electronics

MDPI

Article

A Secure and Anonymous Authentication Protocol Based on Three-Factor Wireless Medical Sensor Networks

JoonYoung Lee [1], Jihyeon Oh [1] and Youngho Park [1,2,*]

[1] School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Republic of Korea; harry250@knu.ac.kr (J.L.); j2hnoh@knu.ac.kr (J.O.)
[2] School of Electronics Engineering, Kyungpook National University, Daegu 41566, Republic of Korea
* Correspondence: parkyh@knu.ac.kr; Tel.: +82-53-950-7842

Abstract: Wireless medical sensor networks (WMSNs), a type of wireless sensor network (WSN), have enabled medical professionals to identify patients' health information in real time to identify and diagnose their conditions. However, since wireless communication is performed through an open channel, an attacker can steal or manipulate the transmitted and received information. Because these attacks are directly related to the patients' lives, it is necessary to prevent these attacks upfront by providing the security of WMSN communication. Although authentication protocols are continuously developed to establish the security of WMSN communication, they are still vulnerable to attacks. Recently, Yuanbing et al. proposed a secure authentication scheme for WMSN. They emphasized that their protocol is able to resist various attacks and can ensure mutual authentication. Unfortunately, this paper demonstrates that Yuanbing et al.'s protocol is vulnerable to smart card stolen attacks, ID/password guessing attacks, and sensor node capture attacks. In order to overcome the weaknesses and effectiveness of existing studies and to ensure secure communication and user anonymity of WMSN, we propose a secure and anonymous authentication protocol. The proposed protocol can prevent sensor capture, guessing, and man-in-the-middle attacks. To demonstrate the security of the proposed protocol, we perform various formal and informal analyses using AVISPA tools, ROR models, and BAN logic. Additionally, we compare the security aspects with related protocols to prove that the proposed protocol has excellent security. We also prove the effectiveness of our proposed protocol compared with related protocols in computation and communication costs. Our protocol has low or comparable computation and communication costs compared to related protocols. Thus, our protocol can provide services in the WMSN environment.

Keywords: WMSN; PUF; biometrics; authentication; AVISPA; ROR

Citation: Lee, J.; Oh, J.; Park, Y. A Secure and Anonymous Authentication Protocol Based on Three-Factor Wireless Medical Sensor Networks. *Electronics* **2023**, *12*, 1368. https://doi.org/10.3390/electronics12061368

Academic Editors: Juan M. Corchado, Byung-Gyu Kim, Carlos A. Iglesias, In Lee, Fuji Ren and Rashid Mehmood

Received: 18 January 2023
Revised: 10 March 2023
Accepted: 10 March 2023
Published: 13 March 2023

1. Introduction

With the development of wireless internet technology, the Internet of Things (IoT) has experienced rapid expansion, with a large number of sensors being deployed in IoT devices. Wireless Sensor Network (WSN) is an essential IoT technology that enables data collection, monitoring, and exchange in diverse environments, e.g., smart grid monitoring and smart healthcare using WSN [1–3]. Applying WSN to a smart healthcare environment and using it for medical monitoring is called Wireless Medical Sensor Networks (WMSNs).

Based on WMSN, medical personnel, such as doctors and nurses, can continuously monitor the health of patients. These healthcare systems collect various medical factors, such as pulse, blood pressure, ECG, and body temperature by medical sensors attached to the patient [4]. By continuously monitoring this information, medical personnel can quickly diagnose a patient. WMSNs, similar to typical WSNs, comprise of users, gateways, and medical sensor nodes. The user (medical professional) and the medical sensor node register their respective information in the gateway node, and the users is able to obtain the patient's bio-information through the medical sensor node. However, since devices equipped with

medical sensors have limited capabilities (e.g, transmission range, calculation, and storage capabilities), protocols that use heavy computations may cause communication failure of the system [4]. In addition, because WMSN exchanges information through wireless open channels vulnerable to attack, if an attacker obtains the information shared on the wireless channel, the attacker can obtain the patient's medical information or deliver incorrect medical information to the user [5]. Thus, a communication failure of the system and manipulation of medical information by an attacker may result in a situation in which the patient's condition cannot be determined. Since this is directly related to the patients' lives, lightweight authentication among users, gateways, and sensor nodes based on a predetermined session key is essential for secure information exchange.

Therefore, in order to provide secure services using WMSN, researchers have proposed two-factor authentication protocols by adopting passwords and smart cards. However, the typical two-factor authentication protocol is not secure against guessing attacks and smartcard stolen attacks. Additionally, some researchers have argued that it is possible for attackers to guess ID/password pairs since users generate easy-to-remember ID/password pairs for their convenience [6–8].

In 2021, Yuanbing et al. [9] proposed an authentication scheme for smart healthcare by applying a WMSN. They argued that their protocol can resist against smartcard stolen, off-line guessing, and man-in-the-middle (MITM) attacks. They also said that their scheme can ensure mutual authentication and session key security. Unfortunately, we figure out that Yuanbing et al.'s scheme is not secure against off-line guessing, impersonation, sensor node capture, and MITM attacks. Furthermore, we discover that their scheme cannot provide session key security and mutual authentication. Our research contributions, motivations, and methodologies are discussed in Section 1.1.

1.1. Research Contributions, Motivations, and Methodology

Yuanbing et al. [9] analyzed Farash et al.'s protocol for secure authentication in smart healthcare systems and suggested an enhanced protocol. However, we identify security vulnerabilities in Yuanbing et al.'s protocol. Their protocol is vulnerable to offline ID/password pair guessing, impersonation, sensor capture, sensor impersonation, and MITM attacks over an WMSN, and does not guarantee essential security features. In addition, their protocol adopts elliptic curve cryptography (ECC), so the computation cost is high.

To overcome these problems, we propose a secure authentication protocol for smart healthcare based on WMSN. We adopt three-factor using the user's biometrics, as well as smart card adoption to defend against ID/password pair guessing attacks. We also introduce Physical Unclonable Function (PUF) [10] technology to defend against sensor node physical capture attacks.

To prove the security of our protocol, we conduct formal and informal (non-mathematical) security analysis. We use the widely adopted Real-Or-Random (ROR) model [11] and Burrows Abadi Needham (BAN) logic [12] to perform formal analysis. Furthermore, we use the AVISPA [13] for proving that the proposed protocol can be secure against replay and MITM attacks. By comparing the proposed protocol with other authentication protocols, the efficiency is analyzed in terms of security, communication costs, and computational costs.

1.2. Organization of the Paper

The rest of this paper is structured as follows: In Section 2, we review diverse authentication protocols using WMSN. We briefly describe the protocol's system model, adversary model, PUF, and fuzzy extraction in Section 3. We review the protocol of Yuanbing et al. in Section 4. We present a cryptographic analysis of the protocol in Section 5. Section 6 explains our proposed protocol and there is a security analysis in Section 7. We analyze the computation and communication costs of the protocol in Section 8. Finally, the work is summarized in Section 9.

2. Related Works

Research on authentication protocol for the WSN environment has been continued since Lamport [14] proposed a password-based authentication protocol for various network environments in 1981. We briefly review authentication protocols related to WMSN and wearable devices in the WSN environment. In 2012, Kumar et al. [15] suggested a new authentication protocol in the WMSN environment. They explained that the proposed protocol using only symmetric encryption and hash functions is effective to protect communication security. However, He et al. [6] found that the scheme of Kumar et al. was not secure against offline password guessing attacks, which could lead to guessing the user's identity. To overcome this security vulnerability, He et al. proposed an improved authentication protocol for resource-limited sensors. Unfortunately, Li et al. [16] pointed out that He et al.'s protocol is wrong in the protocol in the authentication and session key agreement phase. Li et al. proposed an authentication scheme using biometrics. Das et al. [17] found that Li et al.'s scheme is vulnerable to privileged insider attacks, smart device physical capture attacks, and fails to maintain the anonymity of users and smart devices. In 2018, Amin et al. [18] suggested a lightweight two-factor authentication protocol for protecting data transmitted in the WMSN environment. However, Jiang et al. [19] discovered that their protocol is vulnerable to mobile device loss attacks due to failed delivery, sensor key exposure, and desynchronization attacks. Jan et al. [20] also found that security flaws of Amin et al.'s protocol. Then, Jan et al. introduced a two-factor-based authentication scheme for WMSN. In 2020, Fotouhi et al. [21] proposed a lightweight two-factor-based authentication protocol for healthcare monitoring systems. However, Nashwan [22] figured out that Fotouhi et al.'s scheme cannot support full mutual authentication. In 2021, Masud et al. [23] designed a lightweight and privacy-protected authentication protocol for IoT-based healthcare using the sensors of an IoT device carried out by a patient. However, Masud et al.'s protocol is also insecure against offline password guessing, user impersonation, and privileged insider attacks, and do not provide user anonymity [24].

Moreover, some researchers have pointed out that these traditional two-factor authentication protocols are vulnerable to ID/password simultaneous guess attacks [6–8]. Accordingly, a three-factor authentication protocol was proposed using the user's biometric information.

In 2018, Ali et al. [25] discovered the problems of the protocol of Amin et al. [18] and introduced an authentication protocol based on three factors to solve the problem. However, their proposed protocol also cannot protect against desynchronization attacks or achieve full forward secrecy [26]. Shuai et al.'s protocol [26] uses a pseudonymous identification method to ensure forward secrecy, provide user anonymity, and resist desynchronization attacks. However, Nashwan [22] discovered that Shuai et al.'s protocol is not able to support the sensor node's anonymous service and cannot protect against sensor node impersonation attacks. Mo et al. [27] also found that Shuai et al.'s protocol has a flaw at the password change phase. Then, Mo et al. proposed an enhanced protocol for WMSN. Li et al. [28] suggested an authentication scheme that guarantees perfect forward secrecy in the WMSN environment by using a three-factor method. However, their protocol also cannot guarantee the security of the sensor node and is vulnerable to sensor node spoofing attacks [29].

Since the three-factor-based authentication protocol for the WMSN-based medical system is also vulnerable to sensor node vulnerabilities and sensor node spoofing attacks, an improved system is designed according to a new technology called PUF. In 2018, Gope and Sikdar [30] designed a two-factor authentication scheme using PUF technology in the IoT environment. However, their protocol cannot resist ephemeral secret leakage (ESL) attacks and desynchronization attacks [31].

Authentication protocols using technologies such as multi-factor and PUF have been continuously proposed for the WMSN environment, but security vulnerabilities still exist. When security vulnerabilities occur in the WMSN environment, the patient's medical information can be manipulated or leaked by a malicious attacker. The recently proposed

protocol of Yuanbing et al. [9] is not safe against ID/password pair guessing attacks and spoofing attacks, and the security of sensor nodes cannot be guaranteed either. In this paper, we not only analyze the security vulnerabilities of Yuanbing et al.'s protocol, but also the proposed protocol, which can solve the security vulnerabilities that can occur even when using three-factor and PUF technology.

3. Preliminaries

We introduce the system model of the proposed protocol and the adversary model for protocol security analysis. In addition, the security technologies adopted by the proposed protocol, PUF, and fuzzy extraction will be briefly described. The symbols used in this paper are given in Abbreviations.

3.1. System Model of Our Protocol

The proposed system model is shown in Figure 1. Our proposed protocol consists of the following entities:

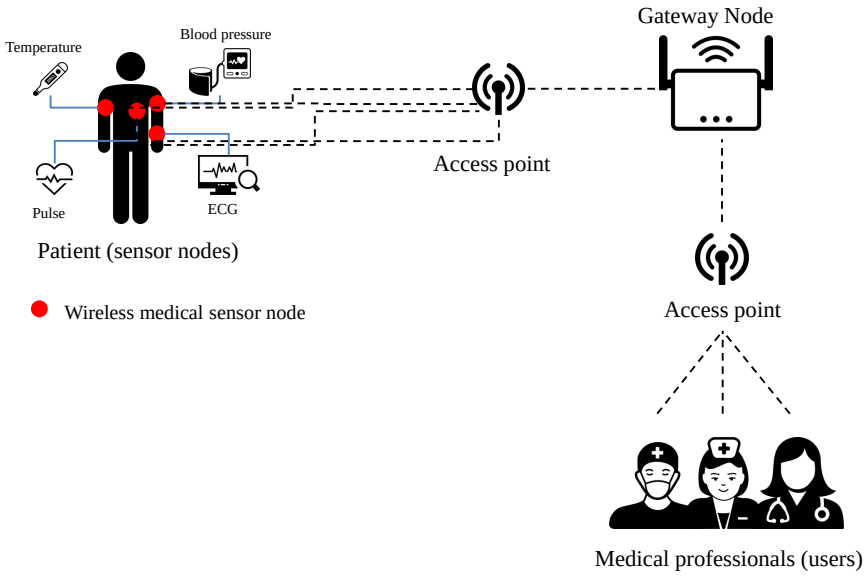

Figure 1. The proposed system model.

- **User (Medical Professional)**: The user obtains the patient's sensor node information by requesting communication to the gateway. To this end, users register their information with the gateway and agree on a session key with the sensor node. In the future, only registered users can request communication to the gateway and use secure services through the session key.
- **Sensor node (patient)**: The sensor node that the patient is equipped with collects various health information of the patient (e.g., body temperature, blood pressure, pulse, and ECG). The patient's sensor nodes transmit the collected information to the user through the session key. Through this, the user can identify and diagnose the patient's condition. Sensor nodes are resource-limited devices.
- **Gateway node**: A gateway is a trusted entity that performs registration and authentication processes, and regulates the authentication of users and sensor nodes. All users and sensor nodes must be registered with the gateway to acquire session keys and to communicate.
- **Access point**: The access point is a wireless connection between the patient's sensor node and the gateway and between the user and the gateway. The communication between each access point and each entity is considered securely authenticated.

