
Tsu-Yang Wu,1 Lei Yang,1 Zhiyuan Lee,1 Shu-Chuan Chu,1 Saru Kumari,2 and Sachin Kumar3

1College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
2Department of Mathematics, Chaudhary Charan Singh University, Meerut, Uttar Pradesh 250004, India
3Department 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

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.
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 into 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-verifyer 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 $S_i$ selects its own identity $SID_i$ and sends $SID_i$ to gateway node $GW_i$. Then, $GW_i$ selects $x$ as the master key and computes $S = M_j = h(SID_j||x)$. Finally, $GW_i$ sends $SM_j$ to $S_j$.

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

2.2. Authentication and Key Exchange. $U_j$ inserts SC and enters $ID_j$, $PW_i$, and $B_j$. Then, the smart card selects $N_j$, $T_j$ and computes $C_0 = H(B_j), r_0 = C_j + h(ID_j||PW_j||C_0), P_j = h(C_0 || PW_j||r_0), B_1 = C_1 + h(ID_i||P_i), B_2 = C_2 + h(ID_j||PW_j), D_1 = B_1 + h(PID_i||N_j||T_j), D_2 = SCN_i + h(ID_i||PW_j||C_0)$. Then, $U_j$ stores $\{C_1, C_2, C_3\}$ to SC and deletes $\{B_i, B_j\}$ from SC. Note that, all communications in this phase are based on a secure channel.

$GW_i$ first checks whether $T_1$ is valid. If it times out, the request is terminated. Otherwise, $GW_i$ calculates $B_3 = h(PID_i||x), N_j = D_3 + B_1$, $ID_j = D_2 + h(PID_i||N_j||T_j)$ and then searches for $ID_j$ in its database. If it is not found, terminates. Otherwise, $GW_i$ computes $SCN_i = D_3 + h(ID_j||N_j||T_j)$, $B_3 = h(SCN_i||x), SED_i = D_2 + h(B_j||N_j||T_j)$, and verifies $D_3 = h(ID_j||PID_i||SCN_i||N_j||SID_j)$. If the verification holds, $GW_i$ selects $T_2$ and calculates $SM_j = h(SID_j||x), D_4 = N_j + h(ID_i||SM_i||T_j)$, and $D_5 = h(N_j||SM_i||SID_j)$. Finally, $GW_i$ sends $M_2 = \{D_4, D_5, ID_j, T_j\}$ to $SA$.

$S_j$ first checks whether $T_2$ is valid. If it times out, the communication is terminated. Otherwise, $S_j$ calculates $N_j = D_6 + h(ID_g||SM_i||T_j)$ and verifies $D_6 = h(N_j||SM_i||SID_j)$. If the verification holds, $S_j$ selects $T_3$ and computes $SK_g = h(N_j||N_2), D_7 = N_j + N_2$, and $D_8 = h(SK_g||SM_i||ID_g||SID_j||T_j)$. Finally, $S_j$ sends $M_3 = \{D_7, D_8, T_3\}$ to $GW_i$.

$GW_i$ first checks whether $T_3$ is valid. If it times out, the communication is terminated. Otherwise, $GW_i$ calculates $N_2 = D_8 \oplus N_1$, $SK_g = h(N_1||N_2)$, and verifies $D_8 = h(SK_g||SM_i||ID_i||SID_i||T_3)$. If the verification holds, $GW_i$ selects $T_4$, $PID_{new}$ and computes $B_1 = h(PID_{new}||x)$. Then, $GW_i$ sends $M = \{B_1, D_9, D_{10}, T_4\}$ to $U_j$.

$U_j$ first checks whether $T_4$ is valid. If it times out, the communication is terminated. Otherwise, $U_j$ calculates $N_2 = D_8 \oplus N_1$, $B_{1new} = D_2 \oplus h(B_1||N_1||T_4)$, $PID_{new} = D_{11} \oplus h(B_{1new}||N_2||T_4)$, and $SK_u = h(N_1||N_2)$, and verifies $D_2 = h(SK_u||B_{1new}||PID_{new}||B_1||ID_i||SID_i)$. Finally, $GW_i$ sends $M_4 = \{D_8, D_9, D_{10}, D_{11}, T_4\}$ to $U_j$.

After finish the above steps, $U_j$, $GW_i$, and $S_j$ can establish a session $SK = SK_u = SK_g = h(N_1||N_2)$ to communicate. Note that, $B_{1new}$ and $PID_{new}$ are used in the next section.

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.

