sensors

MDPI

Article

Provably Secure Three-Factor-Based Mutual Authentication Scheme with PUF for Wireless Medical Sensor Networks

DeokKyu Kwon [1], YoHan Park [2] and YoungHo Park [1,3,*]

[1] School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Korea; kdk145@knu.ac.kr
[2] School of Computer Engineering, Keimyung University, Daegu 42601, Korea; yhpark@kmu.ac.kr
[3] School of Electronics Engineering, Kyungpook National University, Daegu 41566, Korea
* Correspondence: parkyh@knu.ac.kr; Tel.: +82-53-950-7842

Abstract: Wireless medical sensor networks (WMSNs) are used in remote medical service environments to provide patients with convenient healthcare services. In a WMSN environment, patients wear a device that collects their health information and transmits the information via a gateway. Then, doctors make a diagnosis regarding the patient, utilizing the health information. However, this information can be vulnerable to various security attacks because the information is exchanged via an insecure channel. Therefore, a secure authentication scheme is necessary for WMSNs. In 2021, Masud et al. proposed a lightweight and anonymity-preserving user authentication scheme for healthcare environments. We discover that Masud et al.'s scheme is insecure against offline password guessing, user impersonation, and privileged insider attacks. Furthermore, we find that Masud et al.'s scheme cannot ensure user anonymity. To address the security vulnerabilities of Masud et al.'s scheme, we propose a three-factor-based mutual authentication scheme with a physical unclonable function (PUF). The proposed scheme is secure against various security attacks and provides anonymity, perfect forward secrecy, and mutual authentication utilizing biometrics and PUF. To prove the security features of our scheme, we analyze the scheme using informal analysis, Burrows–Abadi–Needham (BAN) logic, the Real-or-Random (RoR) model, and Automated Verification of Internet Security Protocols and Applications (AVISPA) simulation. Furthermore, we estimate our scheme's security features, computation costs, communication costs, and energy consumption compared with the other related schemes. Consequently, we demonstrate that our scheme is suitable for WMSNs.

Keywords: wireless medical sensor networks; PUF; biometrics; BAN logic; RoR model; AVISPA

Citation: 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. https://doi.org/10.3390/s21186039

Academic Editor: Simon Tjoa, Peter Kieseberg and Henri Ruotsalainen

Received: 17August 2021
Accepted: 7 September 2021
Published: 9 September 2021

1. Introduction

With the development of wireless communication and sensor minimization technology, wireless sensor networks (WSNs) have been widely used in various environments, such as industrial Internet of Things [1], healthcare [2], and smart homes [3]. In particular, the demand for remote healthcare services has been increased due to the COVID-19 pandemic [4]. Remote healthcare services can be realized through wireless medical sensor networks (WMSNs). Generally, WMSNs consist of doctors (users), a gateway, and sensor nodes. Doctors communicate with the gateway to access a patient's health data through their smart device. The gateway, such as a smart hospital, stores sensitive data and supports smooth wireless communication between doctors and sensor nodes. Sensor nodes are attached to patients and transmit patients' sensitive health data to doctors through the gateway [5]. Therefore, doctors can perform the diagnosis of patients remotely and patients can receive convenient remote medical services wherever they are.

Although WMSNs can provide convenient medical services to patients, there are several security risks. First of all, each message is exchanged through a public channel;

therefore, malicious adversaries can perform security attacks such as replay and man-in-the-middle attacks [6]. In addition, the smart device of a doctor can be stolen and an adversary can attempt to impersonate the doctor using parameters extracted from the device. In addition, the sensor node can be physically captured by an adversary and the adversary can attempt to impersonate the patient using the secret parameter, extracted from the sensor node. If an adversary obtains and modifies the information of patients using the above security attacks, this can have a serious effect on the patient's health, such as inducing a misdiagnosis by the doctor. Accordingly, secure authentication schemes are necessary to overcome these security vulnerabilities for WMSNs.

In 2021, Masud et al. [7] proposed a lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare environments. They claimed that their scheme is lightweight and prevents various security attacks (e.g., replay, privileged insider, and impersonation attacks). Moreover, they asserted that their scheme can ensure user anonymity and session key agreement. However, we find that Masud et al.'s scheme cannot prevent offline password guessing, user impersonation, and privileged insider attacks. Moreover, we prove that their scheme cannot ensure user anonymity. Their scheme also has a device update problem, where the doctor cannot perform a login process on his own smart device. To overcome these security vulnerabilities of Masud et al.'s scheme, we propose a secure three-factor-based mutual authentication scheme with physical unclonable function (PUF) for WMSNs. In our scheme, we use PUF and fuzzy extractor [8] to enhance the security level. The PUF is a physical circuit that outputs unpredictable random strings, and the fuzzy extractor is a cryptographic algorithm that utilizes the biometrics of users. Therefore, we install the PUF in the sensor node to prevent physical and cloning attacks, and we utilize the fuzzy extractor to overcome offline password guessing attacks. Our scheme also uses hash functions and exclusive-OR operations to ensure real-time communication.

1.1. Research Contributions

The contributions of our paper are as follows.

- We review Masud et al.'s scheme and prove that their scheme cannot ensure user anonymity. Moreover, we show that their scheme is vulnerable to offline password, impersonation, and privileged insider attacks and has a device update problem.
- We propose a secure three-factor-based mutual authentication scheme to overcome the security vulnerabilities of Masud et al.'s scheme. We use hash functions and exclusive-OR operations to provide real-time communication for WMSNs. We also utilize PUF and fuzzy extractor [8] to prevent physical and offline password guessing attacks, respectively.
- We analyze the security features of the proposed scheme using well-known Burrows–Abadi–Needham (BAN) logic [9] and the Real-or-Random (RoR) model [10], which can prove mutual authentication and session key security, respectively. Furthermore, we utilize the Automated Verification of Internet Security Protocols and Applications (AVISPA) simulation tool [11,12] to prove that the proposed scheme has resistance against replay and man-in-the-middle attacks.
- We show that our scheme has resistance against various security attacks, such as offline password, impersonation, privileged insider, replay, and man-in-the-middle attacks, using informal analysis. Moreover, the proposed scheme ensures user anonymity, perfect forward secrecy, and mutual authentication.
- We estimate the security properties and functionalities, communication costs, computation costs, and energy consumption of our scheme in comparison with existing authentication schemes.

1.2. Organization

In Section 2, we introduce related works for WMSNs. We describe the PUF, fuzzy extractor, adversary model, and system model in Section 3. In Section 4, we describe the detailed procedures of Masud et al.'s scheme. In Section 5, we prove the security vulnerabilities of Masud et al.'s scheme. To overcome these security vulnerabilities, we

propose a secure three-factor-based mutual authentication scheme with PUF for WMSNs in Section 6. In Sections 7 and 8, we analyze the security features of our scheme using formal and informal analyses and estimate the performance of our scheme, respectively. Finally, we conclude and summarize our paper in Section 9.

2. Related Works

In the past several decades, researchers have proposed numerous two-factor-based authentication schemes for WMSNs. In 2012, Kumar et al. [13] proposed an authentication scheme for healthcare applications using a smart card. Their scheme used a symmetric encryption method to establish the session key between the user and the medical sensor node. However, He et al. [14] claimed that Kumar et al.'s scheme is vulnerable to password guessing and privileged insider attacks. As a result, He et al. proposed a robust authentication scheme to overcome these security weaknesses. Unfortunately, Mir et al. [15] demonstrated that [14] cannot prevent offline password guessing and masquerading user attacks. To address the security vulnerabilities of He et al's scheme [15], they proposed an authentication and key agreement scheme using hash functions and exclusive-OR operations. In 2018, Wu et al. [16] proposed an authentication scheme for personalized healthcare systems. They used a smart device as a factor to protect the privacy of the doctor. However, the above schemes [13–16] can be vulnerable to smart device theft and offline password guessing attacks because they adopt two-factor-based authentication schemes.

Three-factor-based authentication schemes have been proposed to improve the security level for WMSNs. In 2018, Challa et al. [17] proposed a three-factor-based user authentication and key agreement protocol using bilinear pairings for wireless healthcare sensor networks. Challa et al. employed bilinear pairing and the fuzzy extractor to overcome security vulnerabilities such as smart card theft, offline password guessing, and privileged insider attacks. In 2019, Li et al. [18] proposed a three-factor user authentication protocol based on elliptic curve cryptography (ECC). They claimed that their scheme can resist various security attacks utilizing biometrics verification with error-correcting code and a fuzzy commitment scheme. Shin et al. [19] suggested an authentication and key agreement scheme that can preserve users' privacy in 5G-integrated IoT environments. In [19], each entity establishes the session key using elliptic curve Diffie–Hellman (ECDH). Furthermore, Ali et al. [20] proposed a biometric-based authentication and access control protocol for WMSNs using ECC. They claimed that their scheme is secure against privileged insider, stolen smart card, and offline password guessing attacks. In 2020, Hsu et al. [21] proposed a three-factor user-controlled single sign-on (UCSSO) scheme for telecare medicine information systems. Their scheme can provide fast authentication and privacy protection using only hash functions and exclusive-OR operations. Although the above schemes [18–21] can provide lightweight communications to doctors and patients, they cannot prevent sensor node physical and cloning attacks.

Recently, PUF-based authentication schemes have been proposed to prevent physical attacks. In 2017, Aman et al. [22] suggested a mutual authentication scheme using PUF in IoT systems. They claimed that their scheme is secure against IoT device cloning attacks because PUF is employed on each IoT device. Byun [23] proposed an end-to-end key exchange scheme using PUF. This scheme utilized PUF-embedded devices and the fuzzy extractor to ensure mutual authentication between two devices. In 2020, Fang et al. [24] proposed a PUF-based data transmission scheme for IoT environments. They proved that their scheme can prevent various attacks, such as DoS, eavesdropping, impersonation, and cloning attacks, using PUF. In 2021, Chen et al. [25] suggested an efficient mutual authentication and key agreement scheme using PUF and biometrics for wireless sensor network environments. To reduce the storage overhead of the user, Chen et al. [25] eliminated the password during the login phase.