Sensor nodes and users must first register through the gateway. When sensor nodes and users request registration, the gateway stores their related registration information and controls communication between the users and sensor nodes. The users and relevant sensor nodes agree on a session key through the gateway, and secure communication can be achieved later through the agreed session key. The proposed protocol is composed of registration, login and authentication, and password and biometric update phases. At the registration phase, sensor nodes and users register their information through the gateway. Users, gateways, and sensor nodes perform mutual authentication at the login and authentication phase, and they agree on a session key for communication. At this time, our protocol uses the user's biometric information to defend against malicious attacker's ID/password pair guessing attack. In addition, the sensor node has a built-in PUF technology that guarantees security against physical capture attacks. Later, the user is able to safely collect the sensor node information by using the session key, and can manage the patient's health based on this. At the password and biometric update phase, users are able to update passwords and biometrics.

3.2. Physical Unclonable Function

To securely store secret values and identity information in sensor nodes, we adopt PUF technology. PUF is able to be portrayed as "the representation of the instance-specific functionality, non-replicable, and unique of a physical entity" [10]. The uncertainty and randomity of the integrated circuit manufacturing are less likely to generate a duplicate value, so the PUF attracts more attention. A PUF receives challenge C and then obtains the response R through the physical properties of integrated chip and C. The allowed C and the generated R can be represented by the string of the bit. This operation can be expressed as Equation (1).

$$R = PUF(C) \tag{1}$$

and this is like the nature of a one-way function. Ideally, there is a one-to-one correspondence between the PUF and the C/R pair. Furthermore, if the same PUF is challenged multiple times, responses will be the same, but the responses obtained for different PUFs will be different when given the same challenge. Additional characteristics of PUF are:

- It is not possible to clone a PUF to generate the same sensors or devices [32].
- If an attacker tries to change the sensor or device that the PUF is mounted on, the sensor or device will change the behavior of the PUF and destroy the PUF [33].
- In practical circuit manufacturing, the differences in input and output function mapping are fixed and unpredictable [34].

However, in a realistic situation, due to environmental and circuit noise, it is difficult for a PUF to always return the same R because of the small margin of error of C. To solve this, PUF has been applied together with fuzzy extraction technology [35].

3.3. Fuzzy Extraction

Fuzzy extraction [35] can be used to solve noise problems that may occur in biometric inputs. Furthermore, fuzzy extraction can also help with noisy PUF. The fuzzy extractor can get the same value by removing the noise through the *Gen* function and the *Rep* algorithm.

The *Gen* algorithm generates key information that can respond to input values such as C of PUF or BIO of biometric information. In other words, if the data D_i is used as an input to the *Gen* algorithm, the secret key data R_i is output, and it is a uniform random string. At the same time that R_i is output, the fuzzy extractor outputs the P_i to help recover the key values and remove noise. This algorithm could be presented as Equation (2).

$$Gen(D_i) =< R_i, P_i > \tag{2}$$

The *Rep* algorithm can restore the secret key R_i from P_i and the entered C and BIO values. First, input data D_i such as C or BIO and P_i helper strings are input to the *Rep*

algorithm. This can cause noise in data D_i. In this case, P_i helps remove noise and restore data to output the correct R_i. The metric spatial distance between D_i and $D_i^{'}$ must be within the specified tolerance for the fuzzy extractor to recover a matching R_i. This algorithm could be presented as Equation (3).

$$Rep(D_i, P_i) = R_i \tag{3}$$

3.4. Adversary Model

We use the "Dolev–Yao (DY) adversary model [36]" to conduct the security analysis of the protocol proposed by Yuanbing et al. To this end, we first discuss the attack potential of attackers according to the DY model.

- According to the DY model, attackers have full control and learning of the messages exchanged on open wireless channels that are vulnerable to attack. They can then modify, remove, or insert legitimate messages.
- Attackers can obtain or steal users' legitimate smart cards. After that, they can obtain the secret information stored on the smartcard by performing power analysis attacks [37,38].
- After obtaining the secret information of the smart card or sensor node, the attacker can try potential attacks such as offline identity guessing attacks, impersonation, and so on [39,40].
- Attackers can guess a user's identity and password pairs in polynomial time.

In addition to the DY model, we also adopt the "Canetti–Krawczyk (CK) adversary model" [41]. The CK model is a more practical attacker model compared to the DY model. According to the CK model, for the consensus session key to be secure, the key exchange protocol must minimize the impact of long-term or short-term secret leaks.

4. Review of Yuanbing et al.'s Protocol

This section briefly presents the Yuanbing et al.'s authentication protocol for a smart healthcare system based on WMSN.

4.1. Pre-Deployment Phase

In this phase, a system administrator adopts X_{GWN} that only the GWN knows in offline mode, and predefined SID_j, which is the ID of each sensor node SN_j. The system administrator also sets a pre-shared key X_{GWN-SN_j} for each SN_j with the associated GWN. The protocol of Yuanbing et al. uses the shared key X_{GWN-SN_j} at the sensor node registration phase. It is important to note the password X_{GWN-SN_j} is deleted from SN_j's memory once SN_j is successfully registered.

4.2. Sensor Node Registration Phase

SN_j registers its information in GWN at this phase. The detailed steps are as follows.

Step 1: SN_j chooses a random nonce r_j, and calculates $MP_j = h(X_{GWN-SN_j}||r_j||SID_j||T_1)$ and $MN_j = r_j \oplus X_{GWN-SN_j}$. After that, SN_j transmits $\{SID_j, MN_j, MP_j, T_1\}$ to the gateway.

Step 2: GWN checks the timestamp. If the condition holds, GWN calculates $r_j^{'} = MN_j \oplus X_{GWN-SN_j}$. Then, GWN computes $MP_j^{'} = h(X_{GWN-SN_j}||\ r_j^{'}||SID_j||T_1)$ and checks if $MP_j = MP_j^{'}$. If it is the same, GWN computes $x_j = h(SID_j||X_{GWN})$, $e_j = x_j \oplus X_{GWN-SN_j}$, $d_j = h(X_{GWN}||1) \oplus h(X_{GWN-SN_j}||T_2)$, and $f_j = h(x_j||d_j||X_{GWN-SN_j}||T_2)$. Then, GWN sends $\{d_j, f_j, e_j, T_2\}$ to SN_j.

Step 3: SN_j checks the timestamp. If the condition is correct, SN_j calculates $x_j = e_j \oplus X_{GWN-SN_j}$ and checks if $f_j = h(x_j||d_j||X_{GWN-SN_j}||T_2)$. If it is correct, SN_j computes $h(X_{GWN}||1) = d_j \oplus h(X_{GWN-SN_j}||T_2)$. SN_j stores $\{x_j, h(X_{GWN}||1)\}$ in its memory. Then, SN_j deletes the X_{GWN-SN_j} and sends a respond message to GWN.

Step 4: Upon receiving the response message, GWN removes $\{SID_j, X_{GWN-SN_j}\}$.

4.3. User Registration Phase

Users (such as nurses and doctors) must first register with the GWN when they want to obtain the patient's medical data at this phase. The detailed steps are as follows.

Step 1: The user U_i selects their ID_i and PW_i, r_i. Then, U_i calculates $MID_i = h(r_i||ID_i)$, $MP_i = h(r_i||PW_i)$, and $RSP_i = h(ID_i||MP_i)$. After that, U_i sends $\{RSP_i, MID_i\}$ to GWN.

Step 2: GWN calculates $e_i = h(RSP_i||MID_i)$, $d_i = h(MID_i||X_{GWN})$, $g_i = h(X_{GWN}) \oplus h(RSP_i||d_i)$, and $f_i = d_i \oplus h(RSP_i||e_i)$. Then, GWN stores $\{e_i, f_i, g_i\}$ into SC and issues it to U_i.

Step 3: U_i computes $r_i^* = h(ID_i||PW_i) \oplus r_i$ and stores r_i^* into SC.

4.4. Login and Authentication Phase

At this phase, U_i agrees on a session key by conducting mutual authentication with SN_j before accessing the patient's medical information. The detailed steps are shown in Figure 2 and described below.

User (U_i)	Sensor node (SN_j)	Gateway (GWN)
Inserts the smartcard. Inputs ID_i, PW_i Computes $r_i^{'} = r_i^* \oplus h(ID_i\|\|PW_i)$, $MID_i^{'} = h(r_i^{'}\|\|ID_i)$, $MP_i^{'} = h(r_i^{'}\|\|PW_i)$, and $RSP_i^{'} = h(ID_i\|\|MP_i^{'})$. Checks if $e_i = h(RSP_i^{'}\|\|MID_i^{'})$? Generates random nonce a and c. Computes $d_i = f_i \oplus h(RSP_i^{'}\|\|e_i)$, $h(X_{GWN}) = g_i \oplus h(RSP_i^{'}\|\|d_i)$, $R_1 = aP$, $MID_1 = h(c\|\|ID_i)$, $x_i = h(MID_1\|\|h(X_{GWN}))$, $M_1 = MID_1 \oplus h(h(X_{GWN})\|\|T_1)$, $M_2 = h(M_1\|\|x_i\|\|R_1\|\|T_1)$. $\{M_1, M_2, R_1, T_1\}$ $\xrightarrow{\text{(via open channel)}}$	Checks $\|T_1 - T_c\| < \Delta T$? Computes $ESID_j = SID_j \oplus h(h(X_{GWN})\|\|1)\|\|T_2)$, $R_2 = bP$, $R_3 = bR_1$ and $M_3 = h(SID_j\|\|x_j\|\|R_2\|\|T_1\|\|T_2)$. $\{M_1, M_2, M_3, T_1, T_2, ESID_j, R_1, R_2\}$ $\xrightarrow{\text{(via open channel)}}$	
		Checks $\|T_2 - T_c\| < \Delta T$? Computes $SID_j^{'} = ESID_j \oplus h(h(X_{GWN})\|\|1)\|\|T_2)$, $x_j^{'} = h(SID_j^{'}\|\|X_{GWN})$. Checks $M_3 = h(SID_j^{'}\|\|x_j^{'}\|\|R_2\|\|T_1\|\|T_2)$? Computes $MID_1^{'} = M_1 \oplus h(h(X_{GWN})\|\|T_1)$, $x_i^{'} = h(MID_1^{'}\|\|h(X_{GWN}))$. Checks $M_2 = h(M_1\|\|x_i^{'}\|\|R_1\|\|T_1)$?
	Checks $\|T_3 - T_c\| < \Delta T$? Checks $M_5 = h(x_j\|\|R_1\|\|T_3)$? Computes $MID_1^{'} = M_6 \oplus h(x_j\|\|T_3)$, $SK = h(MID_1^{'}\|\|SID_j\|\|R_3\|\|T_3\|\|T_4)$, $M_7 = h(SK\|\|M_4\|\|T_3\|\|T_4)$. $\{M_4, M_7, R_2, T_3, T_4\}$ $\xleftarrow{\text{(via open channel)}}$	Computes $M_4 = h(x_i^{'}\|\|R_2\|\|T_3)$, $M_5 = h(x_j^{'}\|\|R_1\|\|T_3)$, $M_6 = MID_1^{'} \oplus h(x_j^{'}\|\|T_3)$. $\{M_4, M_5, M_6, R_1, T_3\}$ $\xleftarrow{\text{(via open channel)}}$
Checks $\|T_4 - T_c\| < \Delta T$? Computes $M_4 = h(x_i\|\|R_2\|\|T_3)$, $R_4 = aR_2$, $SK = h(MID_1\|\|SID_j\|\|R_4\|\|T_3\|\|T_4)$ Checks $M_7 = h(SK\|\|M_4\|\|T_3\|\|T_4)$?		

Figure 2. Login and authentication phase.

Step 1: The user U_i inserts SC, and inputs ID_i and PW_i. SC computes $r_i^{'} = r_i^* \oplus h(ID_i||PW_i)$, $MID_i^{'} = h(r_i^{'}||ID_i)$, $MP_i^{'} = h(r_i^{'}||PW_i)$, and $RSP_i^{'} = h(ID_i||MP_i^{'})$. Then, U_i checks if $e_i =?h(RSP_i^{'}||MID_i^{'})$. If it corrects, SC generates random nonce a and c, and computes $d_i = f_i \oplus h(RSP_i^{'}||e_i)$, $h(X_{GWN}) = g_i \oplus h(RSP_i^{'}||d_i)$. SC also calculates $R_1 = aP$, $MID_1 = h(c||ID_i)$, $x_i = h(MID_1||h(X_{GWN}))$, $M_1 = MID_1 \oplus h(h(X_{GWN})||T_1)$, and $M_2 = h(M_1||x_i||R_1||T_1)$. Then, U_i sends $\{M_1, M_2, R_1, T_1\}$ to SN_j through an open channel.

Step 2: SN_j checks $|T_1 - T_c| < \Delta T$?. If this condition holds, SN_j generates timestamp T_2 and random nonce b. SN_j computes $ESID_j = SID_j \oplus h(h(X_{GWN})||1)||T_2)$, $R_2 = bP$, $R_3 = bR_1$ and $M_3 = h(SID_j||x_j||R_2||T_1||T_2)$. After that, SN_j transmits $\{M_1, M_2, M_3, T_1, T_2, ESID_j, R_1, R_2\}$ to GWN through an open channel.

Step 3: After receiving the message, GWN checks $|T_2 - T_c| < \Delta T$?. If this condition holds, GWN computes $SID_j^{'} = ESID_j \oplus h(h(X_{GWN}||1)||T_2)$ and $x_j^{'} = h(SID_j^{'}||X_{GWN})$.