Table 1: Symbols and descriptions.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_j$</td>
<td>User</td>
</tr>
<tr>
<td>$S_j$</td>
<td>Sensor</td>
</tr>
<tr>
<td>$GW_i$</td>
<td>Gateway</td>
</tr>
<tr>
<td>$\mathcal{A}$</td>
<td>Adversary</td>
</tr>
<tr>
<td>$SA$</td>
<td>System administrator</td>
</tr>
<tr>
<td>$SC$</td>
<td>Smart card</td>
</tr>
<tr>
<td>$SK$</td>
<td>Session key</td>
</tr>
<tr>
<td>$x, ID_g$</td>
<td>$GW_i$’s master key and identity</td>
</tr>
<tr>
<td>$t_j$</td>
<td>$S_j$’s secret value</td>
</tr>
<tr>
<td>$ID_i, PW_i, B_i$</td>
<td>$U_j$’s identity, password, and biometrics</td>
</tr>
<tr>
<td>$T_i$</td>
<td>Timestamp</td>
</tr>
<tr>
<td>$Gen(\cdot)$</td>
<td>Fuzzy generator function</td>
</tr>
<tr>
<td>$Rep(\cdot)$</td>
<td>Fuzzy reproduction function</td>
</tr>
<tr>
<td>$h(\cdot)$</td>
<td>Hash function</td>
</tr>
</tbody>
</table>
(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 $M_1 = \{\text{PID}_1, D_1, D_2, D_3, D_4, T_1\}$, $\mathcal{A}$ can recover

$$B_1 = h(\text{PID}_1||x), \ N_1 = D_1 \oplus B_1, \ ID_1 = D_2 \oplus h(\text{PID}_1||N_1||T_1), \ \text{SCN}_1 = D_3 \oplus h(\text{ID}_1||N_1||T_1), \ B_2 = h(\text{SCN}_1||x), \ SID_j = D_4 \oplus h(B_2||N_1||T_1), \ \text{and} \ S \ M_j = h(\text{SID}_j||x).$$

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 $M_2 = \{D_6, D_7, ID_2, T_2\}$ and selects a random number $N_2'$ and timestamp $T_A$. Then, $\mathcal{A}$ computes

$$SK'_A = h(N_1||N_2'), \ D'_1 = N_1 \oplus N_2', \ D'_2 = h(SK'_A||SM_j||ID_2||SID_2||T_A) \ \text{and sends} \ M'_2 = \{D'_6, D'_7, T_A\} \ \text{to GWN}.$$

(2) GWN checks whether $T_A$ is valid. If it times out, the communication is terminated. Otherwise, GWN calculates

$$N_2'' = D'_2 \oplus N_1, \ SK_2 = h(N_1||N_2''), \ \text{and verifies} \ D'_2 = h(SK_2||SM_j||ID_2||SID_2||T_A).$$

The following steps are similar to the authentication phase in Section 2.2 except $D''_{11} = PID_{\text{new}} \oplus h(D''_{11}||N_2''||T_A)$. Then, GWN sends $M''_4 = \{D''_6, D''_7, D''_{11}, T_{12}, T_2\}$ to $U_i$.

(3) $U_i$ checks whether $T_4$ is valid. If it times out, the communication is terminated. Otherwise, $U_i$ calculates

$$N_2'' = D''_2 \oplus N_1, \ B''_{\text{new}} = D''_6 \oplus h(B_2||N_1||T_A), \ PID'_\text{new} = D''_{11} \oplus h(B''_{\text{new}}||N_2''||T_A), \ SK''_2 = h(N_1||N_2''), \ \text{and verifies} \ D''_{12} = h(SK''_2||B''_{\text{new}}||PID'_\text{new}||B_1||SID_1||T_A).$$

It is easy to see that the result is true.

Thus, $U_i$ believes that he can establish a session key $SK = SK''_2 = SK_A = h(N_1||N_2'')$ 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

$$ID_1 = D_1, \ B_1 = h(\text{PID}_1||N_1||T_1) \ \text{and} \ SK = h(N_1||N_2), \ \text{where} \ B_1 = h(\text{PID}_1||x), \ N_1 = D_1 \oplus B_1, \ N_2 = D_2 \oplus N_1.$$ 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 $N_2 = D_9 \oplus N_1$. Furthermore, $\mathcal{A}$ can compute update values $B''_{\text{new}} = D_{10}$ \ $\oplus h(B_1||N_1||T_4)$ and $\text{PID}'_{\text{new}} = D_{11} \oplus h(B''_{\text{new}}||N_2||T_4)$ 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 $N'_1 = B''_{\text{new}} \oplus D'_1, \ N'_1 = N_1 \oplus D'_1$. The session key $SK'$ can be computed by $SK' = h(N'_1||N'_2)$. Meanwhile, the newest updated values $B''_{\text{new}} = D'_{10} \oplus h(B''_{\text{new}}||N'_1||T'_4), \ \text{PID}'_{\text{new}} = D'_{11} \oplus h(B''_{\text{new}}||N'_2||T'_4)$ 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 does 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, $G WN$ calculates $SM_j = h(\text{SID}_j||s_j||x), \ s_j = s_j \oplus SM_j$, and stores $s_j$ in its database. Finally, $G WN$ sends $s_j$ to $S_j$. After receiving $s_j$, $S_j$ computes $SM_j = s_j \oplus s_j$ and stores it in its memory.

