Appl. Sci.* **2018**, *8*, 1074.
5. Rashid, B.; Rehmani, M.H. Applications of wireless sensor networks for urban areas: A survey. *J. Netw. Comput. Appl.* **2016**, *60*, 192–219.

6. Saia, R.; Carta, S.; Recupero, D.R.; Fenu, G. Internet of Entities (IoE): A blockchain-based distributed paradigm for data exchange between wireless-based devices. In Proceedings of the 8th International Conference on Sensor Networks, Setubal, Portugal, 26–27 February 2019; pp. 77–84.
7. Khan, S.; Pathan, A.S.K.; Alrajeh, N.A. *Wireless Sensor Networks: Current Status and Future Trends*; CRC Press: Boca Raton, FL, USA, 2020.
8. Wang, D.; Wang, P.; Wang, C. Efficient multi-factor user authentication protocol with forward secrecy for real-time data access in WSNs. *ACM Trans. Cyber-Phys. Syst.* **2020**, *4*, 1–26.
9. Jing, Q.; Vasilakos, A.V.; Wan, J.; Lu, J.; Qiu, D. Security of the Internet of Things: Perspectives and challenges. *Wirel. Netw.* **2014**, *20*, 2481–2501.
10. Pirbhulal, S.; Zhang, H.; Alahi, M.E.; Ghayvat, H.; Mukhopadhyay, S.C.; Zhang, Y.T.; Wu, W. A novel secure IoT-based smart home automation system using a wireless sensor network. *Sensors* **2017**, *17*, 69.
11. Wazid, M.; Das, A.K.; Odelu, V.; Kumar, N.; Susilo, W. Secure remote user authenticated key establishment protocol for smart home environment. *IEEE Trans. Dependable Secur. Comput.* **2020**, *17*, 391–406.
12. Park, Y.H.; Park, Y.H. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. *Sensors* **2016**, *16*, 2123.
13. Tomic, I.; McCann, J.A. A survey of potential security issues in existing wireless sensor network protocols. *IEEE Internet Things J.* **2017**, *4*, 1910–1923.
14. Xu, Z.; Xu, C.; Liang, W.; Xu, J.; Chen, H. A lightweight mutual authentication and key agreement scheme for medical internet of things. *IEEE Access* **2019**, *7*, 53922–53931.
15. Mo, J.; Chen, H. A lightweight secure user authentication and key agreement protocol for wireless sensor networks. *Secur. Commun. Netw.* **2019**, *2019*, 1–17.
16. Burrows M.; Abadi M.; Needham R. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36.
17. Abdalla M.; Fouque P.A.; Pointcheval, D. Password based authenticated key exchange in the three-party setting. In *Public Key Cryptography*; Springer: Les Diablerets, Switzerland, 2005; pp. 65–84.
18. AVISPA. Automated Validation of Internet Security Protocols and Applications. Available online: http://www.avispa-project.org/ (accessed on 8 February 2020).
19. SPAN: A Security Protocol Animator for AVISPA. Available online: Http://www.avispa-project.org/ (accessed on 8 February 2020).
20. Das, A.K.; Sharma, P.; Chatterjee, S.; Sing, J.K. A dynamic password-based user authentication scheme for hierarchical wireless sensor networks. *J. Netw. Comput. Appl.* **2012**, *35*, 1646–1656.
21. Farash, M.S.; Turkanovic, M.; Kumari, S.; Holbl, M. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. *Ad Hoc Netw.* **2016**, *36*, 152–176.
22. Tai, W.L.; Chang, Y.F.; Li, W.H. An IoT notion–based authentication and key agreement scheme ensuring user anonymity for heterogeneous ad hoc wireless sensor networks. *J. Inf. Secur. Appl.* **2017**, *34*, 133–141.
23. Renuka, K.; Kumar, S.; Kumari, S.; Chen, C.M. Cryptanalysis and improvement of a privacy-preserving three-factor authentication protocol for wireless sensor networks. *Sensors* **2019**, *19*, 4625.
24. Guo, H.; Gao, Y.; Xu, T.; Zhang, X.; Ye, J. A secure and efficient three-factor multi-gateway authentication protocol for wireless sensor networks. *Ad Hoc Netw.* **2019**, *95*, 101965.
25. Wu, F.; Li, X.; Xu, L.; Vijayakumar, P.; Kumar, N. A novel three-factor authentication protocol for wireless sensor networks with IoT notion. *IEEE Syst. J.* **2020**, 1–10, doi:10.1109/JSYST.2020.2981049.
26. Lamport, L. Password authentication with insecure communication. *Commun. ACM* **1981**, *24*, 770–772.
27. Das, M. L. Two-factor user authentication in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1086–1090.
28. Nyang, D.; Lee, M. K. Improvement of Das's two-factor authentication protocol in wireless sensor networks. *IACR Cryptol. ePrint Arch.* **2009**, *2009*, 631.
29. He, D.; Gao, Y.; Chen, S.; Chen, C.; Bu, J. An enhanced two-factor user authentication scheme in wireless sensor networks. *Ad Hoc Wirel. Netw.* **2010**, *10*, 361–371.
30. Kumar, P.; Lee, H.J. Cryptanalysis on two user authentication protocols using smart card for wireless sensor networks. In Proceedings of the Wireless Advanced, London, UK, 20–22 June 2011; pp. 241–245.
31. Das, A.K. A secure and efficient user anonymity-preserving three-factor authentication protocol for large-scale distributed wireless sensor networks. *Wirel. Pers. Commun.* **2015**, *82*, 1377–1404.

