Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5537018, 15 pages
https://doi.org/10.1155/2021/5537018

Research Article

A Provably Secure Three-Factor Authentication Protocol for Wireless Sensor Networks

Tsu-Yang Wu,[1] **Lei Yang**,[1] **Zhiyuan Lee**,[1] **Shu-Chuan Chu**,[1] **Saru Kumari**,[2] **and Sachin Kumar**[3]

[1]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[2]*Department of Mathematics, Chaudhary Charan Singh University, Meerut, Uttar Pradesh 250004, India*
[3]*Department of Computer Science and Engineering, Ajay Kumar Garg Engineering College, Ghaziabad 201009, India*

Correspondence should be addressed to Shu-Chuan Chu; scchu0803@gmail.com

Received 27 January 2021; Revised 12 March 2021; Accepted 1 April 2021; Published 16 April 2021

Academic Editor: Mattin Pirouz Nia

The wireless sensor network is a network composed of sensor nodes self-organizing through the application of wireless communication technology. The application of wireless sensor networks (WSNs) requires high security, but the transmission of sensitive data may be exposed to the adversary. Therefore, to guarantee the security of information transmission, researchers propose numerous security authentication protocols. Recently, Wu et al. proposed a new three-factor authentication protocol for WSNs. However, we find that their protocol cannot resist key compromise impersonation attacks and known session-specific temporary information attacks. Meanwhile, it also violates perfect forward secrecy and anonymity. To overcome the proposed attacks, this paper proposes an enhanced protocol in which the security is verified by the formal analysis and informal analysis, Burross-Abadii-Needham (BAN) logic, and ProVerif tools. The comparison of security and performance proves that our protocol has higher security and lower computational overhead.

1. Introduction

With the development of artificial intelligence technologies [1–3], the application of sensors has become more common, and the demand for high-end sensors is also increasing day by day. Sensors have developed from wired sensors to today's wireless sensors, and wireless sensors are the most common category in daily applications. The wireless sensor network [4, 5] is a self-organizing network formed by multiple functional nodes through wireless communication. These functional nodes include a large number of sensor nodes and gateway nodes. Sensor nodes perceive, collect, process, and transmit the information of the perceived object through the scope covered by the wireless sensor network.

Wireless body area network [6] usually installs sensors on clothes or attached to the human body and can also be implanted into the skin to monitor the user's physical activities and the state of body functions. The physical health data monitored by the sensors are sent to the cloud server for storage and analysis through the Internet of Things (IoTs). Users can view these data through the Internet and understand the physical condition, to achieve the purpose of early treatment of illnesses and reduce the number of deaths due to diseases. Wireless sensors are used in the growth of crops to monitor environmental factors such as humidity, temperature, and light that affect crop growth. The data monitored by the sensors are sent to the gateway node, which can send the data to the user to understand the growth status of crops, achieve the harvesting effect, and increase the income of farmers. The data collected by wireless sensor networks, whether used in military, medical, or other environments, is sensitive and private [7–13], so it is important to establish a secure authentication mechanism. Figure 1 shows a typical architecture in the wireless sensor network.

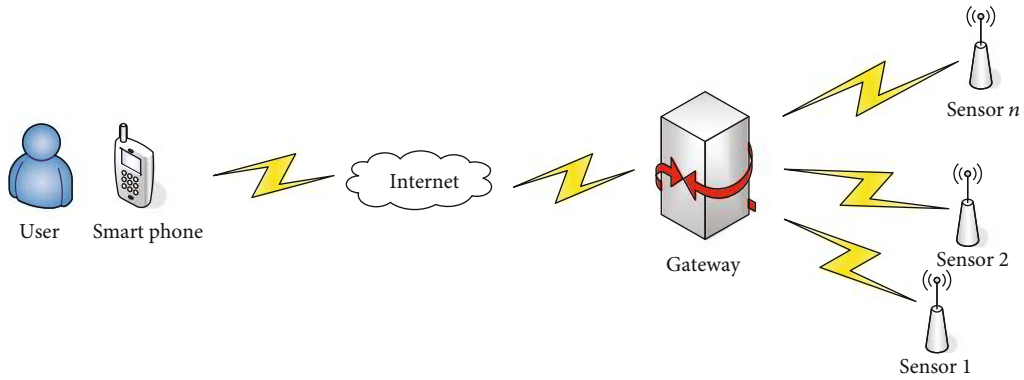

Figure 1: A typical wireless sensor network architecture.

In most authentication mechanisms of the wireless sensor networks, there are three components: user, sensor node, and gateway node. This paper will adopt such a structure, after the user logs in to the network, the data in the sensor are obtained through the gateway, and message authentication and key exchange are completed in this process. Since WSN is an open network, only using the password as a factor for encryption authentication will lead to a large number of vulnerabilities. In 2009, Das [14] proposed a protocol for encryption and authentication in wireless sensor network environments with a password and smart card. In 2010, Khan and Alghathbar [15] considered that in the protocol [14], users could not update their passwords and would be subject to internal privilege attacks. To solve these security vulnerabilities, they improved the protocol based on [14]. Chen and Shih [16] believed that [14] had security flaws in mutual authentication. To solve these flaws, they proposed a mutual authentication protocol that could be robust in wireless sensor networks. Vaidya et al. [17] found that Das's protocol [14] could be attacked by stolen smart card attacks, password guessing attacks, and other attacks, so Vaidya et al. improved a two-factor authentication protocol in the WSN environment. In 2016, Vaidya et al. [18] believed that [14–16] would be subject to stolen smart card attacks and sensor impersonation attacks and proposed two-factor authentication based on the key agreement in WSNs. Kim et al. [19] pointed out that [18] cannot resist gateway node bypass attacks and user impersonation attacks and eliminated these security flaws by improving the scheme. With the rapid development of WSNs, more and more two-factor schemes have been proposed in the wireless sensor network environments [20–23].

To solve the security vulnerabilities in two-factor authentication (such as stolen smart card attacks and password guessing attacks), biometric data is added as the third factor to the authentication scheme of the wireless sensor network. In 2010, Yuan et al. [24] found that Wong et al.'s dynamic authentication scheme [25] was vulnerable to the threat of the same ID and the stolen-verifier attack. They proposed a scheme based on biometric user authentication in the wireless sensor network environment. In 2011, Yoon and Yoo [26] found that Yuan et al.'s scheme [24] would be subject to an insider attack and impersonation attack and also had message integrity problems. Then, they proposed a wireless sensor network authentication scheme based on the smart card and biometric without the password. In 2013, Althobaiti et al. [27] pointed out that Yoon et al.'s scheme [26] would be subject to denial of service attacks and proposed an efficient authentication protocol based on biometric for WSNs. In 2015, Das [28] proposed a three-factor user authentication scheme for distributed WSNs. In 2017, Das [29] also proposed a new user authentication scheme based on biometrics. In the same year, Maurya and Sastry [30] considered that [29] would be attacked by a stolen smart card and proposed efficient user authentication protocols for WSNs and the IoTs. In 2018, Wu et al. [31] believed that both [28, 29] had security vulnerabilities such as offline password guessing attacks, user impersonation attacks, and violation of perfect forward security and then proposed an improved three-factor scheme. In the same year, Das et al. [32] proposed an authentication scheme based on biometrics to protect user privacy in the cloud environment. Then, Ryu et al. [33] pointed out that [31] could not provide user anonymity and was also subject to user impersonation attacks. In 2019, Hussain and Chaudhry [34] found that [32] would be subject to the smart card stolen attacks and traceability attacks and could not provide perfect forward security. In the same year, Chen et al. [35] proposed an improved three-factor authentication scheme under the medical wireless sensor network.

Recently, Wu et al. [36] believed that [32, 35] were attacked by the off-line password guessing attacks. Therefore, they proposed a new three-factor authentication protocol for wireless sensor networks with the concept of the Internet of Things and claimed that the protocol has higher security advantages. However, we found that their protocol cannot resist key compromise impersonation attacks, violates perfect forward security, cannot provide anonymity, and cannot resist known session-specific temporary information attacks. This paper presents an improved three-factor authentication protocol for provable security. Through the formal analysis in the Real-Or-Random (ROR) model and the informal analysis, the security of the protocol is proved. Further, we also prove the security through BAN logic and ProVerif tools. The comparison of security and performance proves that the improved protocol has higher security and lower computational overhead.

The framework of the rest of this paper is as follows. In the second and third sections, we give a brief review and cryptanalysis of the protocol proposed by Wu et al. Section 4 describes the improved protocol in detail. Section 5 is the security proof of the improved protocol. Section 6 is the comparison of performance and security. Section 7 is the summary of the whole paper.

2. Review of Wu et al.'s Protocol

Wu et al.'s protocol [36] mainly includes two phases: registration and authentication and key exchange. The symbols and descriptions used in this paper are shown in Table 1.

2.1. Registration. Sensor Node Registration. Sensor S_j selects its own identity SID_j and sends SID_j to gateway node $GW N$. Then, GWN selects x as the master key and computes $S M_j = h(SID_j||x)$. Finally, GWN sends SM_j to S_j.

User Registration. User U_i selects his own ID_i and sends it to the system administrator SA. Then, SA checks whether ID_i exists in its database. If it exists, reject the request. Otherwise, SA selects SCN_i, PID_i and computes $B_1 = h(PID_i||x)$, $B_2 = h(SCN_i||x)$. The values $\{B_1, B_2, SCN_i, PID_i, H(\cdot)\}$ are stored in a smart card SC and ID_i is stored in SA's database. Finally, SA sends SC to U_i. Upon receiving the smart card, U_i enters his PW_i, B_i, selects r_0, and computes $C_0 = H(B_i)$, $P_i = h(C_0 \|PW_i\|r_0)$, $C_1 = B_1 \oplus h(ID_i||P_i)$, $C_2 = B_2 \oplus h(ID_i||PW_i)$, and $C_3 = r_0 \oplus h(ID_i\|PW_i\|C_0)$. Then, U_i stores $\{C_1, C_2, C_3\}$ to SC and deletes $\{B_1, B_2\}$ from SC. Note that, all communications in this phase are based on a secure channel.

2.2. Authentication and Key Exchange. U_i inserts SC and enters ID_i, PW_i, and B_i. Then, the smart card selects N_1, T_1 and computes $C_0 = H(B_i)$, $r_0 = C_3 \oplus h(ID_i\|PW_i\|C_0)$, $P_i = h(C_0\|PW_i\|r_0)$, $B_1 = C_1 \oplus h(ID_i||P_i)$, $B_2 = C_2 \oplus h(ID_i||PW_i)$, $D_1 = B_1 \oplus N_1$, $D_2 = ID_i \oplus h(PID_i\|N_1\|T_1)$, $D_3 = SCN_i \oplus h(ID_i\|N_1\|T_1)$, $D_4 = SID_j \oplus h(B_2\|N_1\|T_1)$, $D_5 = h(ID_i\|PID_i\|SCN_i\|N_1\|SID_j)$. Finally, U_i sends $M_1 = \{PID_i, D_1, D_2, D_3, D_4, D_5, T_1\}$ to GWN.