In 2021, Masud et al. [7] proposed a lightweight user authentication scheme for IoT-based healthcare. They asserted that their scheme can protect against impersonation attacks and replay attacks and provide data privacy and anonymity. However, we discover that

their scheme is vulnerable to several security issues, such as offline password guessing, user impersonation, and privileged insider attacks. We also find that their scheme cannot ensure user anonymity. Therefore, we propose a three-factor-based mutual authentication scheme using PUF to prevent various security weaknesses such as user anonymity, smart device theft, offline password, privileged insider, and cloning attacks, which are critical for WMSNs.

3. Preliminaries

In this section, we introduce the general system model and the adversary model for WMSNs. Then, we describe PUF and the fuzzy extractor, which can improve the security level of our scheme.

3.1. System Model

Figure 1 shows the general system model of a WMSN, which consists of doctors, a gateway, and sensor nodes. Details are as follows.

- Doctor(user): The doctor, who has a resource-constrained smart device, authenticates with the gateway to access patients' health reports. To communicate with sensor nodes, the doctor must register with the gateway.
- Gateway: The gateway, which is the smart hospital, communicates with doctors and sensor nodes to provide efficient and convenient remote services to patients. We assume that the gateway is a trusted party and has enough storage and computing power.
- Sensor node: The sensor node is a resource-constrained device attached to the patient in the form of a wearable device. The sensor node collects the patient's health information and sends it to the doctor through the gateway.

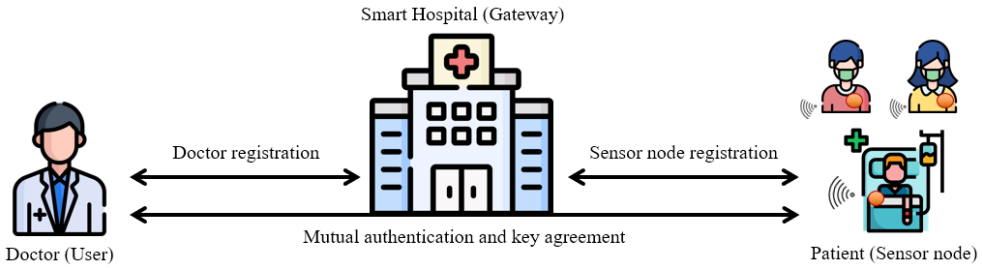

Figure 1. The general system model of WMSNs.

3.2. Adversary Model

In our paper, we assume that an adversary can eavesdrop, insert, remove, and modify messages transmitted through a public channel according to a well-known adversary model, the Dolev–Yao (DY) model [26]. Moreover, we use the Canetti–Krawczyk (CK) adversary model [27]. In this model, an adversary can access ephemeral parameters or the master key of the gateway. With the CK and DY adversary models, we assume that an adversary can perform various attacks. Details are as below.

- An adversary can steal a doctor's smart device and obtain the secret parameter, extracted from the smart device using a power analysis attack [28].
- An adversary can be a privileged insider who can obtain the user's registration message.
- An adversary can obtain the patient's sensor node and perform a cloning attack.
- An adversary can perform various attacks, such as man-in-the-middle, password guessing, and stolen verifier attacks [29].

3.3. Physical Unclonable Function

Physical unclonable functions (PUFs) are physical circuits that operate as a one-way function. In the PUF circuit, there is an input–output bit string pair called the "challenge–

response pair". If a random bit string challenge is entered into the PUF circuit, a unique output response is printed out. In this paper, we express this process as $R = PUF(C)$, where C and R are a challenge and a response, respectively. Ideal PUF properties are as below.

- The PUF is an unclonable circuit.
- The PUF is a unique physical microstructure. The output of the PUF depends on the physical circuit.
- The output of the PUF has to be unpredictable.
- The circuit of the PUF is easy to estimate and implement.

Since a PUF has the properties of a one-way function, the PUF returns the same response when the same challenge is input into a PUF-installed device. Moreover, the PUF gives different responses when the same challenge is input into different devices. Therefore, the PUF can provide a unique one-way function that cannot be duplicated. This uniqueness enables the PUF to prevent various attacks, such as physical and cloning attacks.

3.4. Fuzzy Extractor

In this section, we explain the basic concept and direction of the fuzzy extractor [8]. When a user utilizes his biometrics or the PUF response string, we cannot ensure the accuracy due to the noise of external environmental factors. The fuzzy extractor can control the noise using the helper string. Therefore, we can use the biometric information and the PUF response string as a secret parameter using the fuzzy extractor. The fuzzy extractor consists of "generate ($Gen(.)$)" and "reproduce ($Rep(.)$)" algorithms. Details are as follows.

- $Gen(B_i) = (R_i, P_i)$: This is a probability algorithm to generate a secret string R_i. If a user inputs a random string B_i, the fuzzy extractor generates the secret parameter R_i and a helper string P_i.
- $Rep(B_i^*, P_i) = (R_i)$: This is a deterministic algorithm to reproduce the secret string R_i. If a user enters the random string B_i^*, the fuzzy extractor controls the noise of B_i^* using the helper string P_i and reproduces the secret string R_i.

4. Review of Masud et al.'s Scheme

In 2021, Masud et al. [7] proposed a lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare environments. Their scheme consists of user registration, sensor node registration, and mutual authentication and key agreement phases. Notations and descriptions are explained in Table 1.

Table 1. Notations and descriptions.

Notation	Description
D_{ID}, S_{ID}	Identity of the doctor and the sensor node
PW_D	Password of the doctor
BIO_D	Biometric template of the doctor
s	Master key of the gateway
R_{req}	Registration request message
R_{SG}, R_{SN}	Random number generated by the gateway and the sensor node
D_{TID}, S_{TID}	Temporary identity of the doctor and the sensor node
N_D, N_G, N_S	Random nonce generated by device of the doctor, the gateway, and the sensor node
CH_1, RE_1	Challenge and response pair
SK	Session key
$PUF(.)$	Physical unclonable function
$h(.)$	Hash function
$\|\|$	Concatenation operator
$\oplus$	Exclusive-OR operator

4.1. User Registration Phase

A doctor must register in the gateway to use this network system. We show the user registration phase of Masud et al.'s scheme as follows.

Step 1: The doctor inputs an identity D_{ID} and password PW_D, and generates a registration request message R_{req}. Then, the doctor sends $M^1_{RD} = \{D_{ID}, PW_D, R_{req}\}$ to the gateway through a secure channel.

Step 2: The gateway stores D_{ID} and PW_D, and then generates R^1_{SG} to compute $\alpha = (D_{ID} \oplus R^1_{SG}) \oplus PW_D$ and $D_{TID} = R^1_{SG} \oplus D_{ID}$. The gateway stores $\{R^1_{SG}, D_{TID}\}$ in its secure database and sends α to the doctor via a secure channel.

Step 3: The doctor computes $R^{1*}_{SG} = (\alpha \oplus PW_D) \oplus D_{ID}$ and $D_{TID} = R^{1*}_{SG} \oplus D_{ID}$, and stores $\{R^{1*}_{SG}, D_{TID}\}$ in his device. Then, the doctor computes $\beta = h(PW_D||R^{1*}_{SG}) \oplus D_{TID}$ and stores $\{\beta\}$.

4.2. Sensor Node Registration Phase

To transmit the health information of a patient, the sensor node must register with the gateway. We describe the sensor node registration phase as below.

Step 1: The sensor node generates R^1_{SN}, and sends $\{S_{ID}, R^1_{SN}\}$ to the gateway via a secure channel, where S_{ID} is the real identity of the sensor node.

Step 2: The gateway generates R^2_{SG} and computes $\delta = (S_{ID} \oplus R^2_{SG}) \oplus R^1_{SN}$ and $S_{TID} = R^2_{SG} \oplus S_{ID}$. Then, the gateway stores $\{S_{ID}, R^1_{SN}, R^2_{SG}, S_{TID}\}$ in its secure database and transmits $\{\delta\}$ to the sensor node through a secure channel.

Step 3: When the sensor node receives $\{\delta\}$, it computes $R^{2*}_{SG} = (\delta \oplus R^1_{SN}) \oplus S_{ID}$ and $S_{TID} = R^{2*}_{SG} \oplus S_{ID}$. Finally, the sensor node stores $\{R^1_{SN}, R^{2*}_{SG}, S_{TID}\}$ in its memory.

4.3. Mutual Authentication and Key Agreement Phase

In this phase, the doctor and the sensor node conduct a mutual authentication and key agreement phase to authenticate each other and establish a session key. Figure 2 shows the mutual authentication and key agreement phase of Masud et al.'s scheme and details are as follows.

Step 1: When the doctor inputs his own password PW_D, the smart device of the doctor computes $Q = h(PW_D||R^{1*}_{SG})$ and verifies $Q \stackrel{?}{=} \beta$. If it is correct, the smart device generates a random nonce N^1_D and computes $N^{1*}_D = N^1_D \oplus PW_D$ and $\lambda = h(R^{1*}_{SG}||PW_D)$. Then, the doctor sends $\{N^{1*}_D, D_{TID}, \lambda, S_{TID}\}$ to the gateway via a public channel.

Step 2: The gateway receives $\{N^{1*}_D, D_{TID}, \lambda, S_{TID}\}$ and computes $N^1_D = N^{1*}_D \oplus PW_D$. If N^1_D is a fresh random nonce, the gateway checks the validity of S_{TID} and D_{TID}, and computes $\lambda^* = h(R^1_{SG}||PW_D)$. After verifying the equation $\lambda^* \stackrel{?}{=} \lambda$, the gateway generates N^1_G and computes $G^1_W = N^1_G \oplus S_{TID}$, $G^2_W = h(R^1_{SN}||R^2_{SG})$, $SK_S = (SK \oplus R^1_{SN}) \oplus N^1_G$, and $G^3_W = R^3_{SG} \oplus R^1_{SN}$, where SK is a session key. Then, the gateway sends $\{G^1_W, G^2_W, D_{TID}, SK_S, G^3_W\}$ to the sensor node through a public channel.