User Registration. $U_i$ selects $ID_j, PW_i$, and inputs his $B_i$ to compute $P_i = h(\sigma_i||PW_i||ID_j)$ and $HID_i = h(ID_j||\tau_i)$, where $Gen(B_i) = (\sigma_i, \tau_i)$. Then, $U_i$ sends $\{ID_j, PW_i, HID_i\}$ to $S_j$ via a secure channel. After receiving $\{ID_j, PW_i, HID_i\}$, $S_j$ checks whether $ID_j$ exits its database. If so, deleting the relevant records in the database and reregister. Otherwise, $S_j$ selects $g_j$ and computes $A_j = h(g_j||HID_i||x||ID_j), \ A_j = A_j \oplus P_i$, and $A_j = h(HID_i||P_i)$. Then, $SA$ stores $\{A_j, A_j\}$ in $SC$ and sends $SC$ to $U_i$ via a secure channel. Meanwhile, $\{HID_i, ID_j, g_j\}$ is stored in $SA$’s database. After receiving $SC$, $U_i$ stores $\tau_i$ 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 $ID_j, PW_i$, and $B_i$. Then, $U_i$ can compute $\sigma_i = \text{Rep}(B_i, \tau_i)$, $P_i = h(\sigma_i||PW_i||ID_j)$, $HID_i = h(ID_j||\sigma_i), \ A_j = h(HID_i||P_i)$ to check whether $A_j$ is equal to $A_j$. If the verification holds, $U_i$ generates $N_1, T_1$, and computes $A_1 = A_j \oplus P_i, \ C_1 = N_1 \oplus h(A_1||HID_i), \ C_2 = ID_j \oplus h(HID_i||A_1||T_1), \ C_3 = SID_j \oplus h(A_1||N_1||T_1), \ C_4 = h(ID_j||HID_i||SID_j||N_1||T_1)$.

Finally, $U_i$ sends $M_3 = \{HID_i, C_1, C_2, C_3, C_4, T_j\}$ to $G WN$.

Upon receiving $M_3$, GWN first checks whether $T_j$ is valid. If the times out, the communication is terminated. Otherwise, GWN according to $HID_i$ finds the corresponding $\{ID_j, g_j\}$ in its database and computes $A_j = h(g_j||HID_i||x||ID_j)$, $ID_j' = C_2 \oplus h(HID_i||A_1||T_1)$. Then, GWN checks whether $ID_j'$ equals to $ID_j$. If not, the session is terminated. Otherwise, GWN computes $N_1 = C_1 \oplus h(A_1||HID_i), \ SID_j = C_4 \oplus h(A_1||N_1||T_1)$ and verifies $C_1 = D''_{11} \oplus h(ID_j'\oplus h(SID_j||N_1||T_1).$ If the verification holds, $G WN$ generates $N_2, T_2$, and computes $SM_j = h(\text{SID}_j||s_j||x), \ C_5 = N_2 \oplus h(\text{SID}_j||SM_j||T_2), \ C_6 = N_1 \oplus h(SM_j||N_2), \ C_7 = h(N_1||N_2||SID_j||SM_j||T_2).$ Finally, $G WN$ sends $M_3 = \{HID_i, C_1, C_2, C_3, C_4, C_6, T_2\}$ to $S_j$. 


Upon receiving $M_2$, $S_j$ first checks whether $T_2$ is valid. If the time out, the communication is terminated. Otherwise, $S_j$ computes $N_2 = C_5 \oplus h(SID_j)$ and verifies $C_2 = h(N_1 || T_2)$ and computes $N_1 = C_4 \oplus h(SM_j)$ and verifies $C_4 = h(N_3 || T_2)$ if the verification holds, $S_j$ computes $T_3$ and computes $SK_j = h(N_1 || N_2 || N_3 || SID_j)$, finally, $S_j$ sends $M_3 = \{C_{10}, C_{11}, T_3\}$ to $GWN$.

Upon receiving $M_3$, $GWN$ first checks whether $T_3$ is valid. If times out, the communication is terminated. Otherwise, $GWN$ computes $N_5 = C_6 \oplus h(N_1 || N_2 || N_3 || HID_j || SID_j)$ and verifies $C_5 = h(SK_j || SM_j || SID_j || T_3)$. 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 the time out, the communication is terminated. Otherwise, $U_i$ computes $N_2 = C_{10} \oplus h(A_i || P_i || N_1 || T_4)$, $SK_u = h(N_1 || N_2 || N_3 || HID_j || SID_j)$ and verifies $C_{11} = h(SK_u || A_i || P_i || ID_i || T_4)$. If the verification holds, $SK_u$ is set as a session key to be used 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) $R_1$ (Message meaning (M-M) rule): $(P \equiv P \Rightarrow Q, P \iff Q)$

(ii) $R_2$ (Nonce verification (N-V) rule): $(P \equiv \#(X), P \equiv Q | \Rightarrow X)/(P \equiv Q | \equiv X)$