GWN first checks whether T_1 is valid. If it times out, the request is terminated. Otherwise, GWN calculates $B_1 = h(PID_i||x)$, $N_1 = D_1 \oplus B_1$, $ID_i = D_2 \oplus h(PID_i\|N_1\|T_1)$ and then searches for ID_i in its database. If it is not found, terminates. Otherwise, GWN computes $SCN_i = D_3 \oplus h(ID_i\|N_1\|T_1)$, $B_2 = h(SCN_i||x)$, $SID_j = D_4 \oplus h(B_2\|N_1\|T_1)$, and verifies $D_5 \stackrel{?}{=} h(ID_i\|PID_i\|SCN_i\|N_1\|SID_j)$. If the verification holds, GWN selects T_2 and calculates $SM_j = h(SID_j||x)$, $D_6 = N_1 \oplus h(ID_g\|SM_j\|T_2)$, and $D_7 = h(N_1\|SM_j\|SID_j)$. Finally, GWN sends $M_2 = \{D_6, D_7, ID_g, T_2\}$ to S_j.

S_j first checks whether T_2 is valid. If it times out, the communication is terminated. Otherwise, S_j calculates $N_1 = D_6 \oplus h(ID_g\|SM_j\|T_2)$ and verifies $D_7 \stackrel{?}{=} h(N_1\|SM_j\|SID_j)$. If the verification holds, S_j selects T_3, N_2 and computes $SK_s = h(N_1||N_2)$, $D_8 = N_1 \oplus N_2$, and $D_9 = h(SK_s\|SM_j\|ID_g\|SID_j\|T_3)$. Finally, S_j sends $M_3 = \{D_8, D_9, T_3\}$ to GWN.

GWN first checks whether T_3 is valid. If it times out, the communication is terminated. Otherwise, GWN calculates $N_2 = D_8 \oplus N_1$, $SK_g = h(N_1||N_2)$ and verifies $D_9 \stackrel{?}{=} h(SK_g\|SM_j\|ID_g\|SID_j\|T_3)$. If the verification holds, GWN selects T_4, PID_i^{new} and computes $B_1^{\text{new}} = h(PID_i^{\text{new}}||x)$, $D_{10} = B_1^{\text{new}} \oplus h(B_1\|N_1\|T_4)$, $D_{11} = PID_i^{\text{new}} \oplus h(B_1^{\text{new}}\|N_2\|T_4)$, and $D_{12} = h(SK_g\|B_1^{\text{new}}\|PID_i^{\text{new}}\|B_1\|ID_i\|SID_j)$. Finally, GWN sends $M_4 = \{D_8, D_{10}, D_{11}, D_{12}, T_4\}$ to U_i.

U_i first checks whether T_4 is valid. If it times out, the communication is terminated. Otherwise, U_i calculates $N_2 = D_8 \oplus N_1$, $B_1^{\text{new}} = D_{10} \oplus h(B_1\|N_1\|T_4)$, $PID_i^{\text{new}} = D_{11} \oplus h(B_1^{\text{new}}\|N_2\|T_4)$, $SK_u = h(N_1||N_2)$, and verifies $D_{12} \stackrel{?}{=} h(SK_u\|B_1^{\text{new}}\|PID_i^{\text{new}}\|B_1\|ID_i\|SID_j)$. If the verification holds, U_i computes $C_1^{\text{new}} = B_1^{\text{new}} \oplus h(ID_i||P_i)$ and stores $\{C_1^{\text{new}}, PID_i^{\text{new}}\}$ to the smart card and deletes the old $\{C_1, PID_i\}$.

After finish the above steps, U_i, GWN, and S_j can establish a session $SK = SK_u = SK_g = SK_s = h(N_1||N_2)$ to communicate. Note that, B_1^{new} and PID_i^{new} are used in the next section.

Table 1: Symbols and descriptions.

Symbol	Description
U_i	User
S_j	Sensor
GWN	Gateway
$\mathcal{A}$	Adversary
SA	System administrator
SC	Smart card
SK	Session key
x, ID_g	GWN's master key and identity
s_j	S_j's secret value
ID_i, PW_i, B_i	U_i's identity, password, and biometrics
T_i	Timestamp
$Gen(\cdot)$	Fuzzy generator function
$Rep(\cdot)$	Fuzzy reproduction function
$h(\cdot)$	Hash function

3. Cryptanalysis of Wu et al.'s Protocol

In this section, we found that Wu et al.'s protocol [36] is subject to key compromise impersonation attacks and known session-specific temporary information attacks. Meanwhile, their protocol violates perfect forward secrecy and anonymity.

Here, we define the capabilities of adversary $\mathcal{A}$ according to the literature [29, 35, 37].

(1) Messages transmitted over public channels can be eavesdropped, intercepted, modified, and replayed by $\mathcal{A}$

(2) $\mathcal{A}$ may try to guess the user's password and identity in polynomial time

(3) $\mathcal{A}$ may successfully steal the user's SC such that some important parameters can be obtained by $\mathcal{A}$

(4) $\mathcal{A}$ may obtain the long-term key of each entity

Note that stealing the smart card and obtaining the long-term key cannot be performed at the same time in our proposed following attacks.

3.1. Key Compromise Impersonation Attacks. Key compromise impersonation attacks [38] mean that adversary $\mathcal{A}$ knows the long-term key of one entity and tries to impersonate the other entity. Here, we assume that $\mathcal{A}$ obtains the long-term private key x of GWN. After intercepting $M_1 = \{PID_i, D_1, D_2, D_3, D_4, D_5, T_1\}$, $\mathcal{A}$ can recover $B_1 = h(PID_i||x)$, $N_1 = D_1 \oplus B_1$, $ID_i = D_2 \oplus h(PID_i||N_1||T_1)$, $SCN_i = D_3 \oplus h(ID_i||N_1||T_1)$, $B_2 = h(SCN_i||x)$, $SID_j = D_4 \oplus h(B_2||N_1||T_1)$, and $SM_j = h(SID_j||x)$.

In the following, we show that $\mathcal{A}$ can impersonate S_j to establish a session key with U_i by the above values.

(1) $\mathcal{A}$ intercepts $M_2 = \{D_6, D_7, ID_g, T_2\}$ and selects a random number N_2' and timestamp T_A. Then, $\mathcal{A}$ computes $SK_A = h(N_1||N_2')$, $D_8' = N_1 \oplus N_2'$, $D_9' = h(SK_A||SM_j||ID_g||SID_j||T_A)$ and sends $M_3' = \{D_8', D_9', T_A\}$ to GWN

(2) GWN checks whether T_A is valid. If it times out, the communication is terminated. Otherwise, GWN calculates $N_2' = D_8' \oplus N_1$, $SK_g = h(N_1||N_2')$, and verifies $D_9' = h(SK_g||SM_j||ID_g||SID_j||T_A)$. The following steps are similar to the authentication phase in Subsection 2.2 except $D_{11}' = PID_i^{new} \oplus h(B_1^{new}||N_2'||T_4)$. Then, GWN sends $M_4' = \{D_8', D_{10}, D_{11}', D_{12}, T_4\}$ to U_i

(3) U_i checks whether T_4 is valid. If it times out, the communication is terminated. Otherwise, U_i calculates $N_2' = D_8' \oplus N_1$, $B_1^{new} = D_{10} \oplus h(B_1||N_1||T_4)$, $PID_i^{new} = D_{11}' \oplus h(B_1^{new}||N_2'||T_4)$, $SK_u = h(N_1||N_2')$, and verifies $D_{12} \stackrel{?}{=} h(SK_u||B_1^{new}||PID_i^{new}||B_1||ID_i||SID_j)$. It is easy to see that the result is true

Thus, U_i believes that he can establish a session key $SK = SK_u = SK_A = h(N_1||N_2')$ with S_j (impersonated by $\mathcal{A}$).

3.2. Violating Perfect Forward Secrecy and Anonymity. By the similar attack approach in Subsection 3.1, suppose that $\mathcal{A}$ gets x and intercepts M_1, M_3. Then, $\mathcal{A}$ can recover $ID_i = D_2 \oplus h(PID_i||N_1||T_1)$ and $SK = h(N_1||N_2)$, where $B_1 = h(PID_i||x)$, $N_1 = D_1 \oplus B_1$, $N_2 = D_8 \oplus N_1$. In other words, Wu et al.'s protocol violates perfect forward secrecy and anonymity.

3.3. Known Session-Specific Temporary Information Attacks. Here, assume that the adversary $\mathcal{A}$ gets the temporary value N_1 and intercepts $M_3 = \{D_8, D_9, T_3\}$. Then, $\mathcal{A}$ can recover the current session key $SK = h(N_1||N_2)$, where $N_2 = D_8 \oplus N_1$. Furthermore, $\mathcal{A}$ can compute update values $B_1^{new} = D_{10} \oplus h(B_1||N_1||T_4)$ and $PID_i^{new} = D_{11} \oplus h(B_1^{new}||N_2||T_4)$ by intercepting M_1, M_3, and M_4, where $B_1 = N_1 \oplus D_1$.

In the next section, $\mathcal{A}$ may intercept messages M_1', M_3', and M_4' to recover $N_1' = B_1^{new} \oplus D_1'$, $N_2' = N_1' \oplus D_8'$. The session key SK' can be computed by $SK' = h(N_1'||N_2')$. Meanwhile, the newest updated values $B_1^{new'} = D_{10}' \oplus h(B_1^{new}||N_1'||T_4')$, $PID_i^{new'} = D_{11}' \oplus h(B_1^{new'}||N_2'||T_4')$ can be computed. Thus, under a known session-specific temporary information attack approach, we can conclude that Wu et al.'s protocol not only violates "perfect forward secrecy" but also not provides "backward secrecy."

4. Improved Protocol

In order to fix our proposed security flaws of Wu et al.'s protocol [36], an enhanced protocol is present.

4.1. Registration. Sensor Node Registration. S_j selects SID_j, s_j and sends $\{SID_j, s_j\}$ to GWN via a secure channel. Then, GWN calculates $SM_j = h(SID_j||s_j||x)$, $s_1 = s_j \oplus SM_j$, and stores s_j in its database. Finally, GWN sends s_1 to S_j. After receiving s_1, S_j computes $SM_j = s_j \oplus s_1$ and stores it in its memory.