Step 3: The sensor node computes $N^1_G = G^1_W \oplus S_{TID}$ and checks the freshness of N^1_G. After this, the sensor node computes $S^1_N = h(R^1_{SN}||R^2_{SG})$ and checks the equality of S^1_N and G^2_W. If it is equal, the sensor node generates N^1_S and computes $SK = (SK_S \oplus N^1_G) \oplus R^1_{SG}$, $S^2_N = N^1_S \oplus S_{TID}$, $S^3_N = h(R^{2*}_{SG}||R^1_{SN}||SK)$, $S^4_N = R^2_{SG} \oplus R^2_{SN}$, $R^3_{SG} = G^2_W \oplus R^1_{SN}$, and $S^{new}_{TID} = R^3_{SG} \oplus S_{ID}$. Finally, the sensor node stores $\{R^2_{SN}, R^3_{SG}, S^{new}_{TID}\}$ and transmits $\{S^2_N, S^3_N, S^4_N\}$ to the gateway.

Step 4: When the gateway receives $\{S^2_N, S^3_N, S^4_N\}$ from the sensor node, the gateway computes $N^1_S = S^2_N \oplus S_{TID}$ and verifies the freshness of N^1_S. Then, the gateway computes $G^4_W = h(R^2_{SG}||R^1_{SN}||SK)$ and checks $G^4_W \stackrel{?}{=} S^3_N$. If it is equal, the gateway computes $R^2_{SN} = S^4_N \oplus R^2_{SG}$ and $S^{new}_{TID} = R^3_{SG} \oplus S_{ID}$ and stores $\{R^2_{SN}, R^3_{SG}, S^{new}_{TID}\}$ in its database. The gateway generates a random nonce N^2_G and computes $\mu = D_{ID} \oplus N^2_G$, $SK_U = (SK \oplus PW_D) \oplus N^2_G$, $\eta = h(D_{ID}||PW_D||SK||N^2_G)$, $G^5_W = R^4_{SG} \oplus PW_D$, and $D^{new}_{TID} = R^4_{SG} \oplus D_{ID}$. Lastly, the gateway stores $\{R^4_{SG}, D^{new}_{TID}\}$ in its secure database and sends a message $\{\mu, SK_U, \eta, G^5_W\}$ to the smart device of the doctor.

Step 5: After receiving $\{\mu, SK_U, \eta, G_W^5\}$ from the gateway, the doctor computes $N_G^2 = \mu \oplus D_{ID}$ and checks the freshness of N_G^2. Then, the smart device computes the session key $SK = (SK_U \oplus N_G^2) \oplus PW_D$ and $\phi = h(D_{ID}||PW_D||SK||N_G^2)$, and verifies $\phi \stackrel{?}{=} \eta$. If it is equal, the smart device computes $R_{SG}^4 = G_W^5 \oplus PW_D$ and $D_{TID}^{new} = R_{SG}^4 \oplus D_{ID}$, and stores $\{R_{SG}^4, D_{TID}^{new}\}$ in its memory.

Doctor	Gateway	Sensor node
Inputs PW_D		
Computes $Q = h(PW_D\|\|R_{SG}^{1*})$		
Checks $Q \stackrel{?}{=} \beta$		
Generates a random nonce N_D^1		
Computes $N_D^{1*} = N_D^1 \oplus PW_D$		
$\lambda = h(R_{SG}^{1*}\|\|PW_D)$		
$\xrightarrow{\{N_D^{1*}, D_{TID}, \lambda, S_{TID}\}}$		
	Computes $N_D^1 = N_D^{1*} \oplus PW_D$	
	Verifies the freshness of N_D^1	
	Checks the validity of S_{TID}, D_{TID}	
	Computes $\lambda^* = h(R_{SG}^1\|\|PW_D)$	
	Checks $\lambda^* \stackrel{?}{=} \lambda$	
	Generates N_G^1	
	Computes $G_W^1 = N_G^1 \oplus S_{TID}$	
	$G_W^2 = h(R_{SN}^1\|\|R_{SG}^2)$	
	$SK_S = (SK \oplus R_{SN}^1) \oplus N_G^1$	
	$G_W^3 = R_{SG}^3 \oplus R_{SN}^1$	Computes $N_G^1 = G_W^1 \oplus S_{TID}$
	$\xrightarrow{\{G_W^1, G_W^2, D_{TID}, SK_S, G_W^3\}}$	Verifies the freshness of N_G^1
		Computes $S_N^1 = h(R_{SN}^1\|\|R_{SG}^2)$
		Checks $S_N^1 \stackrel{?}{=} G_W^2$
		Generates N_S^1
		Computes $SK = (SK_S \oplus N_G^1) \oplus R_{SG}^1$
		$S_N^2 = N_S^1 \oplus S_{TID}$
		$S_N^3 = h(R_{SG}^{2*}\|\|R_{SN}^1\|\|SK)$
		$S_N^4 = R_{SG}^2 \oplus R_{SN}^2$
		$R_{SG}^3 = G_W^2 \oplus R_{SN}^1$
		$S_{TID}^{new} = R_{SG}^3 \oplus S_{ID}$
	Computes $N_S^1 = S_N^2 \oplus S_{TID}$	Stores $\{R_{SN}^2, R_{SG}^3, S_{TID}^{new}\}$
		$\xleftarrow{\{S_N^2, S_N^3, S_N^4\}}$
	Verifies the freshness of N_S^1	
	Computes $G_W^4 = h(R_{SG}^2\|\|R_{SN}^1\|\|SK)$	
	Checks $G_W^4 \stackrel{?}{=} S_N^3$	
	Computes $R_{SN}^2 = S_N^4 \oplus R_{SG}^2$	
	$S_{TID}^{new} = R^3SG \oplus S_{ID}$	
	Stores $\{R_{SN}^2, R_{SG}^3, S_{TID}^{new}\}$	
	Generates N_G^2	
	Computes $\mu = D_{ID} \oplus N_G^2$	
	$SK_U = (SK \oplus PW_D) \oplus N_G^2$	
	$\eta = h(D_{ID}\|\|PW_D\|\|SK\|\|N_G^2)$	
	$G_W^5 = R_{SG}^4 \oplus PW_D$	
	$D_{TID}^{new} = R_{SG}^4 \oplus D_{ID}$	
Computes $N_G^2 = \mu \oplus D_{ID}$	Stores $\{R_{SG}^4, D_{TID}^{new}\}$	
	$\xleftarrow{\{\mu, SK_U, \eta, G_W^5\}}$	
Verifies the freshness of N_G^2		
Computes $SK = (SK_U \oplus N_G^2) \oplus PW_D$		
$\phi = h(D_{ID}\|\|PW_D\|\|SK\|\|N_G^2)$		
Checks $\phi \stackrel{?}{=} \eta$		
Computes $R_{SG}^4 = G_W^5 \oplus PW_D$		
$D_{TID}^{new} = R_{SG}^4 \oplus D_{ID}$		
Stores $\{R_{SG}^4, D_{TID}^{new}\}$		

Figure 2. Mutual authentication and key agreement phase of Masud et al.'s scheme.

5. Cryptanalysis of Masud et al.'s Scheme

If an adversary $\mathcal{A}$ obtains a legitimate user's smart device, $\mathcal{A}$ can extract the information $\{\beta, R_{SG}^{1*}, D_{TID}\}$ from the smart device using a power analysis attack [28], according to Section 3.2. With this information, $\mathcal{A}$ can perform various security attacks, such as offline password guessing, user impersonation, and privileged insider attacks. Furthermore, Masud et al.'s scheme does not ensure user anonymity and has a device update problem when signing in for the next session. Details are shown as below.

5.1. User Anonymity

An adversary $\mathcal{A}$ obtains the smart device of a doctor and extracts $\{\beta, R_{SG}^{1*}, D_{TID}\}$ using power analysis attack. Then, $\mathcal{A}$ calculates $D_{ID} = D_{TID} \oplus R_{SG}^{1*}$, where D_{ID} is the real identity of the doctor. Therefore, Masud et al.'s scheme cannot ensure user anonymity.

5.2. Offline Password Guessing Attack

An offline password guessing attack has a purpose of obtaining the valid password for a user using a password dictionary in polynomial time. Thus, an adversary $\mathcal{A}$ needs some information about the user in order to check whether the guessed password is correct or not. In Masud et al.'s scheme, $\mathcal{A}$ can verify the correctness of the guessed password using the information extracted from the smart device of the doctor. We describe the procedures as follows.

Step 1: The adversary $\mathcal{A}$ inputs a guessed password $PW_{\mathcal{A}}$ and calculates $Q^* = h(PW_{\mathcal{A}} || R_{SG}^{1*}) \oplus D_{TID}$.

Step 2: $\mathcal{A}$ compares $Q^* \stackrel{?}{=} \beta$, where $\beta = h(PW_D || R_{SG}^{1*}) \oplus D_{TID}$ is a parameter extracted from the smart device of the doctor. If it is equal, it means that $\mathcal{A}$ has guessed the password PW_D correctly.

Thus, Masud et al.'s scheme is vulnerable to offline password guessing attacks.

5.3. User Impersonation Attack

The adversary $\mathcal{A}$ can obtain the real identity D_{ID} and the password PW_D of the doctor, according to Sections 5.1 and 5.2. Then, $\mathcal{A}$ can impersonate the doctor with this information. We describe the steps as follows.