32. Yu, S.J.; Park, K.S.; Park, Y.H. A secure lightweight three-factor authentication scheme for IoT in cloud computing environment. *Sensors* **2019**, *19*, 3598.
33. Amin, R.; Islam, S.K.H.; Biswas, G.P.; Khan, M.K.; Leng, L.; Kumar, N. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. *Comput. Netw.* **2016**, *101*, 42–62.
34. Jiang, Q.; Zeadally, S.; Ma, J.; He, D. Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. *IEEE Access* **2017**, *5*, 3376–3392.
35. Soni, P.; Pal, A.K.; Islam, S.K.H. An improved three-factor authentication scheme for patient monitoring using WSN in remote health-care system. *Comput. Methods Programs Biomed.* **2019**, *182*, 105054.
36. Ali, Z.; Ghani, A.; Khan, I.; Chaudhry, S.A.; Islam, S.K.H.; Girl, D. A robust authentication and access control protocol for securing wireless healthcare sensor networks. *J. Inf. Secur. Appl.* **2020**, *52*, 102502.
37. Wu, F.; Xu, L.; Kumari, S.; Li X. An improved and provably secure three-factor user authentication scheme for wireless sensor networks. *Peer-to-Peer Netw. Appl.* **2018**, *11*, 1–20.
38. Wang, C.; Xu, G.; Sun, J. An enhanced three-factor user authentication scheme using elliptic curve cryptosystem for wireless sensor networks. *Sensors* **2017**, *17*, 2946.
39. Li, X.; Niu, J.; Kumari, S.; Wu, F.; Sangaiah, A.K.; Choo, K.K.R. A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments. *J. Netw. Comput. Appl.* **2018**, *103*, 194–204.
40. Li, X.; Peng, J.; Obaidat, M.S.; Wu, F.; Khan, K.K.; Chen, C. A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems. *IEEE Syst. J.* **2019**, *14*, 39–50.
41. Lu, Y.; Xu, G.; Li, L.; Yang, Y. Anonymous three-factor authenticated key agreement for wireless sensor networks. *Wirel. Netw.* **2019**, *25*, 1461–1475.
42. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 523–540.
43. Dolev, D.; Yao, A.C. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208.
44. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In *Advances in Cryptology*; Springer: Berlin, Germany, 1999; pp. 388–397.
45. Lee, J.Y.; Yu, S.J.; Park, K.S.; Park, Y.H.; Park, Y.H. Secure three-factor authentication protocol for multi-gateway IoT environments. *Sensors* **2019**, *19*, 2358.
46. Park, K.S.; Park, Y.H.; Das, A.K.; Yu, S.J.; Lee, J.Y.; Park, Y.H. A dynamic privacy-preserving key management protocol for V2G in social internet of things. *IEEE Access* **2019**, *7*, 76812–76832.
47. Xue, K.; Ma, C.; Hong, P.; Ding, R. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. *J. Netw. Comput. Appl.* **2013**, *36*, 316–323.
48. Wang D.; Cheng H.; Wang P.; Huang X.; Jian G. Zipf's law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791.
49. Boyko, V.; MacKenzie, P.; Patel, S. Provably secure password-authenticated key exchange using Diffie-Hellman. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Bruges, Belgium, 14–18 May 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 156–171.
50. Von Oheimb, D. The high-level protocol specification language HLPSL developed in the EU project AVISPA. In Proccedings of the APPSEM 2005 Workshop, Tallinn, Finland, 13–15 September 2005; pp. 1–2.
51. Yu, S.J.; Lee, J.Y.; Lee, K.K.; Park, K.S.; Park, Y.H. Secure authentication protocol for wireless sensor networks in vehicular communications. *Sensors* **2018**, *18*, 3191.
52. Challa, S.; Das, A.K.; Odelu, V.; Kumar, N.; Kumari, S.; Khan M.K.; Vasilakos, A.V. An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks. *Comput. Electr. Eng.* **2018**, *69*, 534–554.
53. Challa, S.; Das, A.K.; Gope, P.; Kumar, N.; Wu, F.; Vasilakos, A.V. Design and analysis of authenticated key agreement scheme in cloud-assisted cyber–physical systems. *Future Gener. Comput. Syst.* **2020**, *108*, 1267–1286.
54. Wazid, M.; Das, A.K.; Bhat, V.K.; Vasilakos, A.V. LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment. *J. Netw. Comput. Appl.* **2020**, *150*, 102496.

55. Banerjee, S.; Odelu, V.; Das, A.K.; Chattopadhyay, S.; Rodrigues, J.J.P.C.; Park, Y.H. Physically secure lightweight anonymous user authentication protocol for internet of things using physically unclonable funztions. *IEEE Access* **2019**, *7*, 85627–85644.
56. Rivest, R.L.; Hellman, M.E.; Anderson, J.C.; Lyons, J.W. Responses to NIST's proposal. *Commun. ACM* **1992**, *35*, 41–54.
57. Burrows, J.H. Secure hash standard. *Natl. Inst. Stand. Technol.* **1995**, *16*, 17–45.