User Registration. U_i selects ID_i, PW_i and inputs his B_i to compute $P_i = h(\sigma_i||PW_i||ID_i)$ and $HID_i = h(ID_i||\sigma_i)$, where $Gen(B_i) = (\sigma_i, \tau_i)$. Then, U_i sends $\{ID_i, P_i, HID_i\}$ to SA via a secure channel. After receiving $\{ID_i, P_i, HID_i\}$, SA checks whether ID_i exits its database. If so, deleting the relevant records in the database and reregister. Otherwise, SA selects g_i and computes $A_1 = h(g_i||HID_i||x||ID_i)$, $A_2 = A_1 \oplus P_i$, and $A_3 = h(HID_i||P_i)$. Then, SA stores $\{A_2, A_3\}$ in SC and sends SC to U_i via a secure channel. Meanwhile, $\{HID_i, ID_i, g_i\}$ is stored in SA's database. After receiving SC, U_i stores τ_i in SC.

The sensor node registration phase and the user registration phase are shown in Figure 2.

4.2. Authentication and Key Exchange. U_i inserts SC and enters ID_i, PW_i, and B_i. Then, U_i can compute $\sigma_i = Rep(B_i, \tau_i)$, $P_i = h(\sigma_i||PW_i||ID_i)$, $HID_i = h(ID_i||\sigma_i)$, $A_3' = h(HID_i||P_i)$ to check whether A_3' is equal to A_3. If the verification holds, U_i generates N_1, T_1, and computes $A_1 = A_2 \oplus P_i$, $C_1 = N_1 \oplus h(A_1||HID_i)$, $C_2 = ID_i \oplus h(HID_i||A_1||T_1)$, $C_3 = SID_j \oplus h(A_1||N_1||T_1)$, $C_4 = h(ID_i||HID_i||SID_j||N_1||T_1)$. Finally, U_i sends $M_1 = \{HID_i, C_1, C_2, C_3, C_4, T_1\}$ to GWN.

Upon receiving M_1, GWN first checks whether T_1 is valid. If the times out, the communication is terminated. Otherwise, GWN according to HID_i finds the corresponding $\{ID_i, g_i\}$ in its database and computes $A_1 = h(g_i||HID_i||x||ID_i)$, $ID_i^* = C_2 \oplus h(HID_i||A_1||T_1)$. Then, GWN checks whether ID_i^* equals to ID_i. If not, the session is terminated. Otherwise, GWN computes $N_1 = C_1 \oplus h(A_1||HID_i)$, $SID_j = C_3 \oplus h(A_1||N_1||T_1)$ and verifies $C_4 \stackrel{?}{=} h(ID_i^*||HID_i||SID_j||N_1||T_1)$. If the verification holds, GWN generates N_2, T_2 and computes $SM_j = h(SID_j||s_j||x)$, $C_5 = N_2 \oplus h(SID_j||SM_j||T_2)$, $C_6 = N_1 \oplus h(SM_j||N_2)$, $C_7 = h(N_1||N_2||SID_j||SM_j||T_2)$. Finally, GWN sends $M_2 = \{HID_i, C_5, C_6, C_7, T_2\}$ to S_j.

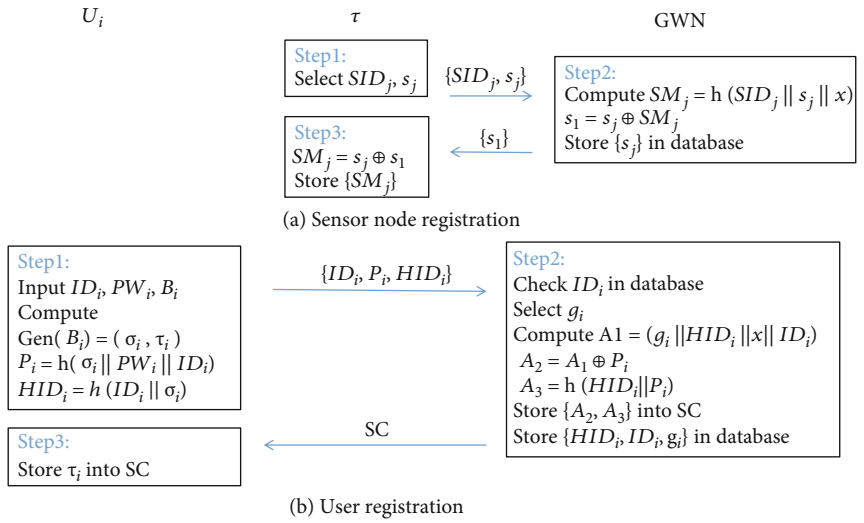

Figure 2: The registration phase.

Upon receiving M_2, S_j first checks whether T_2 is valid. If the times out, the communication is terminated. Otherwise, S_j computes $N_2 = C_5 \oplus h(SID_j \| SM_j \| T_2)$, $N_1 = C_6 \oplus h(SM_j \| N_2)$ and verifies $C_7 \stackrel{?}{=} h(N_1 \| N_2 \| SID_j \| SM_j \| T_2)$. If the verification holds, S_j generates N_3, T_3 and computes $SK_s = h(N_1 \| N_2 \| N_3 \| HID_j \| SID_j)$, $C_8 = N_3 \oplus h(N_1 \| N_2)$, $C_9 = h(SK_s \| SM_j \| SID_j \| T_3)$. Finally, S_j sends $M_3 = \{C_8, C_9, T_3\}$ to GWN.

Upon receiving M_3, GWN first checks whether T_3 is valid. If times out, the communication is terminated. Otherwise, GWN computes $N_3 = C_8 \oplus h(N_1 \| N_2)$, $SK_g = h(N_1 \| N_2 \| N_3 \| HID_j \| SID_j)$ and verifies $C_9 \stackrel{?}{=} h(SK_g \| SM_j \| SID_j \| T_3)$. If the verification holds, GWN generates T_4 and computes $C_{10} = N_2 \oplus h(A_1 \| P_i \| N_1 \| T_4)$, $C_{11} = h(SK_g \| A_1 \| P_i \| ID_i \| T_4)$. Finally, GWN sends $M_4 = \{C_8, C_{10}, C_{11}, T_4\}$ to U_i.

Upon receiving M_4, U_i first checks whether T_4 is valid. If times out, the communication is terminated. Otherwise, U_i computes $N_2 = C_{10} \oplus h(A_1 \| P_i \| N_1 \| T_4)$, $N_3 = C_8 \oplus h(N_1 \| N_2)$, $SK_u = h(N_1 \| N_2 \| N_3 \| HID_j \| SID_j)$ and verifies $C_{11} \stackrel{?}{=} h(SK_u \| A_1 \| P_i \| ID_i \| T_4)$. If the verification holds, $SK_u = SK_g = SK_s$ is set as a session key used to communicate between U_i, GWN, and S_j.

The authentication and key exchange phase is shown in Figure 3.

5. Proof of Security

5.1. Correctness by BAN Logic. In this subsection, we use BAN logic to show the correctness of our improved protocol. As far as the proposed protocol is concerned, we need to prove that U_i, S_j, and GWN share a session key SK through rigorous logical analysis. The symbols and rules used for BAN logic are referred to [39–41].

5.1.1. Rules

(i) $R1$ (Message meaning (M-M) rule): $(P|\equiv P \rightleftharpoons^{Y} Q, P \lhd X_Y)/(P|\equiv Q|\sim X)$

(ii) $R2$ (Nonce verification (N-V) rule): $(P|\equiv \#(X), P|\equiv Q|\sim X)/(P|\equiv Q|\equiv X)$

(iii) $R3$ (Jurisdiction rule): $(P|\equiv Q|\Rightarrow X, P|\equiv Q|\equiv X)/(P|\equiv X)$

(iv) $R4$ (Session key (S-K) rule): $(P|\equiv \#(X), P|\equiv Q|\equiv X)/(P|\equiv P \longleftrightarrow^{K} Q)$

5.1.2. Goals

(i) $G1$: $U_i \,|\equiv U_i \longleftrightarrow^{SK} S_j$

(ii) $G2$: $S_j \,|\equiv U_i \longleftrightarrow^{SK} S_j$

(iii) $G3$: $GWN \,|\equiv U_i \longleftrightarrow^{SK} S_j$

(iv) $G4$: $U_i \,|\equiv S_j \,|\equiv U_i \longleftrightarrow^{SK} S_j$

(v) $G5$: $S_j \,|\equiv U_i \,|\equiv U_i \longleftrightarrow^{SK} S_j$

(vi) $G6$: $GWN \,|\equiv U_i \,|\equiv U_i \longleftrightarrow^{SK} S_j$

(vii) $G7$: $GWN \,|\equiv S_j \,|\equiv U_i \longleftrightarrow^{SK} S_j$

5.1.3. Idealize the Communication Messages

(i) $M_1 : U_i \longrightarrow GWN : \{HID_i, C_1, C_2, C_3, C_4, T_1\}$.

(ii) $M_2 : GWN \longrightarrow S_j : \{HID_i, C_5, C_6, C_7, T_2\}$.

(iii) $M_3 : S_j \longrightarrow GWN : \{C_8, C_9, T_3\}$.

(iv) $M_4 : GWN \longrightarrow U_i : \{C_{10}, C_{11}, T_4\}$.