Step 1: $\mathcal{A}$ generates a random nonce $N_{\mathcal{A}}^1$ and computes $N_{\mathcal{A}}^{1*} = N_{\mathcal{A}}^1 \oplus PW_D$ and $\lambda_{\mathcal{A}} = h(R_{SG}^{1*} || PW_D)$. Then, $\mathcal{A}$ sends $\{N_{\mathcal{A}}^{1*}, D_{TID}, \lambda_{\mathcal{A}}, S_{TID}\}$ to the gateway.

Step 2: After receiving $\{N_{\mathcal{A}}^{1*}, D_{TID}, \lambda_{\mathcal{A}}, S_{TID}\}$ from the adversary $\mathcal{A}$, the gateway retrieves $N_{\mathcal{A}}^1 = N_{\mathcal{A}}^{1*} \oplus PW_D$ and checks the freshness of $N_{\mathcal{A}}^1$. If it is found to be fresh, the gateway verifies D_{TID} and S_{TID} from its database. Then, the gateway computes $\lambda^* = h(R_{SG}^1 || PW_D)$ and compares $\lambda^* \stackrel{?}{=} \lambda_{\mathcal{A}}$. If the equation is correct, the gateway generates a random nonce N_G^1 and computes $G_W^1 = N_G^1 \oplus S_{TID}$, $G_W^2 = h(R_{SN}^1 || R_{SG}^2)$, $SK_S = (SK \oplus R_{SN}^1) \oplus N_G^1$ and $G_W^3 = R_{SG}^3 \oplus R_{SN}^1$. Finally, the gateway sends $\{G_W^1, G_W^2, D_{TID}, SK_S, G_W^3\}$ to the sensor node.

Step 3: The sensor node receives $\{G_W^1, G_W^2, D_{TID}, SK_S, G_W^3\}$ and retrieves $N_G^1 = G_W^1 \oplus S_{TID}$. If N_G^1 is a fresh random nonce, the sensor node computes $S_N^2 = h(R_{SN}^1 || R_{SG}^2)$ and compares $S_N^2 \stackrel{?}{=} G_W^2$. The sensor node generates a random nonce N_S^1 and computes $SK = (SK_S \oplus R_{SN}^1) \oplus N_G^1$, $S_N^2 = N_S^1 \oplus S_{TID}$, $S_N^3 = h(R_{SG}^{2*} || R_{SN}^1 || SK)$, $S_N^4 = R_{SG}^2 \oplus R_{SN}^2$, $R_S^3 G = G_W^3 \oplus R_{SN}^1$ and $S_{TID}^{new} = R^3)_{SG} \oplus S_{ID}$. The sensor node sends $\{S_N^2, S_N^3, S_N^4\}$ and stores $\{R_{SN}^2, R_{SG}^3, S_{TID}^{new}\}$.

Step 4: The gateway receives the message $\{S_N^2, S_N^3, S_N^4\}$ and retrieves $N_S^1 = S_N^2 \oplus S_{TID}$. If N_S^1 is a fresh random nonce, the gateway computes $G_W^4 = h(R_{SG}^{2*} || R_{SN}^1 || SK)$ and checks $G_W^4 \stackrel{?}{=} S_N^3$. The gateway computes $R_{SN}^2 = S_N^4 \oplus R_{SG}^2$ and $S_{TID}^{new} = R_{SG}^3 \oplus S_{ID}$, and stores $\{R_{SN}^2, R_{SG}^3, S_{TID}^{new}\}$. After this, the gateway generates a random nonce N_G^2 and computes $\mu = D_{ID} \oplus N_G^2$, $SK_U = (SK \oplus PW_D) \oplus N_G^2$, $\eta = h(D_{ID} || PW_D || SK || N_G^2)$, $G_W^5 = R_{SG}^4 \oplus PW_D$ and $D_{TID}^{new} = R_{SG}^4 \oplus D_{ID}$. Lastly, the gateway stores $\{R_{SG}^4, D_{TID}^{new}\}$ and sends $\{\mu, SK_U, \eta, G_W^5\}$ to $\mathcal{A}$.

Step 5: $\mathcal{A}$ computes $N_G^2 = \mu \oplus D_{ID}$ and verifies the freshness of N_G^2. Then, $\mathcal{A}$ computes $SK = (SK_U \oplus PW_D) \oplus N_G^2$ and $\phi = h(D_{ID} || PW_D || SK || N_G^2)$, and compares $\phi \stackrel{?}{=} \eta$. Finally, $\mathcal{A}$ computes $R_{SG}^4 = G_W^5 \oplus PW_D$ and $D_{TID}^{new} = R_{SG}^4 \oplus D_{ID}$, and stores these parameters $\{R_{SG}^4, D_{TID}^{new}\}$.

Therefore, Masud et al.'s scheme cannot prevent an impersonation attack.

5.4. Privileged Insider Attack

A privileged insider attack can be performed by an insider adversary $\mathcal{A}$ that has unquestioned authority within the system. Therefore, the privileged insider $\mathcal{A}$ can obtain various information about users, including registration request messages, and may attempt to calculate the session key or impersonate a legal user.

In Masud et al.'s scheme, a privileged insider adversary $\mathcal{A}$ can impersonate a legitimate doctor after obtaining a registration request message $\{D_{ID}, PW_D, R_{req}\}$ and the secret parameter $\{\beta, R_{SG}^{1*}, D_{TID}\}$ extracted from the smart device of the doctor. $\mathcal{A}$ generates a random nonce $N_{\mathcal{A}}^1$ and computes $N_{\mathcal{A}}^{1*} = N_{\mathcal{A}}^1 \oplus PW_D$ and $\lambda_{\mathcal{A}} = h(R_{SG}^{1*}||PW_D)$. Then, $\mathcal{A}$ sends a message $\{N_{\mathcal{A}}^{1*}, D_{TID}, \lambda_{\mathcal{A}}, S_{TID}\}$. The gateway and the sensor node authenticate each other and return a message $\{\mu, SK_U, \eta, G_W^5\}$ to $\mathcal{A}$. Lastly, $\mathcal{A}$ calculates $N_G^2 = \mu \oplus D_{ID}$ and the session key $SK = (SK_U \oplus N_G^2) \oplus PW_D$. Thus, Masud et al.'s scheme is insecure against privileged insider attacks.

5.5. Device Update Problem

The smart device replaces $\{R_{SG}^{1*}, D_{TID}\}$ with $\{R_{SG}^4, D_{TID}^{new}\}$ at the end of the authentication and key agreement phase. After this, the doctor may try to authenticate another sensor node that is attached to a patient in other session. However, the doctor cannot perform the login phase. If the doctor inputs a password PW_D, the smart device computes $Q = h(PW_D||R_{SG}^4) \oplus D_{TID}^{new}$ and verifies $Q \stackrel{?}{=} \beta$. Since $\beta = h(PW_D||R_{SG}^{1*}) \oplus D_{TID}$, the login phase is aborted. Therefore, Masud et al.'s scheme has a device update problem.

6. Proposed Scheme

Although Masud et al.'s scheme has efficiency for WMSNs, their scheme has several security vulnerabilities. To address these security weaknesses, we propose a secure three-factor-based mutual authentication and key agreement scheme using PUF. Our scheme consists of initialization, user registration, sensor node registration, mutual authentication and key agreement, and password change phases.

6.1. Initialization Phase

Before starting the registration phase, the gateway inserts an identity and a challenge into the sensor node. Figure 3 shows the initialization of our scheme and detailed steps are as follows.

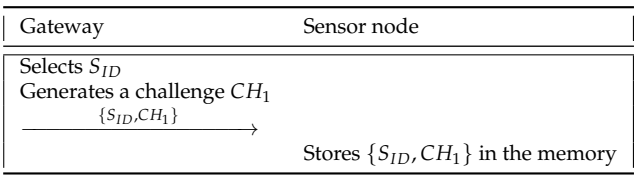

Figure 3. Initialization phase of the proposed scheme.

Step 1: The gateway selects an identity S_{ID}, a challenge CH_1, and sends $\{S_{ID}, CH_1\}$ to the sensor node via a secure channel.

Step 2: The sensor node stores $\{S_{ID}, CH_1\}$ in the memory.

6.2. User Registration Phase

A doctor must register in the network to provide a convenient remote medical service to patients. We show the sensor node registration phase in Figure 4 and detailed steps are as follows.

User (Doctor)	Gateway
Inputs D_{ID}, PW_D	
Imprints BID_D	
Generates R_{req}	
Computes $Gen(BIO_D) =< R_D, P_D >$	
$GPW_D = h(PW_D \| \| R_D)$	
$\xrightarrow{\{D_{ID}, GPW_D, R_{req}\}}$	
	Generates R^1_{SG}, R^2_{SG}
	Computes $\alpha = h(D_{ID} \| \| s)$
	$\beta = \alpha \oplus h(GPW_D \| \| R^2_{SG})$
	$\omega = \alpha \oplus h(R^1_{SG} \| \| s)$
	$D_{TID} = h(\alpha \| \| R^2_{SG} \| \| R^1_{SG})$
	Stores $\{\omega, R^1_{SG}, D_{TID}\}$ in the secure database
	$\xleftarrow{\{\beta, D_{TID}, R^2_{SG}\}}$
Computes $\theta = h(R_D \| \| GPW_D) \oplus R^2_{SG}$	
$Ver_D = h(D_{ID} \| \| GPW_D \| \| R_D)$	
Stores $\{\beta, \theta, Ver_D, D_{TID}, P_D\}$ in the memory	

Figure 4. User registration phase of the proposed scheme.

Step 1: A doctor inputs an identity D_{ID}, a password PW_D, and biometric template BIO_D to the smart device. Then, the smart device generates a registration request message R_{req} and computes $Gen(BID_D) =< R_D, P_D >$ and $GPW_D = h(PW_D||R_D)$, where $Gen(.)$ is a fuzzy extractor generation function. The doctor sends $\{D_{ID}, GPW_D, R_{req}\}$ to the gateway via a secure channel.