Then, GWN checks $M_3 = ?h(SID_j^{'}||x_j^{'}||R_2||T_1||T_2)$. If it holds, GWN computes $MID_1^{'} = M_1 \oplus h(h(X_{GWN})||T_1)$ and $x_i^{'} = h(MID_1^{'}||h(X_{GWN}))$. GWN checks $M_2 = ?h(M_1||x_i^{'}||R_1||T_1)$. If it is correct, GWN computes $M_4 = h(x_i^{'}||R_2||T_3)$, $M_5 = h(x_j^{'}||R_1||T_3)$, and $M_6 = MID_1^{'} \oplus h(x_j^{'}||T_3)$. Then, GWN sends $\{M_4, M_5, M_6, R_1, T_3\}$ to SN_j.

Step 4: Upon receiving the message, SN_j checks $|T_3 - T_c| < \Delta T?$. If this condition holds, SN_j checks $M_5 = ?h(x_j||R_1||T_3)$. If these values are the same, SN_j computes $MID_1^{'} = M_6 \oplus h(x_j||T_3)$, $SK = h(MID_1^{'}||SID_j||R_3||\ T_3||T_4)$, and $M_7 = h(SK||M_4||T_3||T_4)$. Finally, SN_j sends $\{M_4, M_7, R_2, T_3, T_4\}$ to U_i.

Step 5: Upon receiving the message, U_i checks $|T_4 - T_c| < \Delta T?$. If this condition holds, U_i computes $M_4 = h(x_i||R_2||T_3)$, $R_4 = aR_2$, and $SK = h(MID_1||SID_j||R_4\ ||T_3||T_4)$. After that, U_i checks $M_7 = ?h(SK||M_4||T_3||T_4)$. If it corrects, U_i and SN_j shares the same SK at the end.

5. Security Analysis of Yuanbing et al.'s Protocol

We conduct the security analysis proposed by Yuanbing et al. [9] in this section. Yuanbing et al. demonstrated the ability of their protocol to resist diverse types of attacks. They also claimed that it is capable of providing anonymity. However, we contend that their protocol is vulnerable to smartcard theft, impersonation, sensor node capture attacks, and so on. Further, we prove that it fails to provide user anonymity in some cases.

5.1. Off-Line Guessing Attacks

A malicious attacker ATT is able to acquire secret credentials stored in the smartcard, as discussed at Section 3.4. Furthermore, ATT can also obtain transmitted messages over insecure channels. Finally, ATT is able to guess the ID and password pair. The detailed steps are as follows:

Step 1: ATT can obtain the stored values $\{e_i, f_i, g_i, r_i^*\}$ from the smartcard through power analysis attacks, and ATT selects the guessing ID/password pair ID_{att}/PW_{att}.

Step 2: ATT computes $r_{att}^{'} = r_i^* \oplus h(ID_{att}||PW_{att})$, $MID_{att}^{'} = h(r_{att}^{'}||ID_{att})$, $MP_{att}^{'} = h(r_{att}^{'}||PW_{att})$, and $RSP_{att}^{'} = h(ID_{att}||MP_{att}^{'})$. Then, ATT checks if $e_i = h(RSP_{att}^{'}||MID_{att}^{'})?$.

Step 3: If these values match, ATT is considered to have successfully guessed the legitimate user's ID and password pair. If they do not match, ATT repeats Step 1 and 2 again to guess ID_i/PW_i.

5.2. Impersonation Attacks

If ATT can successfully guess the legitimate identity and password pair of a user through Section 5.1, ATT computes a valid RSP_i to perform user impersonation attacks. The detailed steps of user impersonation attacks are as follows.

Step 1: If ATT succeeds in guessing the ID/password pair of U_i, then ATT can compute a valid RSP_i. Then, ATT can calculate $d_i = f_i \oplus h(RSP_i||e_i)$ using the values e_i and f_i stored in the U_i's smartcard. ATT can also compute $h(X_{GWN}) = g_i \oplus h(RSP_i||d_i)$ using g_i stored values in the smartcard.

Step 2: After that, ATT chooses random nonces a_{att} and c_{att} and timestamp T_{att}. Subsequently, ATT can compute $R_{1att} = a_{att}P$, $MID_{1att} = h(c_{att}||ID_i)$, $x_i = h(MID_1||h(X_{GWN}))$, $M_{1att} = MID_{1att} \oplus h(h(X_{GWN})||T_{att})$, and $M_{2att} = h(M_{1att}||x_i||R_1||T_{att})$. Thus, ATT can compute the login request message $\{M_{1att}, M_{2att}, R_{1att}, T_{att}\}$. Therefore, ATT can conduct successful impersonation attacks.

5.3. Sensor Node Impersonation Attacks

ATT can obtain the $\{SID_j, x_j, h(X_{GWN}||1)\}$ stored in SN_j through sensor node capture attacks. ATT can then use the obtained values to compute a valid message and impersonate it as a valid SN_j. After the sensor node capture attacks, ATT can perform sensor node impersonation attacks as follows:

Step 1: ATT can acquire the values stored $\{SID_j, x_j, h(X_{GWN}||1)\}$ in SN_j through the sensor node capture attack. Then, ATT chooses random nonces b_{att} and T_{att}. Subsequently, ATT calculates $ESID_{att} = SID_j \oplus h(h(X_{GWN}||1)||T_{att})$.

Step 2: ATT can obtain the values $\{M_1, M_2, R_1, T_1\}$ through the message sent to insecure channels. ATT can calculate $R_{2att} = b_{att}P$, $R_{3att} = b_{att}R_1$, $M_{3att} = h(SID_j||x_j||R_{2att}||T_1||T_{att})$. ATT chooses a random nonce b_{att} and timestamp T_{att}. Subsequently, ATT can compute $R_{2att} = b_{att}P$, $R_{3att} = b_{att}R_1$, $M_{3att} = h(SID_j||x_j||R_{2att}||T_1||T_{att})$. Then, ATT sends the message $\{M_1, M_2, M_{3att}, T_1, T_{2att}, ESID_j, R_1, R_{2att}\}$ to GWN. Thus, we can say that ATT can impersonate the sensor node.

5.4. MITM Attacks

After sensor node impersonation attacks, ATT can conduct MITM attacks using R_{3att}. The detailed steps are as follows:

Step 1: When ATT receives the message $\{M_4, M_5, M_6, R_1, T_3\}$, ATT computes $MID_1^{'} = M_6 \oplus h(x_j||T_3)$ using x_j obtained through sensor node capture attacks.

Step 2: Then, ATT can compute the fake session key $SK_{att} = h(MID_1^{'}||SID_j||R_{3att}||T_3||T_4)$. ATT also computes $M_{7att} = h(SK_{att}||M_4||T_3||T_4)$. Finally, ATT sends the message $\{M_4, M_{7att}, R_2, T_3, T_4\}$ to U_i.

5.5. Fail to Ensure Anonymity and Mutual Authentication

Yuanbing et al. argued that the proposed protocol guarantees anonymity and provides mutual authentication. However, according to Section 5.1, ATT is able to obtain the legitimate user's real ID ID_i. Furthermore, according to Sections 5.2 and 5.3, ATT can impersonate the user or sensor node. In particular, according to Section 5.4, ATT can interfere with mutual authentication by performing a man-in-the-middle attack. Therefore, Yuanbing et al.'s protocol fails to ensure user anonymity and mutual authentication.

6. Proposed Protocol

We propose a secure authentication protocol to overcome the problems of Yuanbing et al.'s proposed protocol. It utilizes the user's biometrics to prevent off-line guessing attacks of ID/password pairs. We also introduce PUF technology to prevent capture attacks of sensor nodes. Therefore, the proposed protocol is found to be secure against various attacks. Additionally, the proposed protocol is a lightweight protocol to take into account the resource limitations of sensor nodes.

6.1. User Registration Phase

A user U_i who wants to communicate with a specific SN must register with GWN. The detailed steps are shown in Figure 3 and explained below.

Step URP1: User U_i chooses identity ID_i and password PW_i, and imprints their biometrics BIO_i. U_i generates a random nonce RN_u. Then, U_i computes $Gen(BIO_i) =< UR_i, P_i >$, $HID_i = h(UR_i||ID_i)$, $HPW_i = h(RN_u||UR_i||ID_i||PW_i)$, and $RSP_i = h(ID_i||HPW_i)$. Subsequently, U_i sends $\{HID_i, RSP_i, HPW_i\}$ to GWN through a secure channel.

Step URP2: Upon receiving the message $\{HID_i, RSP_i, HPW_i\}$, GWN checks if HID_i is in its database. If not, GWN creates a random nonce RN_{gw}. GWN calculates $\alpha_i = h(HID_i||X_{GWN}||RN_{gw})$, $\beta_i = \alpha_i \oplus HPW_i$, and $\gamma_i = h(HID_i||RSP_i||\alpha_i)$. Subsequently, GWN saves $\{HID_i, R_{gw}\}$ in its database and also stores $\{\beta_i, \gamma_i\}$ in SC. Then, GWN issues SC to U_i via a closed channel.

Step URP3: After receiving SC from GWN, U_i computes $L_i = h(UR_i||PW_i) \oplus RN_u$ and stores L_i and UP_i in SC. Finally, SC stores $\{\beta_i, \gamma_i, L_i, UP_i\}$.

6.2. Sensor Node Registration Phase

SN_j is assigned an ID SID_j by the system administrator before being deployed. To register in GWN, SN_j selects the PUF's challenge and computes R_j and the registration request message. After GWN receives the registration request message, GWW calculates

and sends the values required for authentication to SN_j. The proposed protocol considers the data load for sensor nodes with limited capabilities. The sensor node stores only $\{SID_j, PSID_j, K_j, SNP_j\}$. Assuming that the ID and hash values are 160 bits, the value stored by the sensor node is only 640 bits. The sensor node registration steps are as follows and shown in Figure 3.

User Registration Phase	
User (U_i)	Gateway (GWN)
Chooses ID_i, PW_i and imprints BIO_i Generates a random nonce RN_u Computes $Gen(BIO_i) =< UR_i, UP_i >$, $HID_i = h(UR_i \|\| ID_i)$, $HPW_i = h(RN_u \|\| UR_i \|\| ID_i \|\| PW_i)$, $RSP_i = h(ID_i \|\| HPW_i)$. $\{HID_i, RSP_i, HPW_i\} \longrightarrow$ (via secure channel)	 Checks if HID_i is in its database If not, generates a random nonce RN_{gw} Computes $\alpha_i = h(HID_i \|\| X_{GWN} \|\| RN_{gw})$, $\beta_i = \alpha_i \oplus HPW_i$, $\gamma_i = h(HID_i \|\| RSP_i \|\| \alpha_i)$. Stores $\{HID_i, R_{gw}\}$ in its database. $SC = \{\beta_i, \gamma_i\} \longleftarrow$ (via secure channel)
Computes $L_i = h(UR_i \|\| PW_i) \oplus RN_u$. Stores $\{\beta_i, \gamma_i, L_i, UP_i\}$ into SC.	
Sensor Node Registration Phase	
Sensor Node (SN_j)	Gateway (GWN)
Picks challenge C_j. Generates random nonce RN_{sn}. Computes the response value $R_j = PUF(C_j)$, $Gen(R_j) =< SNR_j, SNP_j >$, $Req_j = SID_j \oplus h(RN_{sn})$, $HS_j = h(SID_j \|\| SNR_j)$. $\{Req_j, RN_{sn}, C_j, HS_j\} \longrightarrow$ (via secure channel)	 Computes $SID_j = Req_j \oplus h(RN_{sn})$. Generate a random secret key y_{GWN}. Computes $PSID_j = h(SID_j \|\| RN_{sn})$, $K_j = h(PSID_j \|\| X_{GWN} \|\| y_{GWN})$. Stores $\{PSID_j, HS_j, y_{GWN}, C_j\}$ $\langle PSID_j, K_j \rangle \longleftarrow$ (via secure channel)
Stores $\{SID_j, PSID_j, K_j, SNP_j\}$	

Figure 3. Registration phase.

Step SRP1: SN_j chooses challenge value C_j, and generates random nonces RN_{sn}. Then, SN_j computes the response value $R_j = PUF(C_j)$. Furthermore, SN_j computes $Gen(R_j) =< SNR_j, SNP_j >$, $Req_j = SID_j \oplus h(RN_{SN})$, and $HS_j = h(SID_j || SNR_j)$. SN_j sends $\{Req_j, RN_{sn}, C_j, HS_j\}$ to GWN via a closed channel.

Step SRP2: When GWN receives the message $\{Req_j, RN_{sn}, C_j, HS_j\}$, GWN computes $SID_j = Req_j \oplus h(RN_{sn})$. GWN creates a random secret key y_{GWN}, and calculates $PSID_j = h(SID_j || RN_{sn})$ and $K_j = h(PSID_j || X_{GWN} || y_{GWN})$. GWN stores $\{PSID_j, y_{GWN}, HS_j, C_j\}$ in its database and sends $\{PSID_j, K_j\}$ to SN_j.

Step SPR3: Upon receiving the message $\{PSID_j, y_{GWN}, HS_j, C_j\}$, SN_j stores $\{SID_j, PSID_j, K_j\}$ in its secure memory.

6.3. Login and Authentication Phase

In the login and authentication phase, a session key is generated for U_i to communicate with a specific SN_j. All entities perform mutual authentication through message verification, and when mutual authentication is successful, a session key SK for future communication is agreed upon. In the proposed protocol, the user manually selects a new pseudo ID HID_{inew}. When authentication is complete, the user updates the β_{inew} and γ_{inew} values associated with HID_{inew}. It is assumed that the IEEE 802.15.4 protocol is used for communication between the sensor node and the GWN, and the IEEE 802.11 protocol is used for communication between the GWN and the user [42]. The detailed steps are shown in Figure 4 and are described in detail below.