(iii) $R_3$ (Jurisdiction rule): $(P \equiv Q | \Rightarrow X, P \equiv Q | \equiv X)/(P \equiv X)$

(iv) $R_4$ (Session key (S-K) rule): $(P \equiv \#(X), P \equiv Q | \equiv X)/(P \equiv P \iff Q)$

5.1.2. Goals

(i) $G_1$: $U_i \equiv U_i \iff SK_j$

(ii) $G_2$: $S_j \equiv S_j \iff SK_j$

(iii) $G_3$: $GWN \equiv U_i \iff SK_j$

(iv) $G_4$: $U_i \equiv S_j \iff U_i \iff SK_j$

(v) $G_5$: $S_j \iff U_i \equiv U_i \iff SK_j$

(vi) $G_6$: $GWN \equiv U_i \equiv U_i \iff SK_j$

(vii) $G_7$: $GWN \equiv S_j \equiv U_i \iff SK_j$

5.1.3. Idealize the Communication Messages

(i) $M_1$: $U_i \rightarrow GWN$: \{HID, C_1, C_2, C_3, C_4, T_1\}$

(ii) $M_2$: $GWN \rightarrow S_j$: \{HID, C_5, C_6, C_7, T_2\}$

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

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

(v) $M_5$: $S_j \rightarrow U_i$: \{C_4\}$

5.1.4. Initial Assumptions

(i) $A_1$: $U_i \equiv \#(N_1)$

(ii) $A_2$: $S_j \equiv \#(N_1)$

(iii) $A_3$: $GWN \equiv \#(N_2)$
6 Wireless Communications and Mobile Computing

(iii) $A_4: \text{GWN} \Rightarrow U_1 \Rightarrow \text{HID}_i$

(v) $A_5: \text{GWN} \Rightarrow U_1 \Rightarrow \#(N_1)$

(vi) $A_6: \text{GWN} \Rightarrow \#(N_1)$

(vii) $A_7: \text{GWN} \Rightarrow U_1 \Rightarrow N_1$

(viii) $A_8: \text{GWN} \Rightarrow \#(\text{SID}_j)$

(ix) $A_9: \text{GWN} \Rightarrow U_1 \Rightarrow \text{SID}_j$

(x) $A_{10}: \text{S}_j \Rightarrow \text{GWN} \Rightarrow \text{HID}_i$

(xi) $A_{11}: \text{S}_j \Rightarrow \text{GWN} \Rightarrow \text{SM}_j$

(xii) $A_{12}: \text{S}_j \Rightarrow \#(\text{SM}_j)$

(xiii) $A_{13}: \text{S}_j \Rightarrow \#(N_2)$

(xiv) $A_{14}: \text{S}_j \Rightarrow \text{GWN} \Rightarrow N_2$

(xv) $A_{15}: \text{S}_j \Rightarrow \#(N_1)$

(xvi) $A_{16}: \text{GWN} \Rightarrow S_j \Rightarrow h(N_1, N_2) \Rightarrow \text{GWN}$

(xvii) $A_{17}: \text{GWN} \Rightarrow \#(N_3)$

(xviii) $A_{18}: \text{GWN} \Rightarrow S_j \Rightarrow N_3$

(xix) $A_{19}: U_1 \Rightarrow \text{GWN} \Rightarrow A_1$

(xx) $A_{20}: U_1 \Rightarrow \#(A_1)$

(xxi) $A_{21}: U_1 \Rightarrow \#(N_2)$

(xxii) $A_{22}: U_1 \Rightarrow \text{GWN} \Rightarrow N_2$

(xxiii) $A_{23}: U_1 \Rightarrow U_1 \Rightarrow h(N_1, N_2) \Rightarrow S_j$

(xxiv) $A_{24}: U_1 \Rightarrow \#(N_3)$

(xxv) $A_{25}: U_1 \Rightarrow S_j \Rightarrow N_3$

(xxvi) $A_{26}: S_j \Rightarrow \text{GWN} \Rightarrow N_1$

5.1.5. The Proof of our Proposed Protocol via BAN Logic. By $M_1$, we have $S_1: \text{GWN} \Leftarrow (\text{HID}_i, C_1: N_1, \text{HID}_{A_1}, C_2, C_3: \text{SID}_j, C_4, T_1)$ and further $S_2: \text{GWN} \Rightarrow U_1 \Rightarrow \text{HID}_i$. Base on $A_4, S_2, T_1$ (Jurisdiction rule), we can obtain $S_3: \text{GWN}$.
According to $S_1$, it implies $S_4: GWN \equiv N_1, HID_{A_i}$. By $A_5, S_4,$ and $R_1 (M-M$ rule), it implies $S_5: GWN \equiv U_i \rightarrow \neg (N_1, HID)$. By $A_6, S_5$, and $R_2 (N-V$ rule), we can obtain $S_6$: $GWN \equiv U_i \equiv N_1$. According to $A_7, S_6$, and $R_3$ (Jurisdiction rule), it implies $S_7: GWN \equiv N_2$. According to $S_1$, we have $S_8: GWN \equiv SJD_i$. By $A_5, S_8$, and $R_1 (M-M$ rule), it implies $S_9: GWN \equiv U_i \rightarrow \neg SJD_i$. By $A_8, S_9$, and $R_2$ (N-V rule), we can obtain $S_{10}: GWN \equiv U_i \equiv SJD_i$. According to $A_9, S_{10}$, and $R_3$ (Jurisdiction rule), it implies $S_{11}: GWN \equiv SJD_i$.