(v) $M_5 : S_j \longrightarrow U_i : \{C_8\}$

5.1.4. Initial Assumptions

(i) $A1 : U_i \,|\equiv \#(N_1)$

(ii) $A2 : S_j \,|\equiv \#(N_3)$

(iii) $A3 : GWN \,|\equiv \#(N_2)$

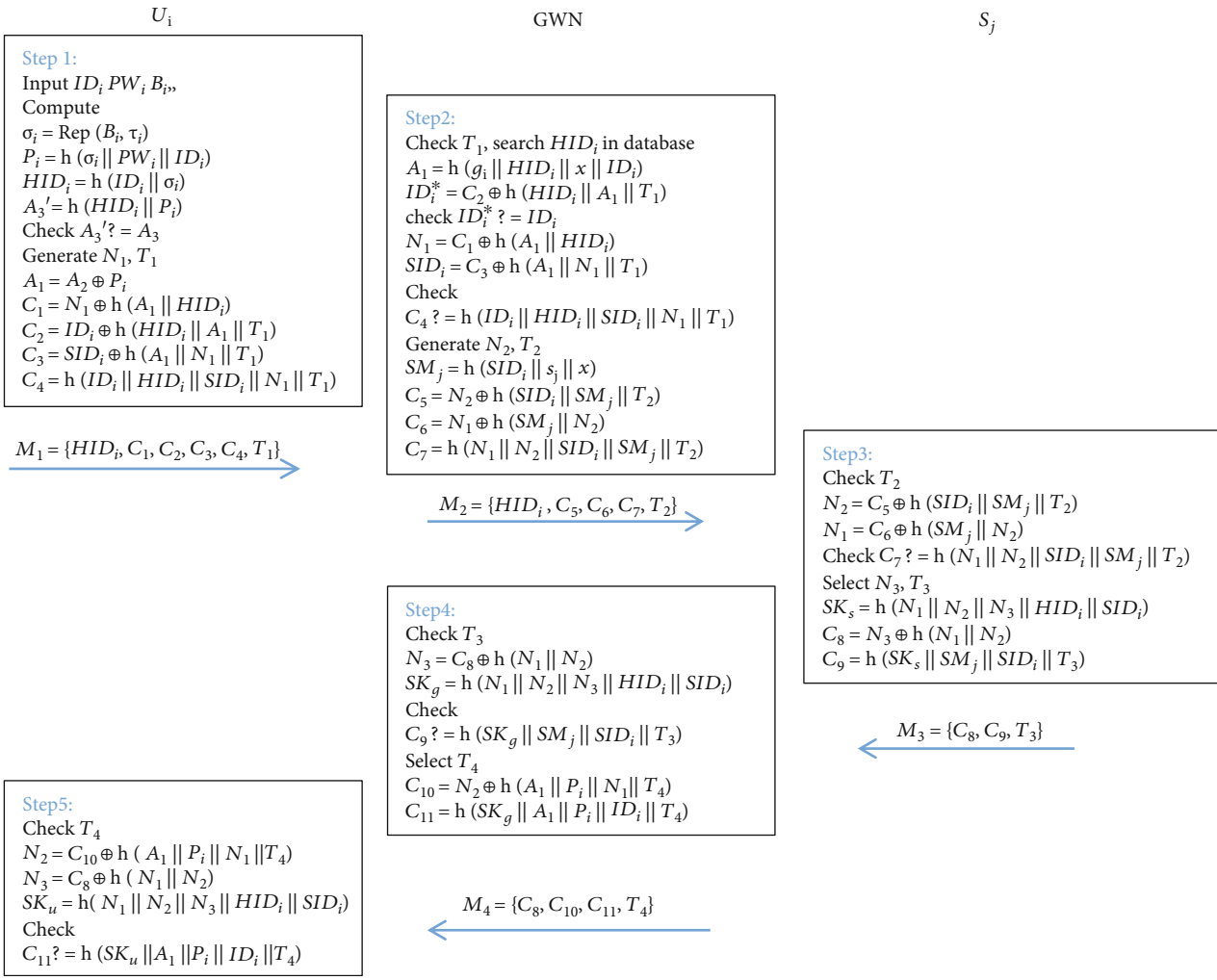

Figure 3: The authentication and key exchange phase.

(iv) $A4 : GWN \mid \equiv U_i \mid \Rightarrow HID_i$

(v) $A5 : GWN \mid \equiv U_i \rightleftharpoons^{A_1} GWN$

(vi) $A6 : GWN \mid \equiv \#(N_1)$

(vii) $A7 : GWN \mid \equiv U_i \mid \Rightarrow N_1$

(viii) $A8 : GWN \mid \equiv \#(SID_j)$

(ix) $A9 : GWN \mid \equiv U_i \mid \Rightarrow SID_j$

(x) $A10 : S_j \mid \equiv GWN \mid \Rightarrow HID_i$

(xi) $A11 : S_j \mid \equiv GWN \mid \equiv SM_j$

(xii) $A12 : S_j \mid \equiv \#(SM_j)$

(xiii) $A13 : S_j \mid \equiv \#(N_2)$

(xiv) $A14 : S_j \mid \equiv GWN \mid \Rightarrow N_2$

(xv) $A15 : S_j \mid \equiv \#(N_1)$

(xvi) $A16 : GWN \mid \equiv S_j \rightleftharpoons^{h(N_1||N_2)} GWN$

(xvii) $A17 : GWN \mid \equiv \#(N_3)$

(xviii) $A18 : GWN \mid \equiv S_j \mid \Rightarrow N_3$

(xix) $A19 : U_i \mid \equiv GWN \mid \equiv A_1$

(xx) $A20 : U_i \mid \equiv \#(A_1)$

(xxi) $A21 : U_i \mid \equiv \#(N_2)$

(xxii) $A22 : U_i \mid \equiv GWN \mid \Rightarrow N_2$

(xxiii) $A23 : U_i \mid \equiv U_i \rightleftharpoons^{h(N_1||N_2)} S_j$

(xxiv) $A24 : U_i \mid \equiv \#(N_3)$

(xxv) $A25 : U_i \mid \equiv S_j \mid \Rightarrow N_3$

(xxvi) $A26 : S_j \mid \equiv GWN \mid \Rightarrow N_1$

5.1.5. The Proof of our Proposed Protocol via BAN Logic. By M_1, we have $S1 : GWN \triangleleft \{HID_i, C_1 : N_1, HID_{iA_1}, C_2, C_3 : SID_{jA_1}, C_4, T_1\}$ and further $S2 : GWN \mid \equiv U_i \mid \equiv HID_i$. Base on $A4$, $S2$, and $R3$ (Jurisdiction rule), we can obtain $S3 : GW$

$N \mid \equiv HID_i$. According to $S1$, it implies $S4 : GWN \triangleleft N_1, HID_{iA_1}$. By $A5$, $S4$, and $R1$ (M-M rule), it implies $S5 : GWN \mid \equiv U_i \mid \sim (N_1, HID_i)$. By $A6$, $S5$, and $R2$ (N-V rule), we can obtain $S6 : GWN \mid \equiv U_i \mid \equiv N_1$. According to $A7$, $S6$, and $R3$ (Jurisdiction rule), it implies $S7 : GWN \mid \equiv N_1$. According to $S1$, we have $S8 : GWN \triangleleft SID_{jA_1}$. By $A5$, $S8$, and $R1$ (M-M rule), it implies $S9 : GWN \mid \equiv U_i \mid \sim SID_j$. By $A8$, $S9$, and $R2$ (N-V rule), we can obtain $S10 : GWN \mid \equiv U_i \mid \equiv SID_j$. According to $A9$, $S10$, and $R3$ (Jurisdiction rule), it implies $S11 : GWN \mid \equiv SID_j$.

By M_2, we have $S12 : S_j \triangleleft \{HID_i, C_5 : N_{2h(SID_j \| SM_j \| T_2)}, C_6 : N_{1h(SM_j \| N_2)}, C_7, T_2\}$ and further $S13 : S_j \mid \equiv GWN \mid \equiv HID_i$. Base on $A10$, $S13$, and $R3$ (Jurisdiction rule), we can obtain $S14 : S_j \mid \equiv HID_i$. By $A11$, $A12$, and $R4$ (S-K rule), it implies $S15 : S_j \mid \equiv S_j \rightleftharpoons^{h(SID_j \| SM_j \| T_2)} GWN$. According to $S12$, we have $S16 : S_j \triangleleft N_{2h(SID_j \| SM_j \| T_2)}$. Base on $S15$, $S16$, and $R1$ (M-M rule), it implies $S17 : S_j \mid \equiv GWN \mid \sim N_2$. By $A13$, $S17$ and $R2$ (N-V rule), we can obtain $S18 : S_j \mid \equiv GWN \mid \equiv N_2$. According to $A14$, $S18$, and $R3$ (Jurisdiction rule), it implies $S19 : S_j \mid \equiv N_2$. Base on $A11$, $A12$, and $R4$ (S-K rule), we have $S20 : S_j \mid \equiv S_j \rightleftharpoons^{h(SM_j \| N_2)} GWN$. According to $S12$, we have $S21 : S_j \triangleleft N_{1h(SM_j \| N_2)}$. By $S20$, $S21$, and $R1$ (M-M rule), it implies $S22 : S_j \mid \equiv GWN \mid \sim N_1$. By $A15$, $S22$, and $R2$ (N-V rule), we can obtain $S23 : S_j \mid \equiv GWN \mid \equiv N_1$. Base on $A26$, $S23$, and $R3$ (Jurisdiction rule), it implies $S24 : S_j \mid \equiv N_1$. Since $SK = h(N_1 \| N_2 \| N_3 \| HID_j \| SID_j)$, $S25 : S_j \mid \equiv U_i \longleftrightarrow^{SK} S_j$ is obtained. (**G2**) According to $A2$, $S25$, and $R4$ (S-K rule), we can obtain $S26 : GWN \mid \equiv U_i \rightleftharpoons^{A_1} GWN$. (**G5**).

By M_3, we have $S27 : GWN \triangleleft \{C_8 : N_{3h(N_1 \| N_2)}, C_9, T_3\}$ and further $S28 : GWN \triangleleft N_{3h(N_1 \| N_2)}$. Base on $A16$, $S28$, and $R1$ (M-M rule), we can obtain $S29 : GWN \mid \equiv S_j \mid \sim N_3$. By $A17$, $S29$ and $R2$ (N-V rule), it implies $S30 : GWN \mid \equiv S_j \mid \equiv N_3$. Base on $A18$, $S30$ and $R3$ (Jurisdiction rule), we can obtain $S31 : GWN \mid \equiv N_3$. According to $S2$, $S11$, and $S31$, it implies $S32 : GWN \mid \equiv U_i \longleftrightarrow^{SK} S_j$. (**G3**) Base on $A6$, $S32$, and $R4$ (S-K rule), we can obtain $S33 : GWN \mid \equiv U_i \mid \equiv U_i \longleftrightarrow^{SK} S_j$. (**G6**) According to $A27$, $S32$, and $R4$ (S-K rule), it implies $S34 : GWN \mid \equiv S_j \mid \equiv U_i \longleftrightarrow^{SK} S_j$. (**G7**)

By M_4, we have $S35 : U_i \triangleleft \{C_{10} : N_{2h(A_1 \| P_i \| N_1 \| T_4)}, C_{11}, T_4\}$. Base on $A19$, $A20$, and $R4$ (S-K rule), we can obtain $S36 : U_i \mid \equiv U_i \rightleftharpoons^{h(A_1 \| P_i \| N_1 \| T_4)} GWN$. According to $S35$, we have $S37 : U_i \triangleleft N_{2h(A_1 \| P_i \| N_1 \| T_4)}$. Base on $S36$, $S37$, and $R1$ (M-M rule), it implies $S38 : U_i \mid \equiv GWN \mid \sim N_2$. By $A21$, $S38$, and $R2$ (N-V rule), we can obtain $S39 : U_i \mid \equiv GWN \mid \equiv N_2$. According to $A22$, $S39$, and $R3$ (Jurisdiction rule), it implies $S40 : U_i \mid \equiv N_2$.