User (U_i)	Gateway (GWN)	Sensor Node (SN_j)
Inputs SC.		
Inputs ID_i, PW_i, BIO_i.		
SC computes		
$Rep(BIO_i, UP_i) = UR_i$,		
$RN_u = L_i \oplus h(UR_i \| PW_i)$,	Checks $\|T_1 - T_c\| < \Delta T$?	
$HID_i = h(UR_i \| ID_i)$,	Retrieves R_{gw} corresponding HID_i	
$HPW_i = h(RN_u \| UR_i \| ID_i \| PW_i)$,	Computes $\alpha_i^{'} = h(HID_i \| X_{GWN} \| R_{gw})$,	
$\alpha_i = \beta_i \oplus HPW_i$,	$h(N_u \| \alpha_i^{'}) = h(T_1 \| \alpha_i \| PSID_j) \oplus M_1$,	
$\gamma_i^* = h(HID_i \| h(ID_i \| HPW_i) \| \alpha_i)$.	$M_2^* = h(HID_i \| h(N_u \| \alpha_i^{'}) \| PSID_j)$.	
Checks $\gamma_i^* \stackrel{?}{=} \gamma_i$. If so,	Checks $M_2^* \stackrel{?}{=} M_2$. If so,	
Generates a random nonce N_u and timestamp T_1.	Computes $HID_{inew} = M_3 \oplus h(h(N_u \| \alpha_i) \| T_1)$	Checks $\|T_2 - T_c\| < \Delta T$?
Computes $M_1 = h(N_u \| \alpha_i) \oplus h(T_1 \| \alpha_i \| PSID_j)$,	Fetches (C_j, y_{GWN}) corresponding to $PSID_j$	Computes $C_j = M_4 \oplus h(PSID_j \| K_j)$,
$M_2 = h(HID_i \| h(N_u \| \alpha_i) \| PSID_j)$.	Generates a random nonce N_g and timestamp T_2.	$PUF(C_j) = R_j$,
Picks new pseudo identity $HID_{inew} = h(UR_i \| ID_i \| N_u)$,	Computes $K_j = h(PSID_j \| X_{GWN} \| y_{GWN})$,	$Rep(R_j, SNP_j) = SNR_j$,
and computes $M_3 = HID_{inew} \oplus h(h(N_u \| \alpha_i) \| T_1)$.	$M_4 = C_j \oplus h(PSID_j \| K_j)$,	$HS_j = h(SID_j \| SNR_j)$,
$\{HID_i, PSID_j, M_1, M_2, M_3, T_1\}$	$M_5 = h(h(N_u \| \alpha_i) \| N_g) \oplus h(PSID_j \| HS_j \| K_j)$,	$h(h(N_u \| \alpha_i) \| N_g) = M_5 \oplus h(PSID_j \| HS_j \| K_j)$,
$\longrightarrow$	$M_6 = h(h(h(N_u \| \alpha_i) \| N_g) \| T_2 \| HS_j)$.	$M_6^* = h(h(h(N_u \| \alpha_i) \| N_g) \| T_2 \| HS_j)$.
(via insecure channel)	$\{M_4, M_5, M_6, T_2\}$	Checks $M_6^* \stackrel{?}{=} M_6$. If so,
	$\longrightarrow$	Generates a timestamp T_3
	(via insecure channel)	Computes $SK = h(PSID_j \| h(h(N_u \| \alpha_i) \| N_g) \| K_j)$,
		$M_7 = h(SK \| T_3 \| K_j \| HS_j)$.
	Checks $\|T_3 - T_c\| < \Delta T$?	$\{M_7, T_3\}$
	Computes $SK = h(PSID_j \| h(h(N_u \| \alpha_i) \| N_g) \| K_j)$,	$\longleftarrow$
	$M_7^* = h(SK \| T_3 \| K_j \| HS_j)$.	(via insecure channel)
	Checks $M_7^* \stackrel{?}{=} M_7$. If so,	
	Computes $\alpha_{inew} = h(HID_{inew} \| X_{GWN} \| N_g)$,	
	$M_8 = SK \oplus h(N_u \| \alpha_i)$,	
Computes $SK = M_8 \oplus h(N_u \| \alpha_i)$,	$M_9 = \alpha_{inew} \oplus h(HID_{inew} \| HID_i \| h(N_u \| \alpha_i))$,	
$\alpha_{inew} = M_9 \oplus h(HID_{inew} \| HID_i \| h(N_u \| \alpha_i))$,	$M_{10} = h(\alpha_{inew} \| SK \| HID_{inew})$.	
$M_{10}^* = h(\alpha_{inew} \| SK \| HID_{inew})$.	If session key agreement is successful,	
Checks $M_{10}^* \stackrel{?}{=} M_{10}$. If so,	Updates $\{HID, R_{gw}\}$ to $\{HID_{inew}, N_g\}$	
Computes $\beta_{inew} = \alpha_{inew} \oplus HPW_i$,	$\{M_8, M_9, M_{10}\}$	
$\gamma_{inew} = h(HID_{inew} \| h(ID_i \| HPW_i) \| \alpha_{inew})$.	$\longleftarrow$	
Updates $\beta_{inew}, \gamma_{inew}$ and HID_{inew}	(via insecure channel)	

Figure 4. Login and authentication phase.

Step AP1: U_i inserts SC and inputs ID_i, PW_i, BIO_i. SC computes $Rep(BIO_i, UP_i) = UR_i$, $RN_u = L_i \oplus h(UR_i \| PW_i)$, $HID_i = h(UR_i \| ID_i)$, $HPW_i = h(RN_u \| UR_i \| ID_i \| PW_i)$, $\alpha_i = \beta_i \oplus HPW_i$, and $\gamma_i^* = h(HID_i \| h(ID_i \| HPW_i) \| A_i)$. Then, SC checks $\gamma_i^* \stackrel{?}{=} \gamma_i$. If it holds, U_i generates a random nonce N_u and timestamp T_1. U_i computes $M_1 = h(N_u \| A_i) \oplus h(T_1 \| A_i \| PSID_j)$ and $M_2 = h(HID_i \| h(N_u \| \alpha_i) \| PSID_j)$. U_i picks new pseudo identity $HID_{inew} = h(UR_i \| ID_i \| N_u)$ and computes $M_3 = HID_{inew} \oplus h(h(N_u \| \alpha_i) \| T_1)$. Then, U_i sends $\{HID_i, PSID_j, M_1, M_2, M_3, T_1\}$ to GWN through insecure channels.

Step AP2: Upon receiving the message $\{HID_i, PSID_j, M_1, M_2, M_3, T_1\}$, GWN checks $|T_1 - T_c| < \Delta T$?. If it holds, GWN retrieves R_{gw} from its database and calculates $\alpha_i^{'} = h(HID_i \| X_{GWN} \| R_{gw})$, $h(N_u \| \alpha_i^{'}) = h(T_1 \| \alpha_i \| PSID_j) \oplus M_1$, and $M_2^* = h(HID_i \| h(N_u \| \alpha_i^{'}) \| PSID_j)$. GWN checks $M_2^* \stackrel{?}{=} M_2$. If it is not correct, then GWN terminates the session. Otherwise, GWN calculates $HID_{inew} = M_3 \oplus h(h(N_u \| \alpha_i) \| T_1)$. Then, GWN fetches (C_j, y_{GWN}) corresponding to $PSID_j$. GWN generates a random nonce N_g and timestamp T_2. GWN computes $K_j = h(PSID_j \| X_{GWN} \| y_{GWN})$, $M_4 = C_j \oplus h(PSID_j \| K_j)$, $M_5 = h(h(N_u \| \alpha_i) \| N_g) \oplus h(PSID_j \| HS_j \| K_j)$, and $M_6 = h(h(h(N_u \| \alpha_i) \| N_g) \| T_2 \| HS_j)$. After that, GWN sends $\{M_4, M_5, M_6, T_2\}$ to SN_j through an open channel.

Step AP3: After receiving the message $\{M_4, M_5, M_6, T_2\}$ from GWN, SN_j checks $|T_2 - T_c| < \Delta T$?. If it holds, SN_j computes $C_j = M_4 \oplus h(PSID_j \| K_j)$, $PUF(C_j) = R_j$, $Rep(R_j, SNP_j) = SNR_j$, $HS_j = h(SID_j \| SNR_j)$, $h(h(N_u \| \alpha_i) \| N_g) = M_5 \oplus h(PSID_j \| HS_j \| K_j)$, and $M_6^* = h(h(h(N_u \| \alpha_i) \| N_g) \| T_2 \| HS_j)$. SN_j checks $M_6^* \stackrel{?}{=} M_6$. If it corrects, SN_j generates a timestamp T_3 and calculates $SK = h(PSID_j \| h(h(N_u \| \alpha_i) \| N_g) \| K_j)$. SN_j computes $M_7 = h(SK \| T_3 \| K_j \| HS_j)$ and sends $\{M_7, T_3\}$ to GWN.

Step AP4: When GWN receives the message $\{M_7, T_3\}$, GWN checks $|T_3 - T_c| < \Delta T$?. If it holds, GWN computes the session key $SK = h(PSID_j \| h(h(N_u \| \alpha_i) \| N_g) \| K_j)$, and computes $M_7^* = h(SK \| T_3 \| K_j \| HS_j)$. Then, GWN checks $M_7^* \stackrel{?}{=} M_7$. If they are same, GWN computes $\alpha_{inew} = h(HID_{inew} \| X_{GWN} \| N_g)$, $M_8 = SK \oplus h(N_u \| \alpha_i)$, $M_9 = \alpha_{inew} \oplus h(HID_{inew} \| HID_i \| h(N_u \| \alpha_i))$, and $M_{10} = h(\alpha_{inew} \| SK \| HID_{inew})$. GWN sends the message $\{M_8, M_9, M_{10}\}$. If session key agreement is successful, GWN updates $\{HID_i, R_{gw}\}$ to $\{HID_{inew}, N_g\}$. Otherwise, GWN keeps HID_i.

Step AP5: When U_i receives the message $\{M_8, M_9, M_{10}\}$, U_i calculates $SK = M_8 \oplus h(N_u||\alpha_i)$, and computes $\alpha_{inew} = M_9 \oplus h(HID_{inew}||HID_i||h(N_u||\alpha_i))$, and $M_{10}^* = h(\alpha_{inew}||SK||HID_{inew})$. U_i checks $M_{10}^* \stackrel{?}{=} M_{10}$. If they are same, U_i computes $\beta_{inew} = \alpha_{inew} \oplus HPW_i$ and $\gamma_{inew} = h(HID_{inew}||h(ID_i||HPW_i)||\alpha_{inew})$. Then, U_i updates $\beta_{inew}, \gamma_{inew}$ and HID_{inew}. Finally, U_i, GWN, and SN_j agrees the same session key SK.

6.4. User's Password and Biometrics Update Phase

U_i may want to change their password and biometrics. To reduce computation and communication costs, we propose this phase to be executed locally without additional connections with GWN.

Step 1: U_i inserts their SC and inputs ID_i, PW_i, and biometrics BIO_i. Then, SC computes $Rep(BIO_i, UP_i) = UR_i$, $RN_u = L_i \oplus h(UR_i||PW_i)$, $HID_i = h(UR_i||ID_i)$, $HPW_i = h(RN_u||UR_i||ID_i||PW_i)$, $\alpha_i = \beta_i \oplus HPW_i$, and $\gamma_i^* = h(HID_i||h(ID_i||HPW_i)||\alpha_i)$. SC checks $\gamma_i = \gamma_i^*$. If it corrects, SC asks U_i to input a new biometrics BIO_{inew} and a new password PW_{inew}.

Step 2: U_i inputs a new biometrics BIO_{inew} and a new password PW_{inew}. SC proceeds to compute parameters $Gen(BIO_{inew}) = (UR_{inew}, UP_{inew})$, $HPW_{inew} = h(RN_u||UR_{inew}||ID_i||PW_{inew})$, $L_{inew} = h(UR_{inew} \,||\, PW_{inew}) \oplus RN_u$, $RSP_{inew} = h(ID_i||HPW_{inew})$, $\beta_{inew} = \alpha_i \oplus HPW_i \oplus HPW_{inew}$, and $\gamma_{inew} = h(HID_i||RSP_{inew}||\alpha_i)$. Then, SC replaces $\beta_i, \gamma_i, L_i, UP_i$ with $\beta_{inew}, \gamma_{inew}, L_{inew}, UP_{inew}$.

7. Security Analysis

This section analyzes the security of the proposed protocol. We prove that session key agreement and mutual authentication of our protocol can be securely achieved through the commonly used ROR model and BAN logic. Through AVISPA simulation tools, we show that the protocol is secure against replay and MITM attacks. At last, through informal security analysis, we demonstrate that the proposed protocol is secure against a variety of attacks.

7.1. ROR Model

Thorugh the ROR model, we demonstrate the session key security of the proposed protocol [43–45]. We present the brief explanation of the ROR model. In the ROR model of our protocol, there are three participants $\mathcal{P}^t$, which are user node $\mathcal{P}_{U_i}^{t_1}$, gateway node $\mathcal{P}_{GWN}^{t_2}$, and sensor node $\mathcal{P}_{SN_j}^{t_3}$. t_1, t_2, and t_3 are the instances for U_i, GWN, and SN_j, respectively. We assume that $\mathcal{A}$ can intercept, eavesdrop, delete, or modify messages exchanged via an open wireless channel. Additionally, $\mathcal{A}$ can conduct security attacks through various queries, such as $Execute$, $CorruptSC$, $Reveal$, $Send$, and $Test$. Detailed descriptions of these queries are as follows.