By $M_4$, we have $S_{12}: S_j \leftarrow \{HID_j, C_6: N_{2h(SJD_j)}(S|T_j), C_6: N_{1b(SJD_j)}(S|T_j), C_7, T_2\}$ and further $S_{13}: S_j \equiv GWN \equiv HID_j$. Based on $A_{10}, S_{13}$, and $R_3$ (Jurisdiction rule), we can obtain $S_{14}: S_j \equiv HID_{D_j}$. By $A_{11}, A_{12}$, and $R_4 (S-K$ rule), it implies $S_{15}: S_j \equiv S_j \equiv h(SJD_j)||T_j) \equiv GWN$. According to $S_{12}$, we have $S_{16}: S_j \equiv S_j \equiv N_1$. Based on $A_{26}, S_{23}$, and $R_3$ (Jurisdiction rule), it implies $S_{24}: S_j \equiv N_2$. Since $SK = h(N_1 || N_2 || HID_j) || SJD_i$, $S_{25}: S_j \equiv U_i \leftarrow SK S_j$ is obtained. (G2) According to $A_{22}, S_{25}$, and $R_4 (S-K$ rule), we can obtain $S_{26}: GWN \equiv U_i \equiv ^{H_{t}} \equiv GWN$. (G5)

By $M_5$, we have $S_{27}: GWN \equiv \{C_8: N_{3b(SJD_j)}(S|T_j), C_9, T_3\}$ and further $S_{28}: GWN \equiv N_{3b(SJD_j)}(S|T_j)$. Based on $A_{16}, S_{28}$, and $R_1 (M-M$ rule), we can obtain $S_{29}: GWN \equiv S_j \equiv N_3$. By $A_{17}, S_{29}$, and $R_2 (N-V$ rule), we can obtain $S_{30}: GWN \equiv S_j \equiv N_3$. Based on $A_{18}, S_{30}$, and $R_3$ (Jurisdiction rule), we can obtain $S_{31}: GWN \equiv N_2$. According to $S_{22}, S_{31}$, and $S_{31}$, it implies $S_{32}: GWN \equiv U_i \leftarrow ^{SK} S_j$. (G3) Based on $A_{6}, S_{32}$, and $R_4 (S-K$ rule), we can obtain $S_{33}: GWN \equiv U_i \equiv U_i \leftarrow ^{SK} S_j$. (G6) According to $A_{27}, S_{32}$, and $R_4 (S-K$ rule), it implies $S_{34}: GWN \equiv S_j \equiv U_i \leftarrow ^{SK} S_j$. (G7)

By $M_4$, we have $S_{35}: U_j \leftarrow \{C_{10}: N_{2h(A_{i}||P_{i})}(S|T_i), C_{11}, T_4\}$. Based on $A_{19}, A_{20}$, and $R_4 (S-K$ rule), we can obtain $S_{36}: U_j \equiv U_i \equiv \{h(A_{i}||P_{i})||T_i\} \equiv GWN$. According to $S_{35}$, we have $S_{37}: U_j \equiv N_{2h(A_{i}||P_{i})}||T_i \equiv GWN$. Based on $S_{36}, S_{37}$, and $R_1 (M-M$ rule), it implies $S_{38}: U_j \equiv GWN \equiv N_2$. By $A_{21}, S_{38}$, and $R_2 (N-V$ rule), we can obtain $S_{39}: U_j \equiv GWN \equiv N_2$. According to $A_{22}, S_{39}$, and $R_3$ (Jurisdiction rule), it implies $S_{40}: U_j \equiv N_2$.

By $M_4$, we have $S_{41}: U_j \leftarrow \{C_8: N_{1b(SJD_j)}(S|T_j)\}$. Based on $A_{23}, S_{41}$, and $R_1 (M-M$ rule), it implies $S_{42}: U_j \equiv S_j \equiv N_2$. By $A_{24}, S_{42}$, and $R_2 (N-V$ rule), we can obtain $S_{43}: U_j \equiv S_j \equiv N_2$. Based on $A_{25}, S_{43}$, and $R_3$ (Jurisdiction rule), it implies $S_{44}: U_j \equiv N_2$. According to $S_{40}$ and $S_{44}$, we can obtain $S_{45}: U_j \equiv U_i \leftarrow ^{SK} S_j$. (G1) According to $A_{24}$ and $S_{45}$, we can obtain $S_{45}: U_j \equiv S_j \equiv U_i \leftarrow ^{SK} S_j$. (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, $U_j, S_j,$ and $GW$. We use $T^2_{GM}$ and $T^3_{GW}$ to represent the xth instance of $U_i$, the yth instance of $S_j$, and the zth instance of $GW$, respectively. Here, we define that adversary $\mathcal{A}$ has the ability to initiate the following query. Note that, $\mathcal{O} = \{T^2_{GM}, T^3_{GW}\}$.