By M_5, we have $S41 : U_i \triangleleft \{C_8 : N_{3h(N_1 \| N_2)}\}$. Base on $A23$, $S41$, and $R1$ (M-M rule), it implies $S42 : U_i \mid \equiv S_j \mid \sim N_3$. By $A24$, $S42$, and $R2$ (N-V rule), we can obtain $S43 : U_i \mid \equiv S_j \mid \equiv N_3$. Base on $A25$, $S43$, and $R3$ (Jurisdiction rule), it implies $S44 : U_i \mid \equiv N_3$. According to $S40$ and $S44$, we can obtain $S45 : U_i \mid \equiv U_i \longleftrightarrow^{SK} S_j$. (**G1**) According to $A24$ and $S45$, we can obtain $S45 : U_i \mid \equiv S_j \mid \equiv U_i \longleftrightarrow^{SK} S_j$. (**G4**)

5.2. Formal Security Analysis. In this section, we perform a formal security analysis of the improved protocol in ROR model [42–48]. The proposed protocol involves three entities, U_i, S_j, and GWN. We use Π_U^x,, and Π_{GWN}^z to represent the xth instance of U_i, the yth instance of S_j, and the zth instance of GWN, respectively. Here, we define that adversary $\mathscr{A}$ has the ability to initiate the following query. Note that, $\mathscr{O} = \{\Pi_U^x, \Pi_S^y, \Pi_{GWN}^z\}$.

(i) *Execute*($\mathscr{O}$): if $\mathscr{A}$ executes this query, it can obtain an entire communication record on the public channel

(ii) *Send*($\mathscr{O}$, M): if $\mathscr{A}$ executes this query, it can send M to $\mathscr{O}$ and receive the response from $\mathscr{O}$

(iii) *Hash*(*string*): if $\mathscr{A}$ executes this query, it can input *string* to get its hash value

(iv) *Corrupt*($\mathscr{O}$): if $\mathscr{A}$ executes this query, it can get secret values of one party, such as some parameter stored in the smart card, long-term secret key, or temporary information

(v) *Test*($\mathscr{O}$): if $\mathscr{A}$ executes this query, it flips a coin $\mathscr{C}$. If $\mathscr{C} = 1$, then can get the correct session key; if $\mathscr{C} = 0$, $\mathscr{A}$ gets a random string of the same length as the session key

Theorem 1. *In the ROR model, assume that $\mathscr{A}$ can make Execute, Send, Hash, Corrupt, and Test queries. Then, the advantage of $\mathscr{A}$ to break the proposed protocol $\mathscr{P}$ in polynomial time ξ is $Adv_{\mathscr{A}}^{\mathscr{P}}(\xi) \le q_{send}/2^{l-2} + 3q_{hash}^2/2^{l-1} + 2\max\{C' \cdot q_{send}^{s'}, q_{send}/2^l\}$, where q_{send} is the number of times to execute Send queries, q_{hash} is the number of times to execute Hash queries, C' and s' are two constants [49], and l is the bits of biological information.*

Proof. We prove this theorem by following game sequences GM_0 to GM_5. $\mathrm{Succ}_{\mathscr{A}}^{GM_n}(\xi)$ is defined by the probability that $\mathscr{A}$ succeeds in GM_n, which is the probability that $\mathscr{C} = 1$. The detailed simulations of queries in real attacks are shown in Tables 2 and 3. The details are as follows.

GM_0 : Flip $\mathscr{C}$ to start the game. GM_0 is a game played without any queries. Therefore, we can get the probability of $\mathscr{A}$ successfully breaking $\mathscr{P}$ as

$$Adv_{\mathscr{A}}^{\mathscr{P}}(\xi) = \left| 2\Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_0}(\xi)\right] - 1 \right|. \quad (1)$$

GM_1 : The difference between GM_1 and GM_0 is that GM_1 adds the *Execute* query. In GM_1, $\mathscr{A}$ just gets messages $M_1 = \{HID_i, C_1, C_2, C_3, C_4, T_1\}$, $M_2 = \{HID_i, C_5, C_6, C_7, T_2\}$, $M_3 = \{C_8, C_9, T_3\}$, and $M_4 = \{C_8, C_{10}, C_{11}, T_4\}$. After GM_1 is over, $\mathscr{A}$ queries the session key through *Test*, but N_1, N_2, and N_3 are all confidential to $\mathscr{A}$. Therefore,

Table 2: Simulation of *Send* query.

On a query $Send(\Pi_U^x, \text{start})$, assuming that Π_U^x is a normal state, we perform the following operations. Select N_{A1}, T_{A1}, and compute $A_1 = A_2 \oplus P_i$, $C_1 = N_1 \oplus h(A_1 \| HID_i)$, $C_2 = ID_i \oplus h(HID_i \| A_1 \| T_{A1})$, $C_3 = SID_j \oplus h(A_1 \| N_{A1} \| T_{A1})$, $C_4 = h(ID_i \| HID_i \| SID_j \| N_{A1} \| T_{A1})$. Then, the query is answered by $M_1 = \{HID_i, C_1, C_2, C_3, C_4, T_1\}$.
On a query $Send(\Pi_{GWN}^z, (HID_i, C_1, C_2, C_3, C_4, T_1))$ and assume that Π_{GWN}^z is a normal state to perform the following operations. Compute A_i, ID_i, N_1, SID_j, C_4, and check A_1. If equal, select N_{A2}, T_{A2}, and compute SM_j, C_5, C_6, C_7. Then, the query is answered by $M_2 = \{HID_i, C_5, C_6, C_7, T_2\}$.
On a query $Send(\Pi_S^y, (HID_i, C_5, C_6, C_7, T_2))$, assuming that Π_S^y is a normal state, do the following. Compute N_2, N_1, C_7, check C_7. If equal, select N_{A3}, T_{A3}, and compute SK_s, C_8, C_9. Then, the query is answered by $M_3 = \{C_8, C_9, T_3\}$.
On a query $Send(\Pi_{GWN}^z, (C_8, C_9, T_3))$ and assume that Π_{GWN}^z is a normal state to perform the following operations. Compute N_3, SK_g, C_9, and check C_9. If equal, select T_{A4}, and compute C_{10}, C_{11}. Then, the query is answered by $M_4 = \{C_8, C_{10}, C_{11}, T_4\}$.
On a query, assuming that Π_U^x is a normal state, we perform the following operations. Compute N_2, N_3, SK_u, C_{11}, the instance Π_U^x checks C_{11}; if not equal, it will be terminated. Otherwise, compute $SK = h(N_1 \| N_2 \| N_3 \| HID_j \| SID_j)$. Finally, the user instance accepts and terminates.

Table 3: Simulation of *Execute*, *Corrupt*, and *Test* query.

On a *Execute* query, we use the simulation of Send query to do the following operations: $(HID_i, C_1, C_2, C_3, C_4, T_1) \longleftarrow Send(\Pi_U^x, \text{start})$, $(HID_i, C_5, C_6, C_7, T_2) \longleftarrow Send(\Pi_{GWN}^z, (HID_i, C_1, C_2, C_3, C_4, T_1))$, $(C_8, C_9, T_3) \longleftarrow Send(\Pi_S^y, (HID_i, C_5, C_6, C_7, T_2))$, $(C_8, C_{10}, C_{11}, T_4) \longleftarrow Send(\Pi_{GWN}^z, (C_8, C_9, T_3))$. This query is answered by $(HID_i, C_1, C_2, C_3, C_4, T_1)$, $(HID_i, C_5, C_6, C_7, T_2)$, (C_8, C_9, T_3), and $(C_8, C_{10}, C_{11}, T_4)$.
For a record $(string, r)$ that appears in the $Hash(string)$ query, return $r = Hash(string)$. Otherwise, select an element r, add the record $(string, r)$ to the list, and return r.
On a query $Corrupt(\Pi_U^x)$, and if Π_U^x is accepted, the query is answered by the parameter $\{A_2, A_3, \tau_i\}$ in the smart card.
On a *Test* query, flip a coin $\mathscr{C}$ to get the result of SK. If $\mathscr{C} = 1$, return SK; otherwise, return a string of the same length.

the probability of GM_1 and GM_0 is equal, that is,

$$\Pr\left[\text{Succ}_{\mathscr{A}}^{GM_1}(\xi)\right] = \Pr\left[\text{Succ}_{\mathscr{A}}^{GM_0}(\xi)\right]. \quad (2)$$

GM_2 : The difference between GM_2 and GM_1 is that GM_2 adds the *Send* query. According to Zipf's law [49], we can get

$$\left|\Pr\left[\text{Succ}_{\mathscr{A}}^{GM_2}(\xi)\right] - \Pr\left[\text{Succ}_{\mathscr{A}}^{GM_1}(\xi)\right]\right| \le \frac{q_{send}}{2^l}. \quad (3)$$

GM_3 : The difference between GM_3 and GM_2 is that GM_3 adds the *Hash* query and deletes the *Send* query. According to the birthday paradox, we can get

$$\frac{\left|\Pr\left[\text{Succ}_{\mathscr{A}}^{GM_3}(\xi)\right] - \Pr\left[\text{Succ}_{\mathscr{A}}^{GM_2}(\xi)\right]\right| \le q_{hash}^2}{2^{l+1}}. \quad (4)$$

GM_4 : In this game, we discuss the security of the session key in two cases. The first is to obtain the long-term private key x of Π_{GWN}^z to verify the perfect forward security; the second is to get temporary information to verify whether the known session-specific temporary information attacks can be resisted.