- $Execute(\mathcal{P}_{U_i}^{t_1}, \mathcal{P}_{GWN}^{t_2}, \mathcal{P}_{SN_j}^{t_3})$: $\mathcal{A}$ can conduct this query for obtaining transmitted messages via public channels between $\mathcal{P}_{U_i}^{t_1}$, $\mathcal{P}_{GWN}^{t_2}$, and $\mathcal{P}_{SN_j}^{t_3}$.
- $CorruptSC(\mathcal{P}_{U_i}^{t_1})$: $CorruptSC$ indicates that the adversary can extract secret data stored in SC of $\mathcal{P}_{U_i}^{t_1}$.
- $Reveal(\mathcal{P}^t)$: $\mathcal{A}$ is able to reveal the current session key SK between $\mathcal{P}_{U_i}^{t_1}$, $\mathcal{P}_{GWN}^{t_2}$, and $\mathcal{P}_{SN_j}^{t_3}$ by executing this query. SK is safe if $\mathcal{A}$ fails to reveal SK using this query.
- $Send(\mathcal{P}^t, M)$: Using the $Send$ query, an adversary is able to send a message to participants and receive response messages.
- $Test(\mathcal{P}^t)$: An unbiased coin uc is flipped to start the game, and the result is only known to $\mathcal{A}$. $\mathcal{A}$ uses this result to determine the $Test$. When $\mathcal{A}$ runs the $Test$ query, $\mathcal{P}^t$ returns SK for uc = 1 or a random number for uc = 0. Otherwise, it returns a null ($\perp$).

$\mathcal{A}$ must distinguish the result value after $\mathcal{A}$ conducts $Test$ query over $\mathcal{P}^t$. $\mathcal{A}$ checks the consistency of the random bit uc using results of the $Test$ query. $\mathcal{A}$ is able to win the

game if the guessed bit uc' equals uc. Additionally, $\mathcal{P}^t$ has access to the collision-resistant cryptographic one-way hash function $h(\cdot)$, which is modeled as a random oracle, *Hash*.

Security Proof

Theorem 1. *An adversary $\mathcal{A}$ attempts to calculate SK in polynomial time. Let $Adv_{\mathcal{A}}^{our}$ be the advantage that $\mathcal{A}$ can break the session key security of the proposed protocol. Then, we obtain the following.*

$$Adv_{\mathcal{A}}^{our} \leq \frac{q_{hash}^2}{|Hash|} + \frac{q_{puf}^2}{|PUF|} + 2max\{C \cdot q_{send}^s, \frac{q_{send}}{2^{l_D}}\}$$

$|PUF|$ and $|Hash|$ indicate that they are the span spaces of the PUF function $PUF(\cdot)$ and the hash function $h(\cdot)$, respectively. q_{send}, q_{hash}, and q_{puf} are the number of Send, Hash, and PUF queries, respectively. In addition, l_D is the number of bits in biometric BIO_i of U_i, and C and s denote Zipf's parameters.

Proof. The five games, GM_i, where $i \in [0, 4]$, are conducted to prove the security of *SK* of the proposed protocol. $Succ_{\mathcal{A},i}$ indicates the event in which $\mathcal{A}$ wins GM_i by guessing the random bit *uc* correctly. We represent the probability that $\mathcal{A}$ wins the game GM_i as $Pr[Succ_{\mathcal{A},GM_i}]$. This is followed by the description of each game.

GM_0: $\mathcal{A}$ executes a real attack to our protocol. $\mathcal{A}$ chooses a random bit *uc* at the beginning of GM_0. The following advantage of $\mathcal{A}$ is about this game.

$$Adv_{\mathcal{A}}^{our} = |2Pr[Succ_{\mathcal{A},GM_0}] - 1| \tag{4}$$

GM_1: $\mathcal{A}$ executes the $Execute(\mathcal{P}_{U_i}^{t_1}, \mathcal{P}_{GWN}^{t_2}, \mathcal{P}_{SN_j}^{t_3})$ query and eavesdrops messages $< HID_i, PSID_j, M_1, M_2, M_3, T_1 >$, $< M_4, M_5, M_6, T_2 >$, $< M_7, T_3 >$, and $< M_8, M_9, M_{10} >$. After that, $\mathcal{A}$ performs *Reveal* and *Test* queries to verify whether the derived *SK* is real. In the proposed protocol, $SK = h(PSID_j||h(h(N_u||\alpha_i)||N_g)||K_j)$ is made up of long-term and short-term secrets. To derive *SK*, $\mathcal{A}$ needs to know the identities and random nonces of U_i, GWN, and SN_j. As a result, $\mathcal{A}$ cannot increase the winning probability of GM_1. Therefore, the probabilities of GM_0 and GM_1 are indistinguishable.

$$Pr[Succ_{\mathcal{A},GM_1}] = Pr[Succ_{\mathcal{A},GM_0}] \tag{5}$$

GM_2: In this game, $\mathcal{A}$ executes *Hash* and *Send* queries to obtain the session key. $\mathcal{A}$ attempts to attack by modifying the exchanged message. However, all messages are masked with one-way hash function $h(\cdot)$, random nonces, and secret credentials. $\mathcal{A}$ cannot derive any information due to a computationally infeasible problem of $h(\cdot)$. Hence, using the birthday paradox, we can get the following equation.

$$|Pr[Succ_{\mathcal{A},GM_2}] - Pr[Succ_{\mathcal{A},GM_1}]| \leq \frac{q_{hash}^2}{2|Hash|} \tag{6}$$

GM_3: This game is performed in analogy as described in GM_2. $\mathcal{A}$ executes *Send* and *PUF* queries. However, the probability obtained by the *PUF* query is similar with the *Hash* query since the physical function $PUF(\cdot)$ has security properties mentioned in Section 3.2. Therefore, we are able to acquire the following equation.

$$|Pr[Succ_{\mathcal{A},GM_3}] - Pr[Succ_{\mathcal{A},GM_2}]| \leq \frac{q_{puf}^2}{2|PUF|} \tag{7}$$

GM_4: In the final game GM_4, $\mathcal{A}$ tries to get *SK* with the *CorruptSC* query. With *CorruptSC* query, $\mathcal{A}$ is able to extract sensitive values $\{\beta_i, \gamma_i, L_i, UP_i\}$ stored in the smart card of U_i, which are expressed as $\beta_i = \alpha_i \oplus HPW_i$, $\gamma_i = h(HID_i||RSP_i||\alpha_i)$, and $L_i = h(UR_i||PW_i) \oplus RN_u$. For computing $SK = h(PSID_j||h(h(N_u||\alpha_i)||N_g)||K_j)$, $\mathcal{A}$ should guess these parame-

ters from the extracted values since $\mathcal{A}$ has no knowledge of identity ID_i, password PW_i, and biometric BIO_i. However, it is a computationally infeasible task for $\mathcal{A}$ to guess ID_i, PW_i, and BIO_i simultaneously. In conclusion, GM_3 and GM_4 are indistinguishable. We can derive the following result by utilizing Zipf's law.

$$|Pr[Succ_{\mathcal{A},GM_4}] - Pr[Succ_{\mathcal{A},GM_3}]| \leq max\{C \cdot q_{send}^s, \frac{q_{send}}{2^{l_D}}\} \tag{8}$$

After all games are completed, $\mathcal{A}$ must guess the uc to win the game. Therefore, we obtain the following equation.

$$Pr[Succ_{\mathcal{A},GM_4}] = \frac{1}{2} \tag{9}$$

By combining (4)–(9), we obtain the result using the triangular inequality as follows.

$$\begin{aligned}
\frac{1}{2}Adv_{\mathcal{A}}^{our} &= |Pr[Succ_{\mathcal{A},GM_0} - \frac{1}{2}]| = |Pr[Succ_{\mathcal{A},GM_1} - \frac{1}{2}]| \\
&= |Pr[Succ_{\mathcal{A},GM_1}] - Pr[Succ_{\mathcal{A},GM_4}]| \\
&= |Pr[Succ_{\mathcal{A},GM_1}] - Pr[Succ_{\mathcal{A},GM_4}]| \\
&\leq |Pr[Succ_{\mathcal{A},GM_1}] - Pr[Succ_{\mathcal{A},GM_3}]| \\
&\quad + |Pr[Succ_{\mathcal{A},GM_3}] - Pr[Succ_{\mathcal{A},GM_4}]| \\
&\leq |Pr[Succ_{\mathcal{A},GM_1}] - Pr[Succ_{\mathcal{A},GM_2}]| \\
&\quad + |Pr[Succ_{\mathcal{A},GM_2}] - Pr[Succ_{\mathcal{A},GM_3}]| \\
&\quad + |Pr[Succ_{\mathcal{A},GM_3}] - Pr[Succ_{\mathcal{A},GM_4}]| \\
&\leq \frac{q_{hash}^2}{2|Hash|} + \frac{q_{puf}^2}{2|PUF|} + max\{C \cdot q_{send}^s, \frac{q_{send}}{2^{l_D}}\}
\end{aligned} \tag{10}$$

Finally, the desired result can be obtained by multiplying both sides of Equation (10) by two.

$$Adv_{\mathcal{A}}^{our} \leq \frac{q_{hash}^2}{|Hash|} + \frac{q_{puf}^2}{|PUF|} + 2max\{C \cdot q_{send}^s, \frac{q_{send}}{2^{l_D}}\} \tag{11}$$

Therefore, we prove Theorem 1. □

7.2. BAN Logic

BAN logic is a widely used mathematical proof method for demonstrating mutual authentication in security schemes [46,47]. With BAN logic, we prove that the proposed protocol can ensure mutual authentication. The notations of BAN logic are described in Table 1.

7.2.1. Rules

The rules used in BAN logic are as follows.

- Nonce verification rule (NVR):

$$\frac{E| \equiv \#(T), E| \equiv K| \sim T}{E| \equiv K| \equiv T}$$

- Message meaning rule (MMR):

$$\frac{E| \equiv E \overset{Skey}{\longleftrightarrow} K, E \triangleleft \{T\}_{Skey}}{E| \equiv K| \sim T}$$

- Jurisdiction rule (JR):

$$\frac{E| \equiv K| \Rightarrow T, E| \equiv K| \equiv T}{E| \equiv T}$$

- Freshness rule (FR):

$$\frac{E| \equiv \#(T)}{E| \equiv \#(T, S)}$$

- Belief rule (BR):

$$\frac{E| \equiv (T, S)}{E| \equiv T}$$

Table 1. BAN Logic Notation.

Notation	Description
$Skey$	Secret key
$E| \equiv T$	E **believes**statement T
$\#T$	Statement T is **fresh**
$E \triangleleft T$	E **receives** statement T
$E| \sim T$	E once **said** T
$E \Rightarrow T$	E **controls** statement T
$< T >_S$	Statement T is **combined** with secret statement S
$\{T\}_{Skey}$	Statement T is **masked** by $Skey$
$E \overset{Skey}{\longleftrightarrow} K$	E and K **share** $Skey$ to communicate with each other

7.2.2. Goals for Mutual Authentication

To prove that the proposed protocol provides mutual authentication, we present the following goals.

Goal 1: $U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} GWN)$

Goal 2: $U_i| \equiv GWN| \equiv (U_i \overset{SK}{\longleftrightarrow} GWN)$

Goal 3: $GWN| \equiv (U_i \overset{SK}{\longleftrightarrow} GWN)$

Goal 4: $GWN| \equiv U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} GWN)$

Goal 5: $SN_j| \equiv (SN_j \overset{SK}{\longleftrightarrow} GW)$

Goal 6: $SN_j| \equiv GWN| \equiv (SN_j \overset{SK}{\longleftrightarrow} GWN)$

Goal 7: $GWN| \equiv (SN_j \overset{SK}{\longleftrightarrow} GWN)$

Goal 8: $GWN| \equiv SN_j| \equiv (SN_j \overset{SK}{\longleftrightarrow} GWN)$

7.2.3. Idealized Form of Exchanged Messages

We describe the idealized form of BAN logic as the message exchanged in the authentication phase as follows.

M_1**:** $U_i \rightarrow GWN : \{PSID_j, h(N_u||\alpha_i)\}_{\alpha_i}$

M_2**:** $GWN \rightarrow SN_j : \{h(h(N_u||\alpha_i)||h(N_g))\}_{K_j}$

M_3**:** $SN_j \rightarrow GWN : \{SK, T_3\}_{K_j}$

M_4**:** $GWN \rightarrow U_i : \{SK\}_{h(N_u||\alpha_i)}$

7.2.4. BAN Logic Initial State Assumptions

We construct all the considered assumptions as follows.

A_1: $GWN| \equiv U_i \overset{\alpha_i}{\longleftrightarrow} GW$

A_2: $GWN| \equiv \#(N_u)$

A_3: $SN_j| \equiv GWN \overset{K_j}{\longleftrightarrow} SN_j$

A_4: $SN_j| \equiv \#(N_g)$

A_5: $GWN| \equiv GWN \overset{K_j}{\longleftrightarrow} SN_j$

A_6: $GWN| \equiv \#(T_3)$

A_7: $U_i| \equiv U_i \overset{h(N_u||\alpha_i)}{\longleftrightarrow} GWN$

A_8: $U_i| \equiv \#(N_g)$

A_9: $U_i| \equiv GWN| \Rightarrow (U_i \overset{SK}{\longleftrightarrow} GWN)$

A_{10}: $GWN| \equiv U_i| \Rightarrow (U_i \overset{SK}{\longleftrightarrow} GWN)$

A_{11}: $SN_j| \equiv GWN| \Rightarrow (SN_j \overset{SK}{\longleftrightarrow} GWN)$

A_{12}: $GWN| \equiv SN_j| \Rightarrow (SN_j \overset{SK}{\longleftrightarrow} GWN)$

7.2.5. Proof of Providing Mutual Authentication

We will now prove that our protocol can guarantee mutual authentication with an idealized form, predefined BAN logic rules, and assumptions. The proof process is as follows.