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

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

(iii) $\mathcal{H}(\mathcal{O})$: if $\mathcal{A}$ executes this query, it can input string to get its hash value

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

Theorem 1. In the ROR model, assume that $\mathcal{A}$ can make Execute, Send, Hash, Corrupt, and Test queries. Then, the advantage of $\mathcal{A}$ to break the proposed protocol $\mathcal{P}$ in polynomial time $\xi$ is $Adv_{\mathcal{P}}(\xi) \leq q_{send}^2/2^{2q} + 3q_{hash}^2/2^{l-1} + 2 \max\{C' \cdot q_{send}, q_{send}/2\}$, 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 $GM_5$. $Succ_{GM_0}^\mathcal{A}(\xi)$ is defined by the probability that $\mathcal{A}$ succeeds in $GM_0$, which is the probability that $\mathcal{C} = 1$. The detailed simulations of queries in real attacks are shown in Tables 2 and 3. The details are as follows.

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

$$Adv_{\mathcal{P}}(\xi) = 2 \Pr \left[ Succ_{\mathcal{P}}^\mathcal{A}(\xi) \right] - 1.$$

$GM_0$: The difference between $GM_1$ and $GM_0$ is that $GM_1$ adds the Execute query. In $GM_1$, $\mathcal{A}$ just gets messages $M_1 = \{HID_{i}, C_1, C_2, C_3, C_4, T_1\}$, $M_2 = \{HID_{i}, C_5, C_6, C_7, T_3\}$, $M_3 = \{C_8, C_9, T_3\}$, and $M_4 = \{C_8, C_{10}, C_{11}, T_4\}$. After $GM_1$ is over, $\mathcal{A}$ queries the session key through $Te_s$, but $N_1, N_2$, and $N_3$ are all confidential to $\mathcal{A}$. Therefore,
Table 2: Simulation of $Send$ query.

On a query $Send(\Pi^u_I, \text{start})$, assuming that $\Pi^u_I$ is a normal state, we perform the following operations. Select $N_A$, $T_A$, and compute $A_1 = A_2 \oplus P_i$, $C_1 = N_i \oplus h(A_1 || \text{HID}_i)$, $C_2 = ID_i \oplus h(\text{HID}_i || T_A)$, $C_3 = \text{SID}_i \oplus h(A_i || N_A || T_A)$, $C_4 = h(ID_i || \text{HID}_i || \text{SID}_i || N_A || T_A)$.

Then, the query is answered by $M_1 = \{\text{HID}_i, C_1, C_2, C_3, C_4, T_1\}$.

On a query $Send(\Pi^GWN, (\text{HID}_i, C_1, C_2, C_3, C_4, T_1))$ and assume that $\Pi^GWN$ is a normal state to perform the following operations. Compute $A_1, ID_i, N_1, \text{SID}_i, C_2$, and check $A_1$. If equal, select $N_A, T_A$, and compute $\text{SID}_j, C_5, C_6, C_7$. Then, the query is answered by $M_2 = \{\text{HID}_i, C_3, C_5, C_6, C_7, T_2\}$.

On a query $Send(\Pi^T_{GWN}, (\text{HID}_i, C_1, C_2, C_3, C_4, T_3))$, assuming that $\Pi^T_{GWN}$ is a normal state, do the following. Compute $N_2, N_3, C_7$, check $C_7$. If equal, select $N_A, T_A$, and compute $\text{SID}_j, C_8, C_9$. Then, the query is answered by $M_4 = \{C_8, C_9, T_4\}$.

On a query, assuming that $\Pi^u_I$ is a normal state, we perform the following operations. Compute $N_2, N_3, \text{SID}_j, C_8$, the instance $\Pi^u_I$ checks $C_1$; if not equal, it will be terminated. Otherwise, compute $SK = h(N_i || N_2 || N_3 || \text{HID}_i || \text{SID}_i)$. Finally, the user instance accepts and terminates.