(1) Perfect forward security. $\mathscr{A}$ uses Π_{GWN}^z to try to get the private key x of GWN or uses Π_U^x or Π_S^y to try to get a secret value in the registration phase

(2) Known session-specific temporary information attacks. $\mathscr{A}$ uses either Π_U^x or Π_S^y or Π_{GWN}^z to try to obtain the temporary information of the corresponding party

In both cases, $\mathscr{A}$ can only compute the session key through Send and Hash queries. For the first case, if $\mathscr{A}$ only knows the private key x of GWN, or a secret value of Π_U^x or Π_S^y in the registration phase, it cannot get the temporary information N_1, N_2, and N_3 in $SK = h(N_1 \| N_2 \| N_3 \| HID_j \| SID_j)$. For the second case, we assume that $\mathscr{A}$ gets N_1, but N_2 and N_3 are kept secret. Similarly, if N_2 or N_3 is leaked, the session key cannot be calculated. Therefore, we have

$$\frac{\left|\Pr\left[\text{Succ}_{\mathscr{A}}^{GM_4}(\xi)\right] - \Pr\left[\text{Succ}_{\mathscr{A}}^{GM_3}(\xi)\right]\right| \le q_{send}}{2^l + q_{hash}^2/2^{l+1}}. \quad (5)$$

GM_5 : In this game, $\mathscr{A}$ uses $Corrupt(\Pi_U^x)$ to get the parameters $\{A_2, A_3, \tau_i\}$ stored in the SC and attempts to launch the stolen smart card attacks and the offline password guessing attacks. Suppose $\mathscr{A}$ gets HID_i according to M_1, and computes $\sigma_i = Rep(B_i, \tau_i)$, $P_i = h(\sigma_i \| PW_i \| ID_i)$, $A_3' = h(HID_i$

$||P_i)$ until $A_3' = A_3$. However, B_i, PW_i, and ID_i are all confidential to $\mathscr{A}$. The probability that $\mathscr{A}$ can guess the biological information of the l bits is $1/2^l$ [50]. In Zipf's law [49], the probability of guessing the password is more than 0.5 when $q_{send} \leq 10^6$. Therefore, we get

$$|\Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_5}(\xi)\right] - \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_4}(\xi)\right]| \leq \max\left\{\frac{C' \cdot q_{send}^{s'}, q_{send}}{2^l}\right\}, \tag{6}$$

where C' and s' are constants depending on the size of the password.

GM_6 : The purpose of this game is to verify whether it can resist impersonation attacks. The difference between GM_6 and GM_5 is that when GM_6 initiates $h(N_1||N_2||N_3||HID_j||SID_j)$ query to guess the session key, the game is terminated. Therefore, we have

$$\frac{|\Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_6}(\xi)\right] - \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_5}(\xi)\right]| \leq q_{hash}^2}{2^{l+1}}. \tag{7}$$

Since the probability of GM_6 success and failure is equal, the probability of $\mathscr{A}$ successfully guessing the session key is

$$\Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_0}(\xi)\right] = \frac{1}{2}. \tag{8}$$

According to formulas (1) to (8), we can get

$$\begin{aligned}
\frac{1}{2Adv_{\mathscr{A}}^{\mathscr{P}}(\xi)} &= \left|\Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_0}(\xi)\right] - \frac{1}{2}\right| \\
&= \left|\Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_0}(\xi)\right] - \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_6}(\xi)\right]\right| \\
&= \left|\Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_1}(\xi)\right] - \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_6}(\xi)\right]\right| \\
&\leq \sum_{i=0}^{5}\left|\Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_{i+1}}(\xi)\right] - \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_i}(\xi)\right]\right| \\
&= \frac{q_{send}}{2^{l-1}} + \frac{3q_{hash}^2}{2^l} + \max\left\{C' \cdot q_{send}^{s'}, \frac{q_{send}}{2^l}\right\}.
\end{aligned} \tag{9}$$

Thus, we have $Adv_{\mathscr{A}}^{\mathscr{P}}(\xi) \leq q_{send}/2^{l-2} + 3q_{hash}^2/2^{l-1} + 2\max\{C' \cdot q_{send}^{s'}, q_{send}/2^l\}$.

5.3. Informal Security Analysis

5.3.1. Replay Attacks. The replay attacks are to send the sent message repeatedly, to launch some other attacks to interfere with normal communication. First, if M_1 is replayed, the session key cannot be successfully established between the user and the sensor, because the message cannot be validated by GWN, and further, because each round g_i and N_1 will be refreshed. So, let us see what happens when $\{M_2, M_3, M_4\}$ are replayed? If M_2 is replayed, the sensor passes the verification, and the same session key is established as the previous round, but the user will not verify this message because g_i or A_1 will be updated every round. If M_3 or M_4 is replayed, the user will not pass the verification, and the session will be terminated for the same reason as that of M_2. Therefore, our improved protocol is resistant to replay attacks.

5.3.2. Privileged-Insider Attacks. In this paper, we specify that privileged insiders only have access to the content stored in the gateway database. In other words, privileged insiders can get $\{HID_i, ID_i, g_i\}$, but to calculate sensitive information such as A_1 and A_3, they also need to obtain private information such as P_i and gateway key x, while $P_i = h(\sigma_i||PW_i||ID_i)$. Therefore, our improved protocol is resistant to privileged-insider attacks.

5.3.3. Three-Factor Secrecy. The three factors are password, smart card, and biometric information. According to the previous analysis, A_1 and P_i are the key parameters for launching an attack to compute the session key. Now, let $\mathscr{A}$ get any two of the three factors.

(1) Password and smart card. Even if $\mathscr{A}$ knows the password and can extract the parameters from SC, he cannot be able to calculate A_1 and P_i for any attack

(2) Password and biometrics. If $\mathscr{A}$ gets the password and biometrics and wants to compute A_1, he needs to know A_2 and P_i. However, A_2 is stored on a smart card

(3) Biometrics and smart card. After $\mathscr{A}$ obtains the biometric and smart card, he/she needs to know the information about PW_i and ID_i to calculate P_i, so $\mathscr{A}$ cannot compute $A_1 = A_2 \oplus P_i$

Therefore, our protocol provides three-factor secrecy.

5.3.4. User Anonymity. The real identity of the user only appears in the registration phase, as well as the authentication phase. However, in the authentication phase, the user enters his/her identity only when he/she logs in. During the authentication process, HID_i is always protecting the user's identity. Therefore, our protocol provides anonymity.

5.4. ProVerif. ProVerif [30, 32, 50–53] is a formal simulation tool for automatic verification of cryptographic protocols developed by Bruno Blanchet and based on the Dolev-Yao model. It can describe various cryptographic primitives such as public-key cryptography, shared key cryptography, and hash function, and the syntax used is easy to master. In this paper, we use the ProVerif tool to verify whether the proposed protocol has vulnerabilities. If there are vulnerabilities, the ProVerif tool will return an attack sequence. The specific operation is as follows.

Our protocol involves three parties communicating with the user, sensor, and gateway, in addition to using two channels, an encrypted channel and a public channel. The symbols, functions, and related definitions involved in ProVerif are described in Figure 4(a).

```
(********** channe **********)
free ch : channel. (*      *)
free sch : channel [private]. (*** secure channel, used for registering ***)
(********** shared keys **********)
free SKu : bitstring [private].
free SKs : bitstring [private].
free SKg : bitstring [private].
free IDi : bitstring [private].
(********** constants **********)
free x : bitstring [private].
(********** functions & reductions & equations *********)
fun h (bitstring ) : bitstring. (********* hash function *********)
fun mult (bitstring, bitstring) : bitstring. (** scalar multiplication operation **)
fun con (bitstring, bitstring) : bitstring. (**** concatenation operation ****)
reduc forall m : bitstring, n : bitstring; getmess (con(m,n)) = m.
fun xor (bitstring, bitstring) : bitstring. (****** XOR operation ******)
equation for all m : bitstring, n : bitstring; xor (xor(m,n), n) = m.
fun Gen (bitstring) : bitstring. (******* Generator operation *******)
fun Rep (bitstring, bitstring) : bitstring.
```

(a) Definitions

```
(********** queries **********)
query attacker (SKu).
query attacker (SKs).
query attacker (SKg).
query attacker (IDi).
query inj-event (UserAuthed ()) ==> inj-event (UserStarted ()).
query inj-event (SensorAcGWN ()) ==> inj-event (GWNAcUser ()).
query inj-event (GWNAcSensor ()) ==> inj-event (SensorAcGWN ()).
query inj-event (UserAcGWN ()) ==> inj-event (GWNAcSensor ()).
(********** event **********)
event UserStarted ().
event UserAuthed ().
event SensorAcGWN ().
event GWNAcUser ().
event GWNAcSensor ().
event UserAcGWN ().
```

(b) Events and queries

Figure 4: Definitions and queries.

The proposed protocol involves 6 events, namely, UserStarted(), UserAuthed(), SensorAcGWN(), GWNAcUser(), GWNAcSensor(), and UserAcGWN(), which, respectively, indicate that the user starts authentication, the user completes the authentication, the sensor completes the authentication to the gateway, the gateway completes the authentication to the user, the gateway completes the authentication to the sensor, and the user completes the authentication to the gateway. For the security of the proposed protocol, ProVerif will verify the user anonymity, the security of the session key, and the reasonableness of the authentication process. Figure 4(b) shows these events and queries.

Figure 5(a) shows the operations performed by the user and the sensor in the ProVerif. Figure 5(b) shows the operation of the gateway in the ProVerif. Figure 5(c) shows the results obtained after using the ProVerif tool to complete the verification. According to Figure 5(c), it is obvious that the proposed protocol can provide user anonymity and session key security, while the authentication process is executed in sequence.

6. Performance Comparison

In this section, we analyze the security and performance efficiency of the advanced protocol with that of [32, 35, 36].

6.1. Security Comparison. In Table 4, we demonstrate the security comparison. It is easy to see that our protocol is secure against well-known attacks. Das et al.'s protocol [32] cannot resist offline password guessing attacks and stolen smart card attacks. Meanwhile, their protocol does not provide perfect forward security and user anonymity. Although Chen et al.'s protocol [35] satisfies the last three vulnerabilities A5, A8, and A9, it still cannot resist the offline password guessing attacks. Wu et al.'s protocol [36] can resist offline password guessing attacks, but it is vulnerable to known session-specific temporary information attacks, impersonation attacks, and cannot provide perfect forward security and user anonymity.