Step 1: S_1 is obtained from M_1.

$$S_1 : GWN \triangleleft \{PSID_j, h(N_u||\alpha_i)\}_{\alpha_i}$$

Step 2: S_2 is obtained from the MMR using S_1 and A_1.

$$S_2 : GWN| \equiv U_i| \sim \{PSID_j, h(N_u||\alpha_i)\}_{\alpha_i}$$

Step 3: S_3 can be gained from the FR with S_2 and A_2.

$$S_3 : GWN| \equiv \#(PSID_j, h(N_u||\alpha_i))$$

Step 4: S_4 can be acquired by applying the NVR with S_2 and S_3.

$$S_4 : GWN| \equiv U_i| \equiv (PSID_j, h(N_u||\alpha_i))$$

Step 5: S_5 is obtained from M_2.

$$S_5 : SN_j \triangleleft \{h(h(N_u||\alpha_i)||h(N_g))\}_{K_j}$$

Step 6: S_6 is gained from MMR using S_5 and A_3.

$$S_6 : SN_j| \equiv GWN| \sim \{h(h(N_u||\alpha_i)||h(N_g))\}_{K_j}$$

Step 7: S_7 can be obtained by applying FR with S_6 and A_4.

$$S_7 : SN_j| \equiv \#(h(h(N_u||\alpha_i)||h(N_g)))$$

Step 8: S_8 can be obtained from NVR with S_6 and S_7.

$$S_8 : SN_j| \equiv GWN| \equiv (h(h(N_u||\alpha_i)||h(N_g)))$$

Step 9: From M_3, S_9 is obtained.

$$S_9 : GWN \triangleleft \{SK, T_3\}_{K_j}$$

Step 10: S_{10} is gained from MMR with S_9 and A_5.

$$S_{10} : GWN| \equiv SN_j| \sim \{SK, T_3\}_{K_j}$$

Step 11: S_{11} can be obtained by applying FR with S_{10} and A_6, since $SK = h(PSID_j||h(h(N_u||\alpha_i)||N_g)||K_j)$.

$$S_{11} : GWN| \equiv \#(SK, T_3)$$

Step 12: S_{12} can be obtained from NVR with S_{10} and S_{11}.

$$S_{12} : GWN| \equiv SN_j| \equiv (SK, T_3)$$

Step 13: S_{13} is obtained from M_4.

$$S_{13} : U_i \triangleleft \{SK\}_{h(N_u||\alpha_i)}$$

Step 14: S_{14} is obtained from MMR with S_{13} and A_7.

$$S_{14} : U_i| \equiv GWN| \sim \{SK\}_{h(N_u||\alpha_i)}$$

Step 15: S_{15} can be obtained from FR with S_{14} and A_8, since $SK = h(PSID_j||h(h(N_u||\alpha_i)||N_g)||K_j)$.

$$S_{15} : U_i| \equiv \#(SK)$$

Step 16: S_{16} can be obtained by using NVR on S_{14} and S_{15}.

$$S_{16} : U_i| \equiv GWN| \equiv (SK)$$

Step 17: S_{17} and S_{18} can be obtained from S_8 and S_{12} since $SK = h(PSID_j||h(h(N_u||\alpha_i)||N_g)||K_j)$.

$$S_{17} : SN_j| \equiv GWN| \equiv (SN_j \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 6)}$$

$$S_{18} : GWN| \equiv SN_j| \equiv (SN_j \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 8)}$$

Step 18: S_{19} and S_{20} can be obtained from JR with S_{17}, S_{18}, A_{11}, and A_{12}.

$$S_{19} : SN_j| \equiv (SN_j \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 5)}$$

$$S_{20} : GWN| \equiv (SN_j \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 7)}$$

Step 19: S_{21} and S_{22} can be obtained from S_4 and S_{16} since $SK = h(PSID_j||h(h(N_u||\alpha_i)||N_g)||K_j)$.

$$S_{21} : U_i| \equiv GWN| \equiv (U_i \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 2)}$$

$$S_{22} : GWN| \equiv U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 4)}$$

Step 20: S_{23} and S_{24} can be obtained by applying JR from S_{21}, S_{22}, A_9, and A_{10}.

$$S_{23} : U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 1)}$$

$$S_{24} : GWN| \equiv (U_i \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 3)}$$

We prove that the proposed scheme meets all the goals in Section 7.2.2. Therefore, the proposed protocol ensures secure mutual authentication.

7.3. AVISPA Simulation Analysis

We use the "AVISPA Simulation Tool" [13] in this section to validate our proposed system security against man-in-the-middle and replay attacks.

In AVISPA, there are four backends: "Tree Automata based on Automatic Approximations for "Tree Automata based on Automatic Approximations for Analysis of Security Protocols (TA4SP)", "SAT based model checker (SATMC)", "On-the-fly-mode-checker (OFMC)", and "Const- raint-logic-based Attack Searcher" (CL-AtSe)". Among these, SATMC and TA4SP backends can not aid "bitwise exclusive OR (XOR)". However, since our system has an XOR operation, two backends are not suitable for analysis. Therefore, we adopt two backends, OFMC and CL-AtSe, which support XOR operations, and use them for analysis. In the proposed system, "High-Level Protocol Specification Language (HLPSL)", a language supported by AVISPA, is used to implement the basic roles of U_i, GWN and SN_j. Figure 5 shows the HLPSL implementation of the role user.

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

played_by UA
def=
local State: nat,
   IDi, PWi, HIDi, HPWi, RNu, UPi, URi, RSPi, RNgw, Ai, Bi, Ci, Xgwn, Li : text,
   SIDj, RNsn, Cj, SNRj, SNPj, Reqj, HSj, Ygwn, PSIDj, Kj : text,
   Nu, T1, M1, M2, M3, HIDinew, Ng, T2, M4, M5, M6, SK, M7, T3: text,
   Ainew, M8, M9, M10, Binew, Cinew : text
const sp1, sp2, sp3, sp4, sp5,sp6, ua_ga_ni, ga_sa_ng, ga_ua_ng, sa_ga_ng: protocol_id
init State := 0
transition

%%%%%%%%%%%%%Registration phase
1. State = 0 ∧ RCV(start) =|>
State' := 1 ∧ URi' := new() ∧ UPi' := new() ∧ RNu' := new() ∧ HIDi' := H(URi'.IDi)
           ∧ HPWi' := H(RNu'.URi'.IDi.PWi) ∧ RSPi' := H(IDi.HPWi')
           ∧ SND({HIDi'.RSPi'.HPWi'}_SKuaga)
           ∧ secret({PWi, URi'}, sp1, {UA})
           ∧ secret({HPWi'}, sp2, {UA,GA})

%%%%%%%%%%%%%Recieve smartcard
2. State = 1 ∧ RCV
({xor(H(H(URi'.IDi).Xgwn.RNgw'),H(RNu'.URi'.IDi.PWi)).H(H(URi'.IDi).H(IDi.H(RNu'.URi'.IDi.PWi)).H(H(URi'.IDi).Xgwn.RNgw')).H(SIDj.RNsn')}_
SKuaga)=|>
 State' := 2 ∧ Li':= xor(H(URi'.PWi),RNu')
%%%%%%%%%%%%%Login & Authentication phase
           ∧ Nu' := new() ∧ T1' := new()
           ∧ M1' := xor(H(Nu'.H(H(URi'.IDi).Xgwn.RNgw')),H(T1'.H(H(URi'.IDi).Xgwn.RNgw').H(SIDj.RNsn')))
           ∧ M2' := H(H(URi'.IDi).H(Nu'.H(H(URi'.IDi).Xgwn.RNgw')).H(SIDj.RNsn'))
           ∧ HIDinew' := H(URi'.IDi.Nu')
           ∧ M3' := xor(HIDinew',H(H(Nu'.H(H(URi'.IDi).Xgwn.RNgw')).T1'))
           ∧ SND(H(URi'.IDi).H(SIDj.RNsn').M1'.M2'.M3'.T1')
           ∧ witness(UA,GA,ua_ga_ni,Nu')