Step 2: After receiving $\{D_{ID}, GPW_D, R_{req}\}$ from the doctor, the gateway generates random numbers R^1_{SG} and R^2_{SG}, and computes $\alpha = h(D_{ID}||s)$, $\beta = \alpha \oplus h(GPW_D||R^2_{SG})$, $\omega = \alpha \oplus h(R^1_{SG}||s)$, and $D_{TID} = h(\alpha||R^2_{SG}||R^1_{SG})$. The gateway stores $\{\omega, R^1_{SG}, D_{TID}\}$ in the secure database, and sends $\{\beta, D_{TID}, R^2_{SG}\}$ to the doctor via a secure channel.

Step 3: The doctor computes $\theta = h(R_D||GPW_D) \oplus R^2_{SG}$ and $Ver_D = h(D_{ID}||GPW_D||R_D)$, and stores $\{\beta, \theta, Ver_D, D_{TID}, P_D\}$ in the memory.

6.3. Sensor Node Registration Phase

A patient must register in the network using a sensor node in order to receive remote medical services from the doctor. In Figure 5, we show the sensor node registration phase of our scheme and details are as below.

Sensor node	Gateway
Retrieves CH_1	
Computes $RE_1 = PUF(CH_1)$	
$Gen(RE_1) =< R^1_{SN}, P_{SN} >$	
$AS_{ID} = h(R^1_{SN} \| \| S_{ID})$	
$\xrightarrow{\{S_{ID}, AS_{ID}, CH_1\}}$	
	Generates R^3_{SG}
	Computes $\delta = h(AS_{ID} \| \| s)$
	$S_{TID} = h(\delta \| \| R^3_{SG} \| \| AS_{ID})$
	Stores $\{AS_{ID}, S_{TID}, CH_1\}$
	$\xleftarrow{\{\delta, S_{TID}\}}$
Deletes CH_1 from the memory	
Stores $\{\delta, P_{SN}, S_{TID}\}$	

Figure 5. Sensor node registration phase of the proposed scheme.

Step 1: The sensor node retrieves the challenge stored in the memory and computes $RE_1 = PUF(CH_1)$, $Gen(RE_1) =< R^1_{SN}, P_{SN} >$, and $AS_{ID} = h(R^1_{SN}||S_{ID})$. Then, the sensor node sends $\{S_{ID}, AS_{ID}, CH_1\}$ to the gateway through a secure channel.

Step 2: The gateway generates R^3_{SG} and computes $\delta = h(AS_{ID}||s)$, $S_{TID} = h(\delta||R^3_{SG}||AS_{ID})$. After this, the gateway stores $\{AS_{ID}, S_{TID}, CH_1\}$ in its secure database and sends $\{\delta, S_{TID}\}$ to the sensor node via a secure channel.

Step 3: Finally, the sensor node deletes the challenge CH_1 and stores $\{\delta, P_{SN}, S_{TID}\}$ in its memory.

6.4. Mutual Authentication and Key Agreement Phase

The doctor sends a login request message to the gateway and establishes a session key among the doctor, the gateway, and the sensor node. After this, the doctor can perform an accurate diagnosis of the patient. We describe the mutual authentication and key agreement phase in Figure 6 and details are as follows.

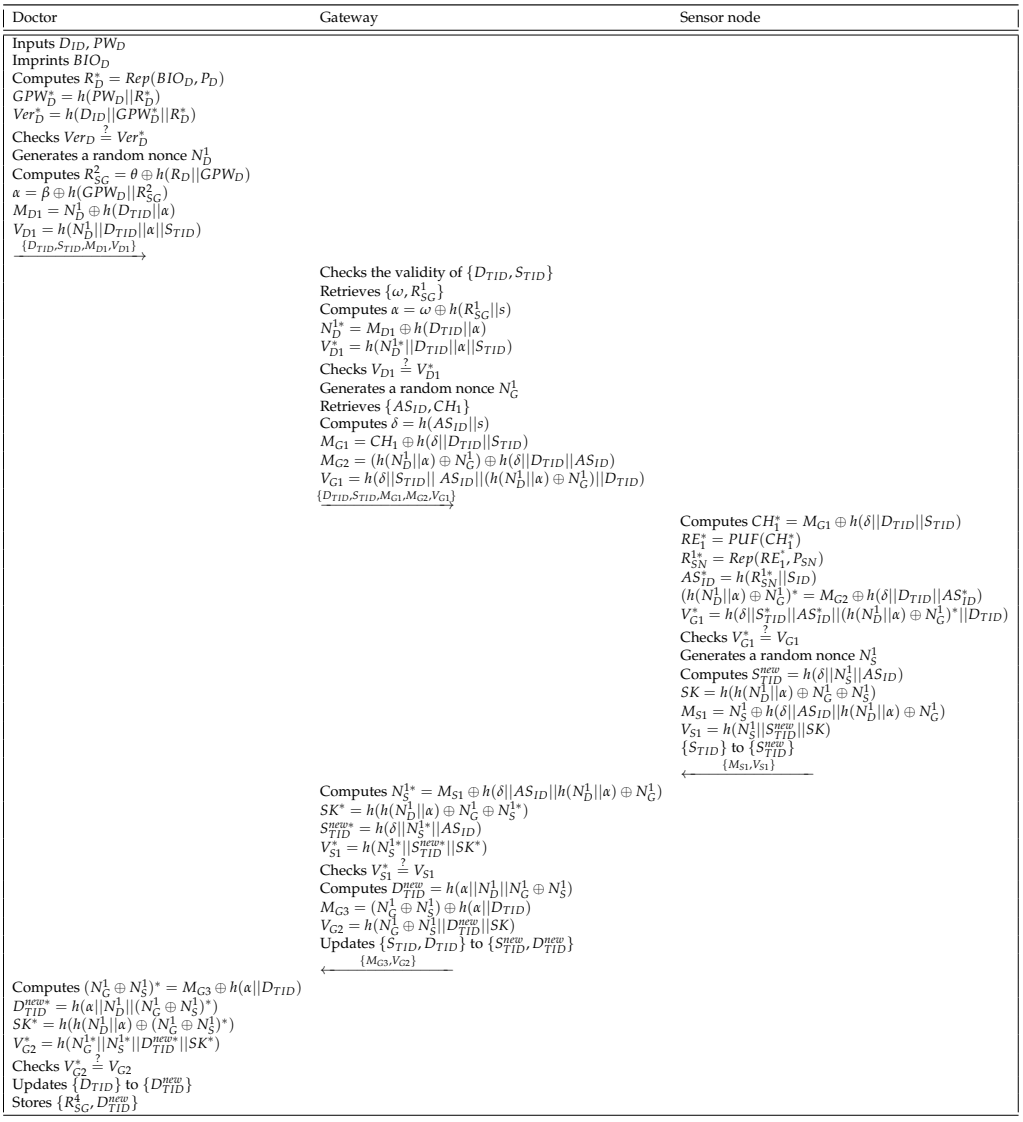

Figure 6. Mutual authentication and key agreement phase of the proposed scheme.

Step 1: The doctor inputs the identity D_{ID}, the password PW_D, and imprints the biometrics BIO_i. Then, the smart device computes $R_D^* = Rep(BIO_D, P_D)$, $GPW_D^* = h(PW_D||R_D^*)$, and $Ver_D^* = h(D_{ID}||GPW_D^*||R_D^*)$, and verifies $Ver_D \stackrel{?}{=} Ver_D^*$. If it is correct, the smart device generates a random nonce N_D^1 and computes $R_{SG}^2 = \theta \oplus h(R_D||GPW_D)$, $\alpha = \beta \oplus h(GPW_D||R_{SG}^2)$, $M_{D1} = N_D^1 \oplus h(D_{TID}||\alpha)$, and $V_{D1} = h(N_D^1||D_{TID}||\alpha||S_{TID})$. The smart device sends $\{D_{TID}, S_{TID}, M_{D1}, V_{D1}\}$ to the gateway through a public channel.

Step 2: When the gateway receives the message $\{D_{TID}, S_{TID}, M_{D1}, V_{D1}\}$ from the doctor, the gateway checks the pseudo identity $\{D_{TID}, S_{TID}\}$ and retrieves $\{\omega, R_{SG}^1\}$ in the database. Then, the gateway computes $\alpha = \omega \oplus h(R_{SG}^1||s)$, $N_D^{1*} = M_{D1} \oplus h(D_{TID}||\alpha)$, and $V_{D1}^* = h(N_D^{1*}||D_{TID}||\alpha||S_{TID})$. If $V_{D1} \stackrel{?}{=} V_{D1}^*$ is correct, the gateway generates a random nonce N_G^1 and retrieves $\{AS_{ID}, CH_1\}$. The gateway computes $\delta = h(AS_{ID}||s)$, $M_{G1} = CH_1 \oplus h(\delta||D_{TID}||S_{TID})$, $M_{G2} = (h(N_D^1||\alpha) \oplus N_G^1) \oplus h(\delta||D_{TID}||AS_{ID})$, and

$V_{G1} = h(\delta||S_{TID}||\ AS_{ID}||(h(N_D^1||\alpha) \oplus N_G^1)||D_{TID})$. After this, the gateway transmits $\{D_{TID}, S_{TID}, M_{G1}, M_{G2}, V_{G1}\}$ to the sensor node via a public channel.