6.2. Computational Cost Comparison. The performance is analyzed from the computation cost of protocols. Because

```
(************ User's process ***********)
let ProcessUser =
new IDi : bitstring; (********* the user's ID *******)
new PWi : bitstring; (****** the user's password ******)
new Bi : bitstring; (******* the user's biometric ******)
let (a: bitstring, b: bitstring) = Gen(Bi) in
let Pi = h (con (con (a,PWi), IDi)) in
let HIDi=h(con (a, IDi)) in
out (sch, (IDi,Pi,HIDi));
in (sch, (xA2:bitstring, xA3:bitstring));
! (event UserStarted (); let a = Rep (Bi,b) in
let Pi = h (con (con (a,PWi), IDi)) in
let HIDi = h (con (a, IDi)) in
let A3' = h (con (HIDi, Pi)) in
if A31 = A3 then
new N1:bitstring;
new T1:bitstring;
new SIDj:bitstring;
let A1 = xor (xA2, Pi) in
let C1 = xor (N1, h (con (A1, HIDi))) in
let C2 = xor (IDi, h (con (con (HIDi, A1), T1))) in
let C3 = xor (SIDj, h (con (con (A1, N1), T1))) in
let C4 = h (con (con (con (con (IDi, HIDi), SIDj) ,N1), T1)) in
out (ch, (HIDi, C1, C2, C3, C4, T1)); (***** authentication *****)
event UserAuthed ();
in (ch, (xC8:bitstring, xC10:bitstring, xC11:bitstring, xT4:bitstring));
let N2 = xor (xC10, h (con (con (con (A1, Pi), N1), xT4))) in
let N3 =xor(xC8, h (con (N1, N2))) in
let SKu = h (con (con (con (con (N1, N2), N3), HIDi), SIDA in
let C11' = h (con (con (con (con (SKu, A1), Pi), IDi), xT4)) in
if C11' = xC11 then event UserAcGWN ();
0).
(************ Sensor's process ***********)
let ProcessSensor = new SIDj:bitstring; new sj:bitstring;
out (sch, (SIDj, sj)); in (sch, (ysl :bitstring));
let SMj = xor (sj, ysl) in
Kin (ch, (yHIDi:bitstring, yC5:bitstring, K6:bitstring, yC7:bitstring,
yT2:bitstring));
let N2 = xor (yC5, h (con (con (SIDj, SMj), yT2))) in
let N1 = xor (yC6, h (con (SMj, N2))) in
let C7' = h (con (con (con (con (N1, N2), SIDj), SMj), yT2)) in
if C7' = yC7 then event SensorAcGWN ();
new N3:bistring:
new T3:bistring:
let SKs = h (con (con (con (con (N1, N2), N3), yHIDi), SIDj)) in
let C8 = xor (N3, h (con (N1, N2))) in
let C9 = h (con (con (con (SKs, SMj), SIDj), T3)) in
out (ch, (C8, C9, T3));
0).
```

(a) Process

```
(************ GWN's process ***********)
let UserReg =
in (sch, (zIDi:bitstring, zPi:bitstring, zHIDi:bitstring));
new gi:bitstring;
let A1 = h (con (con (con (gi, zHIDi), x), zIDi)) in
let A2 = xor (A1, zPi) in let A3 = h (con (zHIDi, zPi)) in
out (sch, (A2,A3));
0.

let SensorReg =
in (sch, (zSIDj:bitstring,zsj:bitstring));
let SMj = h (con (con (zSIDj, zsj), x)) in
let sl = xor (zsj, SMj) in
out (sch, (s1));
0.

let GWNAuth =
in (ch, (zHI Di :bitstring, zC1 :bitstring, zC2 :bitstring, zC3 :bitstring,
zC4:bitstring, zT1:bitstring));
new gi:bitstring; new zIDi:bitstring;
let A1 = h (con (con (con (gi,zHIDi), x), zIDi)) in
let IN = xor (zC2, h (con (con (zHIDi, A1), zT1))) in
if IDi ' = zIDi then let N1 = xor (zC1, h (con (A1, zHIDi))) in
let SIDj = xor (zC3, h (con (con (A1,N1), zT1))) in
let C4i = h (con (con (con (con (zIDi, zHIDi), SIDD, N1), zT1)) in
if C4' = zC4 then event GWNAcUser ();
new N2:bitstring;
new T2:bitstring;
new zsj:bitstring;
let SMj = h (con (con (SIDj, zsj), x)) in
let C5 = xor (N2, h (con (con (SIDj, SMj), T2))) in
let C6 = xor (N1, h (con (SMj, N2))) in
let C7 = h (con (con (con (con (N1, N2), SIDj), SMj), T2)) in
out (ch, (zHIDi, C5, C6, C7, T2));
in let C7 = h (con (con (con (con (N1, N2), SIDj), SMj), T2)) in
out (ch, (zHIDi, C5, C6, C7, T2 ));
in (ch, (zC8:bitstring, zC9:bitstring, zT3 :bitstring));
let N3 = xor (zC8, h (con (N1, N2))) in
let SKg = h (con (con (con (con (N1, N2), N3), zHIDi), SIDj)) in
let C9' = h (con (con (con (SKg, SMASIDD, zT3)) in
if C9' = zC9 then event GWNAcSensor ();
new T4:bitstring, new zPi :bitstring;
let C10 = xor (N2, h (con (con (con (A1, zPi), N1), T4))) in
let C11 = h (con (con (con (con (SKg, A1), zPi), IN), T4)) in
out (ch, (zC8, C10, C11, T4));
0.

let ProcessGWN = UserReg I SensorReg I GWNAuth.
```

(b) Process

Figure 5: Continued.

(********** results **********)

1-- RESULT not attacker (SKu[]) is true.

2-- RESULT not attacker (SKs[]) is true.

3-- RESULT not attacker (SKg[]) is true.

4-- RESULT not attacker (IDi[]) is true.

5-- RESULT inj-event (UserAuthed) ==> inj-event (UserStarted) is true.

6-- RESULT inj-event (SensorAcGWN) ==> inj-event (GWNAcUser) is true.

7-- RESULT inj-event (GWNAcSensor) ==> inj-event (SensorAcGWN) is true.

8-- RESULT inj-event (UserAcGWN) ==> inj-event (GWNAcSensor) is true.

(c) Results

Figure 5: Process and results.

Table 4: Security comparison.

	Das et al.'s protocol [32]	Chen et al.'s protocol [35]	Wu et al.'s protocol [36]	Our protocol
A1	√	√	×	√
A2	√	√	√	√
A3	√	√	×	√
A4	√	√	√	√
A5	× [34]	√	√	√
A6	× [36]	× [36]	√	√
A7	√	√	√	√
A8	× [34]	√	×	√
A9	× [34]	√	×	√

A1: known session-specific temporary information attacks; A2: user impersonation attacks; A3: sensor impersonation attacks; A4: man-in-the-middle attacks; A5: stolen smart card attacks; A6: off-line password guessing attacks; A7: privileged-insider attacks; A8: perfect forward secrecy; A9: user anonymity. The "√" denotes that this protocol can resist the attack. The "×" denotes that the protocol cannot resist the attack.

Table 5: Computational cost comparison.

	Das et al.'s protocol [32]	Chen et al.'s protocol [35]	Wu et al.'s protocol [36]	Our protocol
User	$T_f + 17T_h$	$T_f + T_s + 10T_h$	$T_f + 13T_h$	$T_f + 11T_h$
Gateway	$12T_h$	$2T_s + 3T_h$	$15T_h$	$14T_h$
Sensor	$9T_h$	$T_s + 5T_h$	$4T_h$	$6T_h$
Total	$T_f + 38T_h$	$T_f + 4T_s + 18T_h$	$T_f + 32T_h$	$T_f + 31T_h$

the computational cost of XOR and join operations is too small, it can be ignored in comparison. Here, compare the consumption of login authentication and the key exchange phase. T_f represents the time to execute a fuzzy extraction function. T_h represents the time to perform a hash operation. T_s represents the time to perform the symmetric encryption/decryption operation. Table 5 shows the computational cost comparison. The results show that the fuzzy extraction function T_f is used once in the total computational cost of each protocol. In addition, Das et al.'s protocol [32], Wu et al.'s protocol [36], and our protocol all use hash operations. However, our protocol has the least number of hash operations. Chen et al.'s protocol [35] not only performed 18 hashing operations but also performed four symmetric encryption/decryption operations, consuming $4T_s$. As we all know, the cost of symmetric encryption/decryption operation is very higher than the cost of hash operation. In other words, our improved protocol has a lower computational cost and provides higher security than previous protocols.

Table 6: Communication cost comparison.

	Rounds	Communication cost
Das et al.'s protocol [32]	3	1824 bits
Chen et al.'s protocol [35]	3	1248 bits
Wu et al.'s protocol [36]	4	2912 bits
Our protocol	4	2944 bits

6.3. Communication Cost Comparison. The performance is analyzed from the communication cost of protocols. We accept that the random number and identity are 160 bits, hash operation and the length of the ciphertext for symmetric encryption are 256 bits, and the timestamp is 32 bits.

In Das et al.'s protocol [32], the messages in the login and authentication phase are $Msg_1 = \{TID_i', X_i, Y_i, Z_i, T_1\}$, $Msg_2 = \{X_{gw}, Y_{gw}, Z_{gw}, T_2\}$, and $Msg_3 = \{V_j, W_j, T_3\}$, where TID_i' is an identity, $\{X_i, Y_i, X_{gw}, Y_{gw}, V_j\}$ belong to random strings, $\{Z_i, Z_{gw}, W_j\}$ are hash values, and $\{T_1, T_2, T_3\}$ are timestamps. The total communication cost of [32] is 1824 bits.

In Chen et al.'s protocol [35], the messages in the login and authentication phase are $\{M_1\}$, $\{M_2, N_g\}$, and $\{M_3, CDID_j, Ack, N_j\}$, where $\{M_1, M_2\}$ are ciphertexts, $\{N_g, CDID_j, Ack\}$ are random strings, and M_3 is a hash value. The total communication cost of [35] is 1248 bits.

In Wu et al.'s protocol [36], the messages in the authentication phase are $M_1 = \{PID_i, D_1, D_2, D_3, D_4, D_5, T_1\}$, $M_2 = \{D_6, D_7, ID_g, T_2\}$, $M_3 = \{D_8, D_9, T_3\}$, and $M_4 = \{D_8, D_{10}, D_{11}, D_{12}, T_4\}$, where $\{PID_i, ID_g\}$ are identities, $\{D_1, D_2, D_3, D_4, D_6, D_8, D_{10}, D_{11}\}$ are random strings, $\{D_5, D_7, D_9, D_{12}\}$ are hash values, and $\{T_1, T_2, T_3, T_4\}$ are timestamps. The total communication cost of [36] is 2912 bits.

In our protocol, the messages in the authentication phase are $M_1 = \{HID_i, C_1, C_2, C_3, C_4, T_1\}$, $M_2 = \{HID_i,$

$C_5, C_6, C_7, T_2\}$, $M_3 = \{C_8, C_9, T_3\}$, and $M_4 = \{C_8, C_{10}, C_{11}, T_4\}$, where $\{C_1, C_2, C_3, C_5, C_6, C_8, C_{10}\}$ are random strings, $\{HID_i, C_4, C_7, C_9, C_{11}\}$ are hash values, and $\{T_1, T_2, T_3, T_4\}$ are timestamps. The total communication cost of our protocol is 2944 bits. The communication cost comparison is shown in Table 6.

According to Table 6, we can see that the number of rounds of Das et al.'s and Chen et al.'s protocol is less than the one of Wu et al.'s and our protocol. It is obvious that the communication cost of the first two protocols is lower. However, in Table 5, it can be seen that the computational costs of the first two protocols are relatively high. Although our protocol has a slightly higher communication cost than [36], the efficiency in practical application is almost the same. Furthermore, in Table 4, Wu et al.'s protocol [36] cannot resist known session-specific temporary information attacks and impersonation attacks and cannot provide perfect forward security and user anonymity.