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

On an $Execute$ query, we use the simulation of Send query to do the following operations: $(\text{HID}_i, C_1, C_2, C_3, C_4, T_1) \leftarrow Send(\Pi^u_I, \text{start})$, $(\text{HID}_i, C_1, C_2, C_3, C_4, T_2) \leftarrow Send(\Pi^GWN, (\text{HID}_i, C_1, C_2, C_3, C_4, T_1))$, $(\text{HID}_i, C_1, C_2, C_3, C_4, T_3) \leftarrow Send(\Pi^T_{GWN}, (\text{HID}_i, C_1, C_2, C_3, C_4, T_1))$, $(\text{HID}_i, C_1, C_2, C_3, C_4, T_4) \leftarrow Send(\Pi^T_{GWN}, (\text{HID}_i, C_1, C_2, C_3, C_4, T_3))$. This query is answered by $(\text{HID}_i, C_1, C_2, C_3, C_4, T_1)$, $(\text{HID}_i, C_1, C_2, C_3, T_2)$, $(\text{HID}_i, C_1, C_2, C_3, T_3)$, and $(\text{HID}_i, C_1, C_2, C_3, T_4)$.

For a record $(\text{string}, r)$ that appears in the $\text{Hash(}\text{string}\text{)}$ query, return $r = \text{Hash(}\text{string}\text{)}$. Otherwise, select an element $r$, add the record $(\text{string}, r)$ to the list, and return $r$.

On a query $Corrupt(\Pi^u_I)$, and if $\Pi^u_I$ 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 $\mathcal{E}$ to get the result of $SK$. If $\mathcal{E} = 1$, return $SK$; otherwise, return a string of the same length.

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

$$
\text{Pr} \left[ \text{Succ}_{GM_1}(\xi) \right] = \text{Pr} \left[ \text{Succ}_{GM_0}(\xi) \right].
$$

(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

$$
\left| \text{Pr} \left[ \text{Succ}_{GM_2}(\xi) \right] - \text{Pr} \left[ \text{Succ}_{GM_1}(\xi) \right] \right| \leq \frac{q_{\text{send}}}{2^i}.
$$

(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

$$
\left| \text{Pr} \left[ \text{Succ}_{GM_3}(\xi) \right] - \text{Pr} \left[ \text{Succ}_{GM_2}(\xi) \right] \right| \leq \frac{q^2_{\text{hash}}}{2^{2i+1}}.
$$

(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 $\Pi_{GWN}$ 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. $\mathcal{A}$ uses $\Pi_{GWN}$ to try to get the private key $x$ of $GWN$ or uses $\Pi^u_I$ or $\Pi^T_I$ to try to get a secret value in the registration phase

(2) Known session-specific temporary information attacks. $\mathcal{A}$ uses either $\Pi^u_I$ or $\Pi^T_I$ or $\Pi_{GWN}$ to try to obtain the temporary information of the corresponding party.

In both cases, $\mathcal{A}$ can only compute the session key through $Send$ and $Hash$ queries. For the first case, if $\mathcal{A}$ only knows the private key $x$ of $GWN$, or a secret value of $\Pi^u_I$ or $\Pi^T_I$ in the registration phase, it cannot get the temporary information $N_1, N_2$, and $N_3$ in $SK = h(N_i || N_2 || N_3 || \text{HID}_i || \text{SID}_i)$. For the second case, we assume that $\mathcal{A}$ 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

$$
\left| \text{Pr} \left[ \text{Succ}_{GM_4}(\xi) \right] - \text{Pr} \left[ \text{Succ}_{GM_3}(\xi) \right] \right| \leq \frac{q_{\text{send}}}{2^{2i+1} + q^2_{\text{hash}}}.\quad (5)
$$

$GM_5$ : In this game, $\mathcal{A}$ uses $Corrupt(\Pi^u_I)$ 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 $\mathcal{A}$ gets $\text{HID}_i$ according to $M_1$, and computes $\sigma_i = \text{Rep}(B_i, \tau_i)$, $P_i = h(\sigma_i || PW || ID_i)$, $A'_3 = h(\text{HID}_i,$
\( ||P_i|| \) until \( A_i' = A_2 \). However, \( B_i \), \( PW_i \), and \( ID_i \) are all confidential to \( \mathcal{A} \). The probability that \( \mathcal{A} \) 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 \( q_{\text{send}} \leq 10^6 \). Therefore, we get

\[
\left| \Pr \left[ \text{Succ}^{GM_k} (\xi) \right] - \Pr \left[ \text{Succ}^{GM_{l'}} (\xi) \right] \right| \leq \max \left\{ \frac{C' \cdot q_{\text{send}}^l \cdot q_{\text{send}}^l}{2^l} \right\},
\]

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

GMₖ: The purpose of this game is to verify whether it can resist impersonation attacks. The difference between \( GM_{k} \) and \( GM_{l} \) is that when \( GM_{k} \) initiates \( h(N_1 || N_2 || N_3 || H_{ID_i} || S_{ID_i}) \) query to guess the session key, the game is terminated. Therefore, we have

\[
\frac{1}{2^{l+1}} \left| \Pr \left[ \text{Succ}^{GM_{k}} (\xi) \right] - \Pr \left[ \text{Succ}^{GM_{l'}} (\xi) \right] \right| \leq q_{\text{hash}}^l.
\]