3. State = 2 ∧ RCV(xor(H(H(SIDj.RNsn').H(H(Nu'.H(H(URi'.IDi).Xgwn.RNgw')).Ng').H(H(SIDj.RNsn').Xgwn.Ygwn)),
H(Nu'.H(H(URi'.IDi).Xgwn.RNgw'))).xor(H(H(URi'.IDi.Nu').Xgwn.Ng'),H(H(URi'.IDi.Nu').H(URi'.IDi).H(Nu'.H(H(URi'.IDi).Xgwn.RNgw'))))).H(H(H(U
Ri'.IDi.Nu').Xgwn.Ng').H(H(SIDj.RNsn').H(H(Nu'.H(H(URi'.IDi).Xgwn.RNgw')).Ng').H(H(SIDj.RNsn').Xgwn.Ygwn)).H(URi'.IDi.Nu'))) =|>
State' := 3 ∧ SK' := H(H(SIDj.RNsn').H(H(Nu'.H(H(URi'.IDi).Xgwn.RNgw')).Ng').H(H(SIDj.RNsn').Xgwn.Ygwn))∧ RNu' := new()
                  ∧ Binew' := xor(H(H(URi'.IDi.Nu').Xgwn.Ng'),H(RNu'.URi'.IDi.PWi))
                  ∧ Cinew' := H(H(URi'.IDi.Nu').H(IDi.H(RNu'.URi'.IDi.PWi)).H(H(URi'.IDi.Nu').Xgwn.Ng'))
        ∧ request(GA,UA,ga_ua_ng,Ng')
end role
```

Figure 5. HLPSL specification for user.

At transition 1, U_i sends the request message $\{HID_i, RSP_i, HPW_i\}$ to GWN using the SND operation and $SKuaga$, which means the secure channel. The declaration $secret(\{PWi, URi\}, sp1, \{UA\})$ means that the password PW_i and biometrics UR_i is only known to U_i. The declaration $secret(\{HPWi\}, sp2, \{UA, GA\})$ means that HPW_i is only known to U_i and GWN.

At transition 2, U_i receives the smartcard. In login and authentication phase, U_i sends the message $\{HID_i, PSID_j, M_1, M_2, M_3, T_1\}$ to GWN through insecure channels. The declaration $witness(UA, GA, ua\_ga\_ni, Nu')$ means that U_i generates a random nonce N_u for GWN.

At transition 3, U_i receives the message $\{M_8, M_9, M_{10}\}$ from GWN. The declaration $request(GA, UA, ga\_ua\_ng, Ng')$ specifies the GWN request to the U_i for checking the value of N_g.

The HLPSL of the gateway node and sensor node is implemented similarly to the user's HLPSL. In addition, it implements the "composite roles and goals for sessions and environment" of the proposed system through HLPSL. In the sessions and environment, it specifies whether secret maintenance and authentication of each value are successfully performed through *secret*, *witness*, and *request* declared in the HLPSL of each entity. AVISPA used in this section is a security validation simulation based on the DY model [37].

Figure 6 is a screen showing the intruder simulation step-by-step according to the HLPSL configured in the CL-AtSe mode. It is a simulation in which knowledge is leaked to the intruder one-by-one for each step. In addition, the intruder knows the message transmitted through the wireless channel. Although this information is leaked to the intruder, we can see that our protocol is safe, as shown in Figure 7. Therefore, for replay attack inspection, AVIPA backends (such as OFMC and CL-AtSe) first verify that a legitimate agent can execute a specific protocol. It then provides the intruder's knowledge of some legitimate sessions between legitimate agents. In addition, the DY model ensures that OFMC and CL-AtSe backends are capable of MITM attacks by intruders. Figure 7 gives the analysis results performed on the CL-ATse and OFMC backends. The results are shown in Figure 7 show that the proposed protocol is "safe" on the backends, which proves that our protocol is secure against replay and MITM attacks.

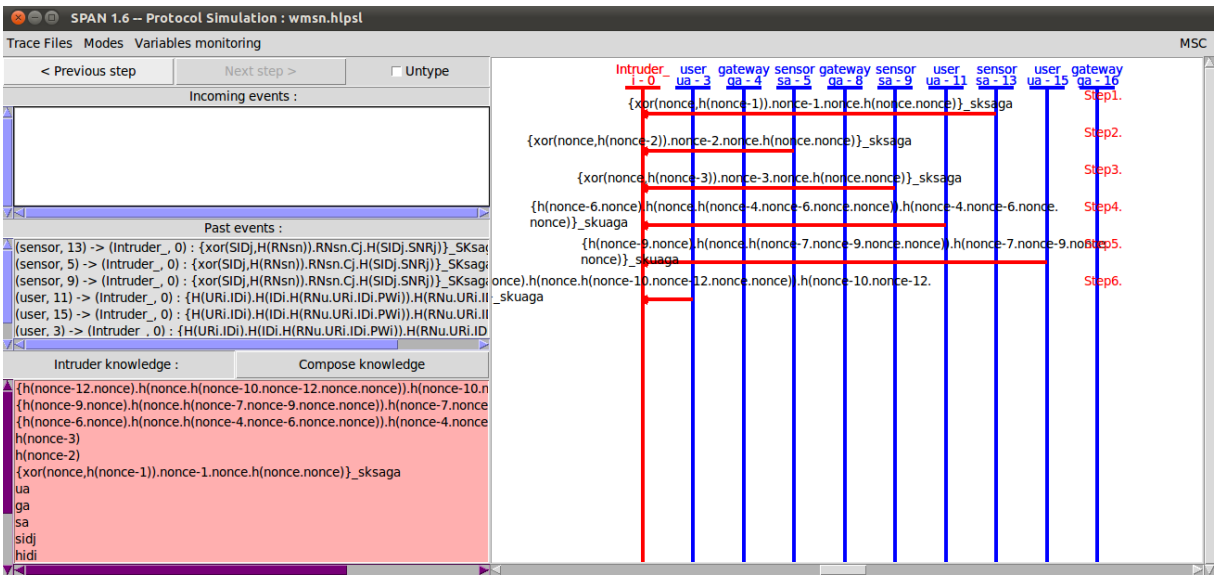

Figure 6. AVISPA implementation results.

% OFMC % Version of 2006/02/13 SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/span/span/testsuite/results/wmsn.if GOAL as_specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00s searchTime: 1.00s visitedNodes: 64 nodes depth: 6 plies	SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/span/span/testsuite/results/wmsn.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 8 states Reachable : 0 states Translation: 0.14 seconds Computation: 0.00 seconds

Figure 7. AVISPA simulation results.

7.4. Informal Security Analysis

We demonstrate through informal security analysis that the proposed protocol can provide various security features are secure against various attacks.

7.4.1. Offline Guessing Attack

Adversary ATT attempts to guess the user's identity or password from the values contained in the user's smartcard or messages on public channels. ATT can obtain sensitive information through the guessed ID/password. However, our protocol is secure against offline guessing attacks. ATT is able to obtain $\{\beta_i, \gamma_i, L_i, UP_i\}$ stored in SC through smartcard stolen attack. ATT needs to compute γ_i^* to guess the ID/password of a valid U_i. However, γ_i^* consists of HID_i and HPW_i. In order for ATT to calculate HID_i, it needs to know the user's biometric key UR_i and the user's ID. Furthermore, in order for ATT to figure out HPW_i, ATT must have the biometric key UR_i, as well as valid ID_i, PW_i and random nonce RN_u. Therefore, ATT cannot compute the user's HID_i and HPW_i according to the "computational infeasible problem". Therefore, since ATT cannot guess the user's ID and password, the proposed protocol can guarantee the resilience of offline guessing attacks.

7.4.2. Privacy Preserving and Anonymity

ATT may trace the use of services of U_i through an identity or pseudonymous ID or intercept personal information. However, our protocol guarantees privacy by preserving U_i and can provide anonymity. ATT tries to obtain U_i's real identity information through SC's information or transmitted messages. However, ATT cannot obtain the real identity or pseudo identity because these values are hidden in the hash function and UR_i. Although pseudo identity HID_i is transmitted via an open channel, HID_{inew} is updated when the authentication and key agree. Moreover, HID_{inew} is masked by N_u. Therefore, HID_i is always updated in every session. Therefore, the proposed protocol can preserve U_i's privacy and anonymity property.

7.4.3. Impersonation Attack

ATT attempts to impersonate U_i, GWN, and SN_j to obtain valid information. To obatin valid information, ATT must be able to calculate messages sent via wireless channels. However, messages sent to the public channel change every session due to the N_u, N_g, and timestamp values. Furthermore, ATT cannot compute a valid message because HID_i is also updated to HID_{inew} upon successful authentication. Therefore, our protocol is security from impersonation attacks.

7.4.4. Sensor Node Physical Capture Attack

ATT performs sensor node physical attack to acquire $\{SID_j, PSID_j, K_j, SNP_j\}$ stored in SN_j. However, ATT cannot compute the correct session key even if it gets the stored values. For ATT to compute the session key, $h(h(N_u||\alpha_i)||N_g) = M_5 \oplus h(PSID_j||HS_j||K_j)$

needs to be calculated. However, in the sensor node registration phase, the valid R_j and SNR_j cannot be obtained because the sensor randomly generates C_j, which is different from the value of each sensor. Because R_j is a value that PUF generates, it cannot be physically replicated. Furthermore, the compromise in SN_j does not help compute the session key between U_i and any other uncompromised medical sensor. Therefore, the proposed protocol is used to secure the sensor node's physical capture attack.

7.4.5. Replay and MITM Attack

ATT is able to obtain the information stored in SN_j and SC of valid U_i and can acquire messages sent to the public channel. However, Adv cannot count valid messages generated by U_i and SN_j, as mentioned in Sections 7.4.3 and 7.4.4. In addition, every message changes every session because of the N_u, N_g and timestamp values. Therefore, it can be said that our protocol is safe to replay MITM attacks.

7.4.6. Desynchronization Attack

An attacker could delay the updating of HID_{inew}, thereby interrupting the entity from being authenticated. These attacks are called desynchronization attacks. In our protocol, U_i picks up a new HID_{inew} during the login and authentication phase and passes it to the GWN. After that, GWN updates HID_i and RN_{gw} with HID_{inew} and N_g upon successful authentication, and transmits the related values to U_i. At this time, GWN maintains $\{HID_i, RN_{gw}\}$ if authentication is not successful. Moreover, if U_i does not succeed in authentication, the existing HID_i is kept, and the login and authentication phase is performed again. Therefore, even if the login and authentication phase is blocked by ATT, U_i and GWN can keep the original HID_i. Therefore, the proposed technique can resist desynchronization attacks.

7.4.7. Stolen Verifier Attack

Even if the information in the verfier table stored in the gateway is leaked to ATT, ATT must not be able to impersonate the user and sensor, and ATT must also be unable to calculate the session key. Assume that ATT obtains GWN's verification tables $\{HID_i, RN_{gw}\}$ and $\{PSID_j, HS_j, y_{gwn}, C_j\}$ to perform impersonation attacks or compute the SK. However, ATT cannot compute $\alpha_i = h(HID_i||X_{GWN}||RN_{gw})$ and $K_j = h(PSID_j||X_{GWN}||y_{GWN})$ without GWN's secret key X_{GWN}. Furthermore, owing to the nature of the PUF, ATT could not calculate $R_j = PUF(C_j)$. Thus, ATT cannot perform impersonation attack and compute SK. Therefore, the proposed protocol can be said to be resistant against a stolen verifier attack.

7.4.8. Perfect Forward Secrecy

Even if the private key of GWN is leaked to ATT, ATT should not be able to compute the session key of the previous session. Assuming that the private key X_{GWN} of GWN is leaked by ATT, ATT attempts to calculate a valid SK using the obtained X_{GWN}. However, in the registration phase of U_i and SN_j, α_i and K_j are masked with RN_{gw} and y_{GWN} is randomly generated by GWN. Therefore, ATT cannot compute valid α_i and K_j, so it is impossible to compute valid SK. Therefore, our protocol can guarantee complete forward secrecy.

7.4.9. Session-Specific Random Number Leakage Attack

Assume that N_u, N_g, a random nonce generated in the session, is leaked to ATT. Using this value, ATT will help compute SK. However, ATT is not able to calculate the valid $SK = h(PSID_j||h(h(N_u||\alpha_i)||N_g)||K_j)$. To calculate a valid SK, ATT needs to calculate α_i and K_j. In order to calculate α_i and K_j, ATT need to compute or obtain the values of X_{GWN}, y_{GWN}, and HPW_i, but it is impossible. Therefore, the proposed protocol is secure against session-specific random number leak attacks.

7.4.10. Ephmeral Secret Leakage Attack

According to the CK attack model, when long-term or short-term secrets are leaked to ATT, ATT can calculate a valid session key. In our protocol, ATT has acquired long-term secrets (e.g., X_{gwn} and y_{GWN}). In this way, the session key $SK = h(PSID_j||h(h(N_u||\alpha_i)||N_g)||K_j) = h(PSID_j||h(h(N_u||h(HID_i||X_{GWN}||RN_{gw}))||N_g)||h(PSID_j||X_{GWN}||y_{GWN}))$ includes X_{gwn} and y_{GWN} as well as N_u, RN_{gw}, N_g is also included. ATT cannot calculate the correct SK without knowing these values. Furthermore, according to Section 7.4.9, even if a short-term secret value is leaked, ATT cannot compute SK. Therefore, our method is resistant to ESL attacks.

7.4.11. Session Key Security and Mutual Authentication

ATT computes SK to obtain sensitive information or attempts mutual authentication by disguising itself as a valid entity. However, as discussed in Sections 7.4.7–7.4.10, ATT cannot compute a valid SK because of a "computationally infeasible problem". Additionally, in our proposed protocol, all entities verify each message and mutually authenticate each other. At this time, messages are changed every session due to random number and timestamp and are encrypted with long-term key and short-term key. Therefore, an attacker cannot impersonate a valid entity. Therefore, the proposed protocol guarantees secure session key security and mutual authentication.

8. Efficiency Analysis

We compare communication and computation costs, and security aspect with related protocols for showing the efficiency of our proposed protocol.

8.1. Functionality and Security Features Comparison

This section compares the proposed protocol to related protocols in replay and MITM, guessing, impersonation, device or sensor capture, and desynchronization attacks. We also compare the provision of security features such as forward secrecy and anonymity. Table 2 indicates that the proposed protocol meets all essential security for communication in a WMSN, whereas existing protocols do not satisfy all security requirements.

Table 2. Security and functional properties comparison.

Security Properties	Our Protocol	Yuanbing et al. [9]	Ali et al. [25]	Li et al. [28]	Masud et al. [23]
Replay attack	o	o	o	o	o
MITM attack	o	x	o	o	o
Guessing attack	o	x	o	o	x
Impersonation attack	o	x	o	x	x
Smart card stolen attack	o	x	o	o	-
Device or sensor capture attack	o	x	x	x	x
Desynchronization attack	o	-	x	-	-
Anonymity	o	x	o	x	x
Perfect forward secrecy	o	o	x	o	o
Using three factors	o	x	o	o	x
Using PUF	o	x	x	x	x
Secure mutual authentication	o	x	o	x	o

x: insecure against an attack; o: secure against an attack; -: not considered.

8.2. Computation Costs Comparison

The cryptographic computation costs in [30,48] are used for comparative analysis of the computational costs. For the computation cost of cryptographic functions, except PUF and fuzzy extractor, the PBC library (version 0.5.12) built in the GMP library is used on a personal computer environment with Intel Pentium Dual CPU E2200 2.20 GHz processor, 2048 MB RAM, and Ubuntu 12.04.1 LTS 32-bit operating system [48]. The computational costs of PUF and fuzzy functions are obtained in the environment of a single-core 798 MHz CPU with 256 MB RAM, adopting a code offset mechanism using BCH, assuming a 128-bit arbiter PUF [30]. Accordingly, we assumed the times for the notation of the cryptographic function and the computational cost of the function as follows: T_{hash}, T_{pm}, T_{enc}, T_{dec}, T_{fuzzy}, T_{puf}, and T_{rg} denote hash function, point multiplication, encryption, decryption, fuzzy extraction, PUF function, and random nonce generation. The execution time for T_{hash}, T_{pm}, T_{enc}, T_{dec}, T_{fuzzy}, T_{puf}, and T_{rg} are 0.23 ms, 2.226 ms, 3.85 ms, 3.85 ms, 2.68 ms, 12 ms, and 53.9 ms. Table 3 provides an overview of the comparison results.

Table 3. Computation costs at login and authentication phase.

Protocol	User	Gateway/Sever	Sensor Node	Total Cost
Ali et al. [25]	$10T_{hash} + 1T_{enc} + 1T_{dec} + 1T_{rg}$	$13T_{hash} + 2T_{enc} + 1T_{dec}$	$5T_{hash} + 1T_{dec}$	$28T_{hash} + T_{rg} + 3T_{enc} + 3T_{dec}(83.44ms)$
Li et al. [28]	$8T_{hash} + 3T_{pm} + 1T_{rg}$	$8T_{hash} + 1T_{pm} + 1T_{rg}$	$5T_{hash} + 1T_{pm} + 1T_{rg}$	$20T_{hash} + 6T_{pm} + 3T_{rg}(179.656ms)$
Masud et al. [23]	$3T_{hash} + 1T_{rg}$	$4T_{hash} + 2T_{rg}$	$2T_{hash} + 1T_{rg}$	$9T_{hash} + 4T_{rg}(217.67ms)$
Yuanbing et al. [9]	$14T_{hash} + 2T_{pm} + 2T_{rg}$	$10T_{hash}$	$6T_{hash} + 2T_{pm} + 1T_{rg}$	$30T_{hash} + 4T_{pm} + 3T_{rg}(177.504ms)$