7. Conclusion

In this paper, we have described the protocol of Wu et al. and found that their protocol was unable to resist known session-specific temporary information attacks, violated perfect forward and backward security, and could not provide user anonymity. In order to solve the vulnerabilities, we proposed a provably secure three-factor authentication protocol, which is proved to be secure by formal and informal security analysis, and the BAN logic, and the ProVerif tool. Finally, through the comparison of performance and security, our protocol can better ensure security and efficiency. In future work, we will work to further improve the security and performance of protocols in wireless sensors.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Z. Meng, J.-S. Pan, and K.-K. Tseng, "Pade: an enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization," *Knowledge-Based Systems*, vol. 168, pp. 80–99, 2019.

[2] X. Xue, C. Yang, C. Jiang, P.-W. Tsai, G. Mao, and H. Zhu, "Optimizing ontology alignment through linkage learning on entity correspondences," *Complexity*, vol. 2021, Article ID 5574732, 12 pages, 2021.

[3] J.-S. Pan, N. Liu, S.-C. Chu, and T. Lai, "An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems," *Information Sciences*, vol. 561, pp. 304–325, 2021.

[4] X. Xue, X. Wu, C. Jiang, G. Mao, and H. Zhu, "Integrating sensor ontologies with global and local alignment extractions," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6625184, 10 pages, 2021.

[5] J. Zhang, H. Nian, X. Ye, X. Ji, and Y. He, "A spatial correlation based partial coverage scheduling scheme in wireless sensor networks," *Journal of Network Intelligence*, vol. 5, no. 2, pp. 34–43, 2020.

[6] Z. Wang, L. Gong, J. Yang, and X. Zhang, "Cloud-assisted elliptic curve password authenticated key exchange protocol for wearable healthcare monitoring system," *Concurrency and Computation: Practice and Experience*, article e5734, 2020.

[7] Y. Wang, Y. Liu, H. Ma, Q. Ma, and Q. Ding, "The research of identity authentication based on multiple biometrics fusion in complex interactive environment," *Journal of Network Intelligence*, vol. 4, no. 4, pp. 124–139, 2019.

[8] W. Zeng and J. Zhang, "Leakage-resilient and lightweight authenticated key exchange for e-health," in *2020 6th International Conference on Information Management (ICIM)*, pp. 162–166, London, UK, 2020.

[9] J. M.-T. Wu, G. Srivastava, A. Jolfaei, P. Fournier-Viger, and J. C.-W. Lin, "Hiding sensitive information in ehealth datasets," *Future Generation Computer Systems*, vol. 117, pp. 169–180, 2020.

[10] H. Xiong, Y. Zhao, Y. Hou et al., "Heterogeneous signcryption with equality test for IIoT environment," *IEEE Internet of Things Journal*, p. 1, 2020.

[11] H. Xiong, Y. Wu, C. Jin, and S. Kumari, "Efficient and privacy preserving authentication protocol for heterogeneous systems in IIoT," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11713–11724, 2020.

[12] J.-S. Pan, X.-X. Sun, S.-C. Chu, A. Abraham, and B. Yan, "Digital watermarking with improved sms applied for qr code," *Engineering Applications of Artificial Intelligence*, vol. 97, p. 104049, 2021.

[13] T.-Y. Wu, T. Wang, Y.-Q. Lee, W. Zheng, S. Kumari, and S. Kumar, "Improved authenticated key agreement scheme for fog-driven IoT healthcare system," *Security and Communication Networks*, vol. 2021, Article ID 6658041, 16 pages, 2021.

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

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

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

[17] B. Vaidya, D. Makrakis, and H. T. Mouftah, "Improved two-factor user authentication in wireless sensor networks," in *2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications*, pp. 600–606, Niagara Falls, ON, Canada, 2010.

[18] B. Vaidya, D. Makrakis, and H. Mouftah, "Two-factor mutual authentication with key agreement in wireless sensor networks," *Security and Communication Networks*, vol. 9, no. 2, 183 pages, 2016.

[19] J. Kim, D. Lee, W. Jeon, Y. Lee, and D. Won, "Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks," *Sensors*, vol. 14, no. 4, pp. 6443–6462, 2014.

[20] H.-F. Huang, Y.-F. Chang, and C.-H. Liu, "Enhancement of two-factor user authentication in wireless sensor networks,"

in *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 27–30, Darmstadt, Germany, 2010.

[21] P. Kumar, M. Sain, and H. J. Lee, "An efficient two-factor user authentication framework for wireless sensor networks," in *13th International Conference on Advanced Communication Technology (ICACT2011)*, pp. 574–578, Gangwon, Korea (South), 2011.

[22] F. Wang, Y. Zhang, Y. Xu, L. Wu, and B. Diao, "A dos-resilient enhanced two-factor user authentication scheme in wireless sensor networks," in *2014 International Conference on Computing, Networking and Communications (ICNC)*, pp. 1096–1102, Honolulu, HI, USA, 2014.

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

[24] J. Yuan, C. Jiang, and Z. Jiang, "A biometric-based user authentication for wireless sensor networks," *Wuhan University Journal of Natural Sciences*, vol. 15, no. 3, pp. 272–276, 2010.

[25] K. H. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," in *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, vol. 1, p. 8, Taichung, Taiwan, 2006.

[26] E.-J. Yoon and K.-Y. Yoo, "A new biometric-based user authentication scheme without using password for wireless sensor networks," in *2011 IEEE 20th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 279–284, Paris, France, 2011.

[27] O. Althobaiti, M. Al-Rodhaan, and A. Al-Dhelaan, "An efficient biometric authentication protocol for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 9, no. 5, Article ID 407971, 2013.

[28] A. K. Das, "A secure and efficient user anonymity-preserving Three-Factor authentication protocol for large-scale distributed wireless sensor networks," *Wireless Personal Communications*, vol. 82, no. 3, pp. 1377–1404, 2015.

[29] A. K. Das, "A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor," *International Journal of Communication Systems*, vol. 30, no. 1, article e2933, 2017.

[30] A. K. Maurya and V. N. Sastry, "Fuzzy extractor and elliptic curve based efficient user authentication protocol for wireless sensor networks and internet of things," *Information*, vol. 8, no. 4, p. 136, 2017.

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

[32] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. Rodrigues, "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4900–4913, 2018.

[33] J. Ryu, H. Lee, H. Kim, and D. Won, "Secure and efficient three-factor protocol for wireless sensor networks," *Sensors*, vol. 18, no. 12, p. 4481, 2018.

[34] S. Hussain and S. A. Chaudhry, "Comments on "biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment"," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10936–10940, 2019.

[35] Y. Chen, Y. Ge, Y. Wang, and Z. Zeng, "An improved three-factor user authentication and key agreement scheme for wireless medical sensor networks," *IEEE Access*, vol. 7, pp. 85440–85451, 2019.

[36] F. Wu, X. Li, L. Xu, P. Vijayakumar, and N. Kumar, "A novel three-actor authentication protocol for wireless sensor networks with IoT notion," *IEEE Systems Journal*, vol. 15, pp. 1120–1129, 2020.

[37] M. F. Ayub, S. Shamshad, K. Mahmood, S. H. Islam, R. M. Parizi, and K.-K. R. Choo, "A provably secure two-factor authentication scheme for USB storage devices," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 4, pp. 396–405, 2020.

[38] B. A. Alzahrani, S. A. Chaudhry, A. Barnawi, A. Al-Barakati, and T. Shon, "An anonymous device to device authentication protocol using ECC and self certified public keys usable in internet of things based autonomous devices," *Electronics*, vol. 9, no. 3, p. 520, 2020.

[39] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 426, no. 1871, pp. 233–271, 1989.

[40] R. Madhusudhan, M. Hegde, and I. Memon, "A secure and enhanced elliptic curve cryptography-based dynamic authentication scheme using smart card," *International Journal of Communication Systems*, vol. 31, no. 11, 2018.

[41] T.-Y. Wu, Z. Lee, L. Yang, J.-N. Luo, and R. Tso, "Provably secure authentication key exchange scheme using fog nodes in vehicular ad hoc networks," *The Journal of Supercomputing*, 2021.

[42] O. Goldreich and S. Halevi, "The random oracle methodology, revisited," in *Proc. 30th ACM Symp. Theory of Computing*, pp. 209–218, Dallas, TX, USA, 1998.

[43] K.-H. Wang, C.-M. Chen, W. Fang, and T.-Y. Wu, "A secure authentication scheme for internet of things," *Pervasive and Mobile Computing*, vol. 42, pp. 15–26, 2017.

[44] C.-T. Li, T.-Y. Wu, C.-L. Chen, C.-C. Lee, and C.-M. Chen, "An efficient user authentication and user anonymity scheme with provably security for IoT-based medical care system," *Sensors*, vol. 17, no. 7, p. 1482, 2017.

[45] S. Banerjee, V. Odelu, A. K. Das et al., "A provably secure and lightweight anonymous user authenticated session key exchange scheme for internet of things deployment," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8739–8752, 2019.

[46] D. Abbasinezhad-Mood, S. M. Mazinani, M. Nikooghadam, and A. O. Sharif, "Efficient provably-secure dynamic ID-based authenticated key agreement scheme with enhanced security provision," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.

[47] J.-C. Hsu, Y.-S. Jheng, S. M. M. Rahman, and R. Tso, "Password-based authenticated key exchange from lattices for client server model," *Journal of Computer Security and Data Forensics*, vol. 1, no. 1, pp. 1–17, 2021.

[48] T.-Y. Wu, Y.-Q. Lee, C.-M. Chen, Y. Tian, and N. A. Al-Nabhan, "An enhanced pairing-based authentication scheme for smart grid communications," *Journal of Ambient Intelligence and Humanized Computing*, 2021.

[49] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[50] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.

[51] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, "ProVerif 2.02 pl1: automatic cryptographic protocol verifier," *User Manual and Tutorial*, 2020, https://prosecco.gforge.inria.fr/personal/bblanche/proverif/manual.pdf.

[52] T.-Y. Wu, Z. Lee, M. S. Obaidat, S. Kumari, S. Kumar, and C.-M. Chen, "An authenticated key exchange protocol for multi-server architecture in 5G networks," *IEEE Access*, vol. 8, pp. 28096–28108, 2020.

[53] T.-Y. Wu, L. Yang, Z. Lee, C.-M. Chen, J.-S. Pan, and S. H. Lslam, "Improved ECC-based three-factor multiserver authentication scheme," *Security and Communication Networks*, vol. 2021, Article ID 6627956, 14 pages, 2021.