Step 3: The sensor node computes $CH_1^* = M_{G1} \oplus h(\delta||D_{TID}||S_{TID})$, $RE_1^* = PUF(CH_1^*)$, $R_{SN}^{1*} = Rep(RE_1^*, P_{SN})$, $AS_{ID}^* = h(R_{SN}^{1*}||S_{ID})$, $(h(N_D^1||\alpha) \oplus N_G^1)^* = M_{G2} \oplus h(\delta\ ||D_{TID}\ ||AS_{ID}^*)$, and $V_{G1}^* = h(\delta||S_{TID}^*||AS_{ID}^*||(h(N_D^1||\alpha) \oplus N_G^1)^*||D_{TID})$. If the equation $V_{G1}^* \stackrel{?}{=} V_{G1}$ is correct, the sensor node generates a random nonce N_S^1 and computes a new pseudo identity $S_{TID}^{new} = h(\delta||N_S^1||AS_{ID})$, a session key $SK = h(h(N_D^1||\alpha) \oplus N_G^1 \oplus N_S^1)$, $M_{S1} = N_S^1 \oplus h(\delta||AS_{ID}||h(N_D^1||\alpha) \oplus N_G^1)$, and $V_{S1} = h(N_S^1||S_{TID}^{new}||SK)$. Lastly, the sensor node sends $\{M_{S1}, V_{S1}\}$ to the gateway through a public channel and updates $\{S_{TID}\}$ to $\{S_{TID}^{new}\}$.

Step 4: After receiving $\{M_{S1}, V_{S1}\}$ from the sensor node, the gateway computes $N_S^{1*} = M_{S1} \oplus h(\delta||AS_{ID}||h(N_D^1||\alpha) \oplus N_G^1)$, the session key $SK^* = h(h(N_D^1||\alpha) \oplus N_G^1 \oplus N_S^{1*})$, the new pseudo identity of the sensor node $S_{TID}^{new*} = h(\delta||N_S^{1*}||AS_{ID})$, and $V_{S1}^* = h(N_S^{1*}\ ||S_{TID}^{new*}\ ||SK^*)$. If the equation $V_{S1}^* \stackrel{?}{=} V_{S1}$ is correct, the gateway computes a new pseudo identity of the doctor $D_{TID}^{new} = h(\alpha||N_D^1||N_G^1 \oplus N_S^1)$, $M_{G3} = (N_G^1 \oplus N_S^1) \oplus h(\alpha||D_{TID})$, and $V_{G2} = h(N_G^1 \oplus N_S^1||D_{TID}^{new}||SK)$. Then, the gateway sends $\{M_{G3}, V_{G2}\}$ to the doctor and updates $\{S_{TID}, D_{TID}\}$ to $\{S_{TID}^{new}, D_{TID}^{new}\}$.

Step 5: The doctor computes $(N_G^1 \oplus N_S^1)^* = M_{G3} \oplus h(\alpha||D_{TID})$, $D_{TID}^{new*} = h(\alpha||\ N_D^1||\ (N_G^1 \oplus N_S^1)^*)$, $SK^* = h(h(N_D^1||\alpha) \oplus (N_G^1 \oplus N_S^1)^*)$, and $V_{G2}^* = h(N_G^{1*}||N_S^{1*}||D_{TID}^{new*}||\ SK^*)$ and verifies $V_{G2}^* \stackrel{?}{=} V_{G2}$. If it is correct, the doctor replaces $\{D_{TID}\}$ with $\{D_{TID}^{new}\}$ in the smart device.

6.5. Password Change Phase

In our scheme, we provide a convenient password update process for the doctor. Detailed steps are as follows.

Step 1: A doctor inputs D_{ID}, PW_D, and BIO_D to the smart device.

Step 2: The smart device computes $R_D^* = Rep(BIO_D, P_D)$, $GPW_D^* = h(PW_D||R_D^*)$, and $Ver_D^* = h(D_{ID}||GPW_D^*||R_D^*)$ and verifies $Ver_D \stackrel{?}{=} Ver_D^*$. If the equation is correct, the smart device demands a new password from the doctor.

Step 3: The doctor inputs a new password PW_D^{new} to the smart device.

Step 4: The smart device computes $GPW_D^{new} = h(PW_D^{new}||R_D)$, $\beta = \alpha \oplus h(GPW_D^{new}\ ||R_{SG}^2)$, $\theta = h(R_D||GPW_D^{new}) \oplus R_{SG}^2$, and $Ver_D^{new} = h(D_{ID}\ ||GP^{new}W_D\ ||R_D)$ and updates $\{\beta, \theta, Ver_D\}$ to $\{\beta^{new}, \theta^{new}, Ver_D^{new}\}$.

7. Security Analysis

To prove the security features of the proposed scheme, we use BAN logic and the RoR model, which can prove the mutual authentication properties and session key security, respectively. Moreover, we show that our scheme has resistance against man-in-the-middle and replay attacks using AVISPA. Furthermore, we claim that the proposed scheme can prevent various security attacks using informal analysis.

7.1. BAN Logic

BAN logic is a well-known formal proof to verify the mutual authentication of a protocol. Therefore, many researchers have used BAN logic to prove the mutual authentication of their schemes [30–33]. In this section, we prove the mutual authentication of the proposed scheme using BAN logic [9]. The basic notations and descriptions of BAN logic are shown in Table 2.

7.1.1. Rules

The logical rules of BAN logic are as follows.

Table 2. Notations of BAN logic.

Notation	Description
P_1, P_2	Principals
M_1, M_2	Statements
SK	Session key
$P_1 \mid \equiv M_1$	P_1 believes M_1
$P_1 \mid \sim M_1$	P_1 once said M_1
$P_1 \mapsto M_1$	P_1 controls M_1
$P_1 \lhd M_1$	P_1 receives M_1
$\#M_1$	M_1 is fresh
$\{M_1\}_K$	M_1 is encrypted with K
$P_1 \overset{K}{\leftrightarrow} P_2$	P_1 and P_2 have shared key K

1. Message meaning rule (MMR) :

$$\frac{P_1 \mid \equiv P_1 \overset{K}{\leftrightarrow} P_2, \quad P_1 \lhd \{M_1\}_K}{P_1 \mid \equiv P_2 \mid \sim M_1}$$

2. Nonce verification rule (NVR) :

$$\frac{P_1 \mid \equiv \#(M_1), \quad P_1 \mid \equiv P_2 \mid \sim M_1}{P_1 \mid \equiv P_2 \mid \equiv M_1}$$

3. Jurisdiction rule (JR) :

$$\frac{P_1 \mid \equiv P_2 \mapsto M_1, \quad P_1 \mid \equiv P_2 \mid \equiv M_1}{P_1 \mid \equiv M_1}$$

4. Belief rule (BR) :

$$\frac{P_1 \mid \equiv (M_1, M_2)}{P_1 \mid \equiv M_1}$$

5. Freshness rule (FR) :

$$\frac{P_1 \mid \equiv \#(M_1)}{P_1 \mid \equiv \#(M_1, M_2)}$$

7.1.2. Goals

The BAN logic goals of the proposed scheme are as follows. We define the principals DO, GWN, and SN as the doctor, the gateway, and the sensor node, respectively.

Goal 1: $DO \mid \equiv DO \overset{SK}{\leftrightarrow} GWN$

Goal 2: $DO \mid \equiv GWN \mid \equiv DO \overset{SK}{\leftrightarrow} GWN$

Goal 3: $GWN \mid \equiv DO \overset{SK}{\leftrightarrow} GWN$

Goal 4: $GWN \mid \equiv DO \mid \equiv DO \overset{SK}{\leftrightarrow} GWN$

Goal 5: $SN \mid \equiv SN \overset{SK}{\leftrightarrow} GWN$

Goal 6: $SN| \equiv GWN| \equiv SN \overset{SK}{\longleftrightarrow} GWN$

Goal 7: $GWN| \equiv SN \overset{SK}{\longleftrightarrow} GWN$

Goal 8: $GWN| \equiv SN| \equiv SN \overset{SK}{\longleftrightarrow} GWN$

7.1.3. Idealized Forms

In the proposed scheme, there are four messages exchanged through a public channel. We transform these messages into idealized forms. Our scheme's idealized forms for the messages are as follows:

$Message_1 : DO \rightarrow GWN : \{N_D^1\}_\alpha$

$Message_2 : GWN \rightarrow SN : \{N_G^1, h(N_D^1||\alpha)\}_\delta$

$Message_3 : SN \rightarrow GWN : \{N_S^1\}_\delta$

$Message_4 : GWN \rightarrow DO : \{N_G^1, N_S^1\}_\alpha$

7.1.4. Assumptions

The assumptions in the proposed scheme are shown below.

$A_1 : GWN| \equiv \#(N_D^1)$

$A_2 : GWN| \equiv \#(N_S^1)$

$A_3 : SN| \equiv \#(h(N_D^1||\alpha))$

$A_4 : DO| \equiv \#(N_G^1)$

$A_5 : DO| \equiv GWN \Rightarrow (DO \overset{SK}{\longleftrightarrow} GWN)$

$A_6 : GWN| \equiv DO \Rightarrow (DO \overset{SK}{\longleftrightarrow} GWN)$

$A_7 : SN| \equiv GWN \Rightarrow (SN \overset{SK}{\longleftrightarrow} GWN)$

$A_8 : GWN| \equiv SN \Rightarrow (SN \overset{SK}{\longleftrightarrow} GWN)$

$A_9 : DO| \equiv DO \overset{\alpha}{\longleftrightarrow} GWN$

$A_{10} : GWN| \equiv DO \overset{\alpha}{\longleftrightarrow} GWN$

$A_{11} : SN| \equiv SN \overset{\delta}{\longleftrightarrow} GWN$

$A_{12} : GWN| \equiv SN \overset{\delta}{\longleftrightarrow} GWN$

7.1.5. BAN Logic Proof

Step 1: We can obtain S_1 from the message $Message_1$.

$$S_1 : GWN \lhd \{N_D^1\}_\alpha$$

Step 2: We can obtain S_2 from the message meaning rule using S_1 and A_{10}.

$$S_2 : GWN| \equiv DO| \sim (N_D^1)$$

Step 3: We can obtain S_3 from the freshness rule using S_2 and A_1.