Ours	$13T_{hash} + 1T_{rg} + 1T_{fuzzy}$	$15T_h + 1T_{rg}$	$6T_h + 1T_{puf} + 1T_{fuzzy}$	$34T_h + 2T_{rg} + 1T_{puf} + 2T_{fuzzy}(132.98ms)$

8.3. Communication Costs Comparison

In this secion, we compare the communication costs of our protocol and related protocols at the login and authentication phases. For comparison, assume that the hash value, entity ID, random nonce, and symmetric encryption value are 160 bits and the ECC value is 320 bits at the 160 bit security level of Fp, AES, and SHA1 [49]. We also assume that the timestamp value is 32 bits [50]. Based on these assumptions, the communication cost of our protocol is analyzed. Message $\{HID_i, PSID_j, M_1, M_2, M_3, T_1\}$, $\{M_4, M_5, M_6, T_2\}$, $\{M_7, T_3\}$, and $\{M_8, M_9, M_{10}\}$ have (160 + 160 + 160 + 160 + 160 + 32 = 832), (160 + 160 + 160 + 32 = 512), (160 + 32 = 192), and (160 + 160 + 160 = 480) bits are required. The total communication cost is $832 + 512 + 192 + 480 = 2016$ bits. Table 4 shows an analysis of communication costs of related protocols.

Table 4. Communication costs at login and authentication phase.

Protocol	Total Communication Costs
Ali et al. [25]	1952 bits
Li et al. [28]	2720 bits
Masud et al. [23]	2560 bits
Yuanbing et al. [9]	3552 bits
Ours	2016 bits

8.4. Results of Comparative Analysis

The results of the comparative analysis of the proposed protocol and other studies are as follows. Our proposed protocol has higher computation and communication costs than

Ali et al.'s protocol [25]. However, in terms of security, Ali et al.'s protocol is vulnerable to sensor capture and desynchronization attacks and does not guarantee perfect forward secrecy, but the proposed protocol is safe against various attacks and guarantees perfect forward secrecy. In addition, our protocol has a computation cost of 132.98 ms, which is lighter than other papers except for the protocol of Ali et al. Furthermore, the communication cost of our protocol is 2016 bits, which is higher than that of Ali et al., 1952 bits, but there is no big difference. In other papers, the communication cost of the proposed protocol is low. Moreover, from a security perspective, our proposed protocol is secure to replay, MITM, impersonation, smartcard stolen, and desynchronization attacks. PUF and three-factor can be used to provide security against ID/password pair guessing and sensor node capture attacks. Therefore, the proposed protocol can provide secure services to users in a WMSN environment and is a lightweight protocol that considers the resource limitations of sensor nodes.

9. Conclusions

With the development of WSN, patient status identification and medical diagnostic services using WMSNs, a type of WSN, have become common. However, since WMSN exchanges information through an open channel, it is vulnerable to attacks by attackers, and this vulnerability is an important security problem directly related to the patient's life. Therefore, in order to provide a secure WMSN service, an authentication protocol is required. In this study, we identify the problems of various authentication protocols using two-factor, three-factor, and PUF, and analyze the security vulnerabilities of Yuanbing et al.'s protocol in 2021. To address security vulnerabilities in these protocols, in this paper, we propose a secure authentication protocol applied with three-factor and PUF technology. To prove that the proposed protocol is secure against various attacks and provides security functions, formal verification and informal verification were performed through the ROR model, BAN logic, and AVISPA tool. In addition, through a comparative analysis of protocols, it was found that the calculation and communication costs were lower than those of the related protocols, and provide a more secure service in WMSN environments. Therefore, our proposed protocol can be secure against guessing, replay, MITM, impersonation, and sensor capture attacks and can provide anonymity, perfect forward secrecy, and secure mutual authentication. Our protocol also solves the problem of the sensor node, which has resource limitation, and ultimately can be applied to the actual WMSN environment. In the future, we plan to develop a better protocol by constructing and applying the proposed protocol to a practical testbed.

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

Funding: This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Education under grant 2020R1I1A3058605.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Sample Availability: Samples of the compounds are available from the authors.

Abbreviations

The following abbreviations are used in this manuscript:

Symbols	Meanings
U_i	i-th user (medical professional)
SN_j	j-th sensor node
GWN	Gateway node
PUF	Physical Unclonable Function
C_j, R_j	The challenge/response pair
ID_i, SID_j	Identity of U_i and SN_j
PW_i	Password of U_i
BIO_i	Biometrics of U_i
Gen, Rep	Fuzzy extractor's generation and reproduction algorithm
X_{GWN}	Secret key of GWN
RN_x, N_x	Random nonces
T_x	Timestamps
$HID_i, PSID_j$	Pseudo identity of U_i and SN_j
SK	Session key
$h(*)$	Collision resistant one-way hash function
$\oplus$	Bitwise exclusive-or operator

References

1. Rashid, B.; Rehmani, M.H. Applications of wireless sensor networks for urban areas: A survey. *J. Netw. Comput. Appl.* **2016**, *60*, 192–219. [CrossRef]
2. Pierce, F.J.; Elliott, T.V. Regional and on-farm wireless sensor networks for agricultural systems in Eastern Washington. *Comput. Electron. Agric.* **2008**, *61*, 32–43. [CrossRef]
3. Ryu, J.; Oh, J.; Kwon, D.; Son, S.; Lee, J.; Park, Y.; Park, Y. Secure ECC-based three-factor mutual authentication protocol for telecare medical information system. *IEEE Access* **2022**, *10*, 11511–11526. [CrossRef]
4. Bahache, A.N.; Chikouche, N.; Mezrag, F. Authentication Schemes for Healthcare Applications Using Wireless Medical Sensor Networks: A Survey. *SN Comput. Sci.* **2022**, *3*, 382. [CrossRef]
5. Zhang, L.; Zhang, Y.; Tang, S.; Luo, H. Privacy protection for e-health systems by means of dynamic authentication and three-factor key agreement. *IEEE Trans. Indust. Elec.* **2017**, *65*, 2795–2805. [CrossRef]
6. He, D.; Kumar, N.; Chen, J.; Lee, C.-C.; Chilamkurti, N.; Yeo, S.-S. Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. *Multimed. Syst.* **2015**, *21*, 49–60. [CrossRef]
7. Wu, F.; Xu, L.; Kumari, S.; Li, X. An improved and anonymous two factor authentication protocol for health-care applications with wireless medical sensor networks. *Multimed. Syst.* **2017**, *23*, 195–205. [CrossRef]
8. Wang, C.; Xu, G.; Li, W. A secure and anonymous two-factor authentication protocol in multiserver environment. *Secur. Commun. Netw.* **2018**, *2018*, 1–15. [CrossRef]
9. Yuanbing, W.; Wanrong, L.; Bin, L. An Improved Authentication Protocol for Smart Healthcare System Using Wireless Medical Sensor Network. *IEEE Access* **2021**, *9*, 105101–105117. [CrossRef]
10. Maes, R. Physically unclonable functions: Properties. In *Physically Unclonable Functions*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 49–80.
11. Abdalla, M.; Fouque, P. -A.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. In *Lecture Notes in Computer Science, Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05), Les Diablerets, Switzerland, 23–26 January 2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 65–84.
12. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [CrossRef]
13. AVISPA. Automated Validation of Internet Security Protocols and Applications. Available online: http://www.avispa-project.org/ (accessed on 21 September 2022).
14. Lamport, L. Password authentication with insecure communication. *Commun. ACM* **1981**, *24*, 770–772. [CrossRef]
15. Kumar, P.; Lee, S.-G.; Lee, H.-J. E-SAP: Efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks. *Sensors* **2012**, *12*, 1625–1647. [CrossRef]
16. Li, X.; Niu, J.; Kumari, S.; Liao, J.; Liang, W.; Khan, M.K. A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity. *Secur. Commun. Netw.* **2016**, *9*, 2643–2655. [CrossRef]
17. Das, A.K.; Sutrala, A.K.; Odelu, V.; Goswami, A. A secure smartcard-based anonymous user authentication scheme for healthcare applications using wireless medical sensor networks. *Wirel. Pers. Commun.* **2017**, *94*, 1899–1933. [CrossRef]
18. Amin, R.; Islam, S.H.; Biswas, G.P.; Khan, M.K.; Kumar, N. A robust and anonymous patient monitoring system using wireless medical sensor networks. *Future Gener. Comput. Syst.* **2018**, *80*, 483–495. [CrossRef]

19. Jiang, Q.; Ma, J.; Yang, C.; Ma, X.; Shen, J.; Chaudhry, S.A. Efficient end-to-end authentication protocol for wearable health monitoring systems. *Comput. Electr. Eng.* **2017**, *63*, 182–195. [CrossRef]
20. Jan, S.U.; Ali, S.; Abbasi, I.A.; Mosleh, M.A.; Alsanad, A.; Khattak, H. Secure patient authentication framework in the healthcare system using wireless medical sensor networks. *J. Healthc. Engin.* **2021**, *2021*, 9954089. [CrossRef]
21. Fotouhi, M.; Bayat, M.; Das, A.K.; Far, H.A.N.; Pournaghi, S.M.; Doostari, M.A. A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT. *Comput. Netw.* **2020**, *177*, 107333. [CrossRef]
22. Nashwan, S. An end-to-end authentication scheme for healthcare IoT systems using WMSN. *Comput. Mater. Contin.* **2018**, *68*, 607–642. [CrossRef]
23. Masud, M.; Gaba, G.S.; Choudhary, K.; Hossain, M.S.; Alhamid, M.F.; Muhammad, G. Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare. *IEEE Internet Things J.* **2021**, *9*, 2649–2656. [CrossRef]
24. Kwon, D.; Park, Y.; Park, Y. Provably Secure Three-Factor-Based Mutual Authentication Scheme with PUF for Wireless Medical Sensor Networks. *Sensors* **2021**, *21*, 6039. [CrossRef]
25. Ali, R.; Pal, A.K.; Kumari, S.; Sangaiah, A.K.; Li, X.; Wu, F. An enhanced three factor based authentication protocol using wireless medical sensor networks for healthcare monitoring. *J. Ambient. Intell. Humani. Comput.* **2018**, 1–22. [CrossRef]
26. Shuai, M.; Liu, B.; Yu, N.; Xiong, L. Lightweight and secure three-factor authentication scheme for remote patient monitoring using on-body wireless networks. *Secur. Commun. Netw.* **2019**, *2019*, 8145087. [CrossRef]
27. Mo, J.; Hu, Z.; Lin, Y. Cryptanalysis and security improvement of two authentication schemes for healthcare systems using wireless medical sensor networks. *Secur. Commun. Netw.* **2020**, *2020*, 5047379. [CrossRef]
28. Li, X.; Peng, J.; Obaidat, M.S.; Wu, F.; Khan, M.K.; Chen, C. A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems. *IEEE Syst. J.* **2019**, *14*, 39–50. [CrossRef]
29. Saleem, M.A.; Shamshad, S.; Ahmed, S.; Ghaffar, Z.; Mahmood, K. Security analysis on "A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems". *IEEE Syst. J.* **2021**, *15*, 5557–5559. [CrossRef]
30. Gope, P.; Sikdar, B. Lightweight and privacy-preserving two-factor authentication scheme for IoT devices. *IEEE Internet Things J.* **2018**, *6*, 580–589. [CrossRef]
31. Chen, C.M.; Li, X.; Liu, S.; Wu, M.E.; Kumari, S. Enhanced authentication protocol for the Internet of Things environment. *Secur. Commu. Netw.* **2022**, *2022*, 8543894. [CrossRef]
32. Aman, M.N.; Chua, K.C.; Sikdar, B. Mutual authentication in IoT systems using physical unclonable functions. *IEEE Internet Things J.* **2017**, *4*, 1327–1340. [CrossRef]
33. Frikken, K.B.; Blantonm, M.; Atallahm, M.J. Robust authentication using physically unclonable functions. In *International Conference on Information Security*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 262–277.
34. Chatterjee, U.; Chakraborty, R.S.; Mukhopadhyay, D. A PUF-based secure communication protocol for IoT. *ACM Trans. Embedded Comput. Syst.* **2017**, *16*, 1–25. [CrossRef]
35. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Lecture Notes in Computer Science, Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 523–540.
36. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
37. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In *Advances in Cryptology*; Springer Science and Business Media: Berlin, Germany; New York, NY, USA, 1999; pp. 388–397.
38. Messerges, T.S.; Dabbish, E.A.; Sloan, R.H. Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Comput.* **2002**, *51*, 541–552. [CrossRef]
39. Lee, J.; Kim, G.; Das, A.K.; Park, Y. Secure and efficient honey list-based authentication protocol for vehicular ad hoc networks. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 2412–2425. [CrossRef]
40. Son, S.; Lee, J.; Park, Y.; Park, Y.; Das, A.K. Design of blockchain-based lightweight V2I handover authentication protocol for VANET. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 1346–1358. [CrossRef]
41. Canetti, R.; Krawczyk, H. Universally composable notions of key exchange and secure channels. In *International Conference on the Theory and Applications of Cryptographic Thechniques (EUROCRYPT'02)*; Springer: Amsterdam, The Netherlands, 2002; pp. 337–351.
42. Li, J.; Su, Z.; Guo, D.; Choo, K.K.R.; Ji, Y. PSL-MAAKA: Provably secure and lightweight mutual authentication and key agreement protocol for fully public channels in internet of medical things. *IEEE Internet Things J.* **2021**, *8*, 13183–13195. [CrossRef]
43. Park, K.; Lee, J.; Das, A.K.; Park, Y. BPPS: Blockchain-Enabled Privacy-Preserving Scheme for Demand-Response Management in Smart Grid Environments. *IEEE Trans. Depend. Secur. Comput.* **2022**. [CrossRef]
44. Kim, M.; Lee, J.; Oh, J.; Park, K.; Park, Y.; Park, K. Blockchain based energy trading scheme for vehicle-to-vehicle using decentralized identifiers. *Appl. Energy* **2022**, *322*, 119445. [CrossRef]
45. Yu, S.; Das, A.K.; Park, Y.; Lorenz, P. SLAP-IoD: Secure and Lightweight Authentication Protocol Using Physical Unclonable Functions for Internet of Drones in Smart City Environments. *IEEE Trans. Veh. Technol.* **2022**, *71*, 10374–10388. [CrossRef]
46. Cho, Y.; Oh, J.; Kwon, D.; Son, S.; Yu, S.; Park, Y.; Park, Y. A Secure Three-Factor Authentication Protocol for E-Governance System Based on Multiserver Environments. *IEEE Access* **2022**, *10*, 74351–74365. [CrossRef]
47. Oh, J.; Lee, J.; Kim, M.; Park, Y.; Park, K.; Noh, S. A Secure Data Sharing Based on Key Aggregate Searchable Encryption in Fog-Enabled IoT Environment. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 4468–4481. [CrossRef]

48. Kilinc, H.H.; Yanik, T. A survey of SIP authentication and key agreement schemes. *IEEE Commun. Surv. Tutor.* **2013**, *16*, 1005–1023. [CrossRef]
49. Wu, F.; Xu, L.; Kumari, S.; Li, X. A new and secure authentication scheme for wireless sensor networks with formal proof. *Peer-to-Peer Netw. Appl.* **2017**, *10*, 16–30. [CrossRef]
50. He, D.; Zeadally, S.; Xu, B.; Huang, X. An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *IEEE Trans. Inf. For. Secur.* **2015**, *10*, 2681–2691. [CrossRef]