Since the probability of \( GM_{k} \) success and failure is equal, the probability of \( \mathcal{A} \) successfully guessing the session key is

\[
\Pr \left[ \text{Succ}^{GM_{k}} (\xi) \right] = \frac{1}{2}.
\]

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

\[
\frac{1}{2^{l+1}} \left| \Pr \left[ \text{Succ}^{GM_{k}} (\xi) \right] - \Pr \left[ \text{Succ}^{GM_{l'}} (\xi) \right] \right| \leq q_{\text{send}}^l + 3q_{\text{hash}}^l + \max \left\{ \frac{C' \cdot q_{\text{send}}^l \cdot q_{\text{send}}^l}{2^l} \right\}.
\]

Thus, we have \( Adv_{\mathcal{A}}^G (\xi) \leq q_{\text{send}}^l / 2^{l+1} + 3q_{\text{hash}}^l + \max \left\{ \frac{C' \cdot q_{\text{send}}^l \cdot q_{\text{send}}^l}{2^l} \right\} \).

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 \( M_i \) 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 \( g_j \) 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 \( g_j \) 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 \( \{H_{ID_i}, ID_i, g_j\} \), but to calculate sensitive information such as \( A_1 \) and \( A_2 \), 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 \( \mathcal{A} \) get any two of the three factors.

1. Password and smart card. Even if \( \mathcal{A} \) 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 \( \mathcal{A} \) 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 \( \mathcal{A} \) obtains the biometric and smart card, he/she needs to know the information about \( PW_i \) and \( ID_i \) to calculate \( P_i \), so \( \mathcal{A} \) 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, \( H_{ID_i} \) 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).
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, IDi, HIDi))) in
let C2 = xor (IDi, h (con (con (HIDi, A1), T1))) in
let C3 = xor (SIDj, h (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 (xC8, h (con (con (A1, Pi), N1), xT4)) in
let N3 = xor (xC8, h (con (N1, N2))) in
let SKu = h (con (con (con (SIDj, SMj), ysl), x)) in
let C7' = h (con (con (con (con (N1, N2), SIDj), SMj), yT2)) in
if C7' = yC7 then event SensorAcGWN ();
new N2:bitstring;
new T2:bitstring;
new zsj:bitstring;
let SMj = h (con (con (con (SMj), zsj), x)) in
let C5 = xor (N2, h (con (con (con (A1, zPi), N1), 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 (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 (N1, N2), zHIDi), SIDj)) in
let C9 = h (con (con (SKg, SMj), T3)) 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, SMj), SIDj), T4), T2)) in
out (ch, (zC8, C10, C11, T4));
0).

(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.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>A2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>A3</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>A4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>A5 × [34]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>A6 × [36]</td>
<td>× [36]</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>A7</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>A8 × [34]</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>A9</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>12 T_f + 177 T_h</td>
<td>12T_f + 10T_h</td>
<td>15T_f + 13T_h</td>
</tr>
<tr>
<td>Gateway</td>
<td>2T_f + 37T_h</td>
<td>15T_f</td>
<td>14T_f</td>
</tr>
<tr>
<td>Sensor</td>
<td>9T_f + 5T_h</td>
<td>4T_f</td>
<td>6T_f</td>
</tr>
<tr>
<td>Total</td>
<td>12T_f + 38T_h</td>
<td>14T_f</td>
<td>15T_f</td>
</tr>
</tbody>
</table>

operations. Chen et al.’s protocol [35] not only performed 18 hashing operations but also performed four symmetric encryption/decryption operations, consuming 4T_c. 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.

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 \( Ms_{g1} = \{ TID_f, X_g, Y_g, Z_g, T_1 \} \), \( Ms_{g2} = \{ X_gw, Y_gw, Z_gw, T_2 \} \), and \( Ms_{g3} = \{ V_j, W_j, T_3 \} \), where \( TID_f \) is an identity, \( \{ X_g, Y_g, Z_gw, Y_gw, V_j \} \) belong to random strings, \( \{ Z_gw, W_j \} \) 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_j\}, \{M_3, CDI_{Df}, Ack, N_j\} \}, where \{M_1, M_2\} are ciphertexts, \{N_g, CDI_{Df}, 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 \( M_1 = \{ PID_1, D_1, D_2, D_3, D_4, T_3 \} \), \( M_2 = \{ D_5, D_6, D_7, D_8, T_2 \} \), \( M_3 = \{ D_9, D_10, D_{11}, D_{12}, T_4 \} \), and \( M_4 = \{ PID_2, ID_{Df} \} \) are identities, \{D_1, D_2, D_3, D_4, D_6, D_9, D_{10}, D_{11}\} are random strings, \{D_5, D_7, D_8, 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 \( M_1 = \{ HID, C_1, C_2, C_3, C_4, T_1 \} \), \( M_2 = \{ HID, \)
the one of Wu et al.'s 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 protocol is less than the one of Wu et al.'s and Chen et al.'s 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