$$S_3 : GWN| \equiv \#(N_D^1)$$

Step 4: We can obtain S_4 from the nonce verification rule using S_2 and S_3.

$$S_4 : GWN| \equiv DO| \equiv (N_D^1)$$

Step 5: We can obtain S_5 from the message $Message_2$.

$$S_5 : SN \lhd \{N_G^1, h(N_D^1||\alpha)\}_\delta$$

Step 6: We can obtain S_6 from the message meaning rule using S_5 and A_{11}.

$$S_6 : SN| \equiv GWN| \sim (N_G^1, h(N_D^1||\alpha))$$

Step 7: We can obtain S_7 from the freshness rule using S_6 and A_3.

$$S_7 : SN| \equiv \#(N_G^1, h(N_D^1||\alpha))$$

Step 8: We can obtain S_8 from the nonce verification rule using S_6 and S_7.

$$S_8 : SN| \equiv GWN| \equiv (N_G^1, h(N_D^1||\alpha))$$

Step 9: We can obtain S_9 from the message $Message_3$.

$$S_9 : GWN \lhd \{N_S^1\}_\delta$$

Step 10: We can obtain S_{10} from the message meaning rule using S_9 and A_{12}.

$$S_{10} : GWN| \equiv SN| \sim (N_S^1)$$

Step 11: We can obtain S_{11} from the nonce verification rule using A_2 and S_{10}.

$$S_{11} : GWN| \equiv SN| \equiv (N_S^1)$$

Step 12: We can obtain S_{12} and S_{13} from S_8 and S_{11}. SN and GWN can compute the session key $SK = h(h(N_D^1||\alpha) \oplus N_G^1 \oplus N_S^1)$.

$$S_{12} : GWN| \equiv SN| \equiv (SN \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 8)}$$
$$S_{13} : SN| \equiv GWN| \equiv (SN \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 6)}$$

Step 13: We can obtain S_{14} and S_{15} from the jurisdiction rule using S_{12} and A_8, and S_{13} and A_7, respectively.

$$S_{14} : GWN| \equiv (SN \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 7)}$$
$$S_{15} : SN| \equiv (SN \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 5)}$$

Step 14: We can obtain S_{16} from the message $Message_4$.

$$S_{16} : DO \lhd \{N_G^1, N_S^1\}_\alpha$$

Step 15: We can obtain S_{17} from the message meaning rule using A_9 and S_{16}.

$$S_{17} : DO| \equiv GWN| \sim (N_G^1, N_S^1)$$

Step 16: We can obtain S_{18} from the freshness rule using S_{17} and A_4.

$$S_{18} : DO| \equiv \#(N_G^1, N_S^1)$$

Step 17: We can obtain S_{19} from the nonce verification rule using S_{17} and S_{18}.

$$S_{19} : DO| \equiv GWN| \equiv (N_G^1, N_S^1)$$

Step 18: We can obtain S_{20} and S_{21} using S_4 and S_{19}. DO and GWN can compute the session key $SK = h(h(N_D^1||\alpha) \oplus N_G^1 \oplus N_S^1)$.

$$S_{20} : DO| \equiv GWN| \equiv (DO \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 2)}$$
$$S_{21} : GWN| \equiv DO| \equiv (DO \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 4)}$$

Step 19: We can obtain S_{22} and S_{23} using the jurisdiction rule using S_{20} and A_5, S_{21}, and A_6, respectively.

$$S_{22} : DO| \equiv (DO \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 1)}$$
$$S_{23} : GWN| \equiv (DO \overset{SK}{\longleftrightarrow} GWN) \quad \textbf{(Goal 3)}$$

7.2. RoR Model

In this section, we prove that the session key in the proposed scheme is secure, using the Real-or-Random (RoR) model [10]. To apply our scheme into the RoR model, we discuss the basic concepts of participants, adversaries, and queries. There are three participants in our scheme: $\mathcal{P}_{User}^{t_1}$, $\mathcal{P}_{Gateway}^{t_2}$, and $\mathcal{P}_{Sensor}^{t_3}$, where t_k is the participant instance of the user, the gateway, and the sensor node. We assume that an adversary $\mathcal{A}$ can control the whole network, which intercepts, deletes, inserts, and eavesdrops messages transmitted through a public channel. Moreover, $\mathcal{A}$ attempts to attack the network utilizing *Execute*, *CorruptSD*, *Reveal*, *Send*, and *Test* queries in the RoR model. Details of the queries are as follows.

- $Execute(\mathcal{P}_{User}^{t_1}, \mathcal{P}_{Gateway}^{t_2}, \mathcal{P}_{Sensor}^{t_3})$: The query *Execute* is a passive attack. This query explains that $\mathcal{A}$ can eavesdrop messages generated by $\mathcal{P}_{User}^{t_1}$, $\mathcal{P}_{Gateway}^{t_2}$, and $\mathcal{P}_{Sensor}^{t_3}$.
- $CorruptSD(\mathcal{P}_{User}^{t_1})$: This query is an active attack. By this query, $\mathcal{A}$ can obtain sensitive information extracted from the smart device of $\mathcal{P}_{User}^{t_1}$.
- $Reveal(\mathcal{P}^t)$: $\mathcal{A}$ can reveal the current session key SK.
- $Send(\mathcal{P}^t, \mathcal{M})$: Using the query *Send*, $\mathcal{A}$ can send a message $\mathcal{M}$ to $\mathcal{P}_{User}^{t_1}$, $\mathcal{P}_{Gateway}^{t_2}$, and $\mathcal{P}_{Sensor}^{t_3}$. Moreover, $\mathcal{A}$ can receive the return message. Therefore, this query is an active attack.
- $Test(P^t)$: If $\mathcal{A}$ performs a *Test* query, an unbiased coin C is flipped prior to starting the game. When the session key SK is fresh, $\mathcal{A}$ obtains $C = 1$. $\mathcal{A}$ also obtains $C = 0$ when the session key is not fresh. Otherwise, $\mathcal{A}$ will receive a null value ($\perp$). If $\mathcal{A}$ cannot distinguish between the session key and the random number, we can ensure that the proposed scheme can provide the security of the session key.

Security Proof

Theorem 1. *In the RoR model, an adversary $\mathcal{A}$ tries to calculate the session key of the proposed scheme in polynomial time. Let $Adv_{\mathcal{A}}(P)$ be the possibility that $\mathcal{A}$ breaks the security of the session key. We define Hash and PUF as the range space of hash function $h(.)$ and PUF function $PUF(.)$, respectively. In addition, we define q_h, q_p, and q_s as the number of Hash, PUF, and Send queries, respectively. l_D is the number of bits in biometric secret key BIO_D of the doctor, C' and s' are the Zipf's parameter [34].*

$$Adv_{\mathcal{A}}(P) \leq \frac{q_h^2}{|Hash|} + \frac{q_p^2}{|PUF|} + 2max\{C' q_s^{s'}, \frac{q_s}{2^{l_D}}\}$$

Proof. We follow the security proof as performed in [35–37]. In our proof, there are five games $Game_k$ where $k = 0, 1, 2, 3, 4$. We denote S_{Game_k} as the winning probability of the adversary $\mathcal{A}$ and $Pr[S_{Game_k}]$ as the advantage of the S_{Game_k}.

- $Game_0$: $Game_0$ is the starting game, where the adversary $\mathcal{A}$ picks up the random bit c. Therefore, we obtain the following:

$$Adv_{\mathcal{A}}(P) = |2Pr[S_{Game_0}] - 1| \tag{1}$$

- $Game_1$: In this game, $\mathcal{A}$ performs an eavesdropping attack, which is the *Execute* query in the RoR model. When obtaining messages $\{D_{TID}, S_{TID}, M_{D1}, V_{D1}\}$, $\{D_{TID}, S_{TID}, M_{G1}, M_{G2}, V_{G1}\}$, $\{M_{S1}, V_{S1}\}$, and $\{M_{G3}, V_{G2}\}$, $\mathcal{A}$ carries out *Test* and *Reveal* queries to distinguish between the session key SK and a random number. To obtain the session key $SK = h(h(N_D^1||\alpha) \oplus N_G^1 \oplus N_S^1)$, $\mathcal{A}$ needs N_D^1, N_G^1, and N_S^1, which are random numbers generated by the user (doctor), the gateway, and the sensor node, respectively. α is the shared secret parameter between the gateway and the user. For these reasons, the adversary $\mathcal{A}$ cannot compute the session key SK. This means that $\mathcal{A}$ does not enhance the probability compared with the $Game_0$.

$$[Pr[S_{Game_1}]] = [Pr[S_{Game_0}]] \tag{2}$$

- $Game_2$: In $Game_2$, the adversary $\mathcal{A}$ performs *Send* and *Hash* queries. In the message $\{D_{TID}, S_{TID}, M_{D1}, V_{D1}\}$, $\{D_{TID}, S_{TID}, M_{G1}, M_{G2}, V_{G1}\}$, $\{M_{S1}, V_{S1}\}$, and $\{M_{G3}, V_{G2}\}$, parameters D_{TID}, S_{TID}, V_{D1}, V_{G1}, V_{S1}, and V_{G2} are masked by the cryptographic one-way hash function, which provides resistance against hash collision. Moreover, random numbers N_D^1, N_G^1, N_S^1, and the hash functions are contained in M_{D1}, M_{G1}, M_{G2}, M_{G3}, and M_{S1}. Therefore, there is no collision problem when $\mathcal{A}$ performs a *Hash* query. We apply the birthday paradox [38] and obtain the result as follows:

$$|Pr[S_{Game_2}] - Pr[S_{Game_1}]| \leq \frac{q_h^2}{|Hash|} \tag{3}$$

- $Game_3$: $Game_3$ is similar to $Game_2$. $\mathcal{A}$ performs *Send* and *PUF* queries. As explained in Section 3.3, the physical function $PUF(.)$ has a secure property. Therefore, we can obtain the following inequation:

$$|Pr[S_{Game_3}] - Pr[S_{Game_2}]| \leq \frac{q_p^2}{|PUF|} \tag{4}$$

- $Game_4$: In the final game $Game_4$, $\mathcal{A}$ performs a $CorruptSD$ query and extracts sensitive data $\{\beta, \theta, Ver_D, D_{TID}, P_D\}$ from the smart device of the user. $\mathcal{A}$ attempts to calculate parameters α and R_{SG}^2 from $\beta = \alpha \oplus h(GPW_D||R_{SG}^2)$ and $\theta = R_{SG}^2 \oplus h(R_D||GPW_D)$, respectively. Since parameters R_D and $GPW_D = h(PW_D||R_D)$ are composed of the password and biometrics, $\mathcal{A}$ must guess these parameters. Therefore, $\mathcal{A}$ cannot enhance the probability because guessing the password and biometrics is a computationally infeasible task. According to Zipf's law [34], we can make the following inequation:

$$|Pr[S_{Game_4}] - Pr[S_{Game_2}]| \leq max\{C' q_s^{s'}, \frac{q_s}{2^{l_D}}\} \tag{5}$$

When the games are completed, the adversary $\mathcal{A}$ obtains the guessed bit c. Therefore, it is clear that

$$Pr[S_{Game_4}] = \frac{1}{2} \tag{6}$$

By (2) and (3), we can obtain the following equation:

$$\frac{1}{2} Adv_{\mathcal{A}}(P) = |Pr[S_{Game_0}] - \frac{1}{2}| = |Pr[S_{Game_1}] - \frac{1}{2}| \tag{7}$$

We can obtain the following equation using (6) and (7):

$$\frac{1}{2}Adv_{\mathcal{A}}(P) = |Pr[S_{Game_1}] - Pr[S_{Game_4}]| \tag{8}$$

Applying the triangular inequality, we obtain the following result:

$$\begin{aligned}\frac{1}{2}Adv_{\mathcal{A}}(P) &= |Pr[S_{Game_1}] - Pr[S_{Game_4}]| \\ &\leq |Pr[S_{Game_1}] - Pr[S_{Game_3}]| \\ &+|Pr[S_{Game_3}] - Pr[S_{Game_4}]| \\ &\leq |Pr[S_{Game_1}] - Pr[S_{Game_2}]| \\ &+|Pr[S_{Game_2}] - Pr[S_{Game_3}]| \\ &+|Pr[S_{Game_3}] - Pr[S_{Game_4}]| \\ \leq \frac{q_h^2}{2|Hash|} &+ \frac{q_p^2}{2|PUF|} + max\{C'q_s^{s'}, \frac{q_s}{2^{l_D}}\}\end{aligned} \tag{9}$$

Finally, we obtain the required result multiplying (9) by 2:

$$Adv_{\mathcal{A}}(P) \leq \frac{q_h^2}{|Hash|} + \frac{q_p^2}{|PUF|} + 2max\{C'q_s^{s'}, \frac{q_s}{2^{l_D}}\}$$

Thus, we have proven Theorem 1.

□

7.3. AVISPA Simulation

We simulate the proposed scheme using AVISPA [11,12] to analyze the security features of our scheme. AVISPA is a formal verification tool that can detect security vulnerabilities regarding replay and man-in-the-middle attacks. Therefore, various authentication schemes [39–41] have been simulated by using AVISPA.

To simulate our protocol, we need to create a code written in the High-Level Protocol Specification Language (HLPSL). The code written in HLPSL is converted to the Intermediate Format (IF) by the translator. Then, the translator inputs the IF into back-ends. AVISPA has four back-ends, named On-the-Fly Model Checker (OFMC), Constraint Logic-based Attack Searcher (CL-AtSe), SAT-based Model Checker (SATMC), and Three Automata based on Automatic Approximations for Analysis of Security Protocol (TA4SP). In this paper, the OFMC and CL-AtSe back-ends are used because these back-ends provide exclusive-OR operations. Lastly, we obtain the Output Format (OF), which is the security analysis result of the protocol. If we obtain a "SAFE" message in the summary of OF, we can consider that the protocol is secure against replay and man-in-the-middle attacks.

7.3.1. HLPSL Specification

In this section, we explain the HLPSL code of our scheme. There are three basic roles in HLPSL: the doctor *DO*, the gateway *GW*, and the sensor node *SN*. With these roles, we describe the session and the environment roles. The goals, the environment, and the session of our scheme written in HLPSL are shown in Figure 7.

```
role session(SN, GW, DOC : agent, SKdosn, SKgwsn, SKgwdo : symmetric_key, PUF,H : hash_func)

def=
local SN1, SN2, SN3, RV1, RV2, RV3 : channel(dy)
composition
doctor(DOC, GW, SN, SKdosn, SKgwdo, SKgwsn, PUF,H, SN1, RV1)
/\ gateway(DOC, GW, SN, SKdosn, SKgwdo, SKgwsn, PUF,H, SN2, RV2)
/\ sensor(DOC, GW, SN, SKdosn, SKgwdo, SKgwsn, PUF,H, SN3, RV3)

end role

role environment()
def=
const sn, gw, doc : agent,
                    puf, h : hash_func,
                    skdosn, skgwsn, skgwdo : symmetric_key,
                    doc_gw_n1d, doc_gw_n1g, doc_gw_n1s, gw_sn_n1s : protocol_id,
                    sp1, sp2, sp3, sp4, sp5, sp6 : protocol_id,
                    did, sidj, dtid, stid : text
intruder_knowledge = {did, sidj, dtid, stid, puf, h}
composition
session(doc, gw, sn, skdosn, skgwdo, skgwsn, puf, h)
/\session(i, gw, sn, skdosn, skgwdo, skgwsn, puf, h)
/\session(doc, i, sn, skdosn, skgwdo, skgwsn, puf, h)
/\session(doc, gw, i, skdosn, skgwdo, skgwsn, puf, h)

end role

goal
secrecy_of sp1,sp2,sp3,sp4,sp5,sp6
authentication_on doc_gw_n1d
authentication_on doc_gw_n1g
authentication_on doc_gw_n1s
authentication_on gw_sn_n1s
end goal

environment()
```

Figure 7. Role specification for the session, environment, and goals.

We show the role of the doctor in Figure 8. When state 1 starts, the doctor receives a start message and generates the registration request message R_{req}. Then, the doctor computes GPW_D with his password PW_D, the biometrics BIO_D, and sends $\{D_{ID}, GPW_D, R_{req}\}$ to the gateway via a secure channel. After this, the doctor receives $\{\beta, D_{TID}, R^2_{SG}\}$ from the gateway and computes Ver_D and θ in state 2. The doctor stores $\{\beta, \theta, Ver_D, D_{TID}, P_D\}$ in the smart device. With these parameters, the doctor sends a login and authentication request message $\{D_{TID}, S_{TID}, M_{D1}, V_{D1}\}$ to the gateway via a public channel. $witness(DOC, GW, doc\_gw\_n1d, N'^1_D)$ indicates the freshness of N^1_D. When the doctor receives the message $\{M_{G3}, V_{G2}\}$ in state 3, the doctor performs $request(GW, DOC, doc\_gw\_n1s, N'^1_S)$ and $request(GW, DOC, doc\_gw\_n1g, N'^1_G)$, which represent the freshness acceptance of the random nonces N^1_G and N^1_S.

```
%%%AVISPA Simulation
role doctor(SN, GW, DOC : agent, SKdosn, SKgwdo, SKgwsn : symmetric_key, PUF,H : hash_func, SND,RCV : channel(dy))

played_by DOC
def=
local State : nat,
                    ST, SIDj, CH1, RE1, ASID, R3SG, DELTA, STID : text,
                    DID, PWD, BIOD, GPWD, Rreq, S, R1SG, R2SG, ALPHA, BETA, OMEGA, DTID, THETA, VerD : text,
                    N1D, N1G, N1S, MD1, VD1, MG1, MG2, MG3, VG1, MS1, VS1, VG2, SnewTID, SK, DnewTID : text
                    const sp1, sp2, sp3, sp4, sp5, sp6, doc_gw_n1d, doc_gw_n1g, doc_gw_n1s, gw_sn_n1s : protocol_id

init State := 0
transition
%%%%%%%%Doctor registration phase
1. State = 0 /\ RCV(start) =|>
State' := 1
/\ Rreq' := new()
/\ GPWD' := H(PWD. BIOD)
/\ SND({DID. GPWD'. Rreq'}_SKgwdo)
/\ secret({DID, BIOD}, sp4, DOC)

2. State = 1 /\ RCV({xor(H(DID. S), H(h(PWD. BIOD). R2SG')). H(H(DID. S). R2SG'. R1SG'). R2SG'}_SKgwdo) =|>
State' := 2
/\ THETA' := xor(R2SG', H(BIOD. H(PWD. BIOD)))
/\ VerD' := H(DID. H(BIOD. H(PWD. BIOD)). BIOD)
/\ secret({R2SG'}, sp6, DOC)
%Mutual authentication
/\ N1D' := new()
/\ MD1' := xor(N1D', H(H(H(DID. S). R2SG'. R1SG'). H(DID. S)))
/\ VD1' := H(N1D'. H(H(H(DID. S). R2SG'. R1SG'). H(DID. S)))
/\ SND(H(H(DID. S). R2SG'. R1SG'). MD1'. VD1')
/\ witness(DOC,GW,doc_gw_n1d,N1D')

3. State = 2 /\ RCV(xor(xor(N1G', N1S'), H(H(DID. S). H(H(DID. S). R2SG'. R1SG'))). H(xor(N1G',N1S'). H(H(DID. S). N1D'.
xor(N1G', N1S')). xor(xor(H(N1D'.H(DID. S)), N1G'), N1S'))) =|>
State' := 3
/\ request(GW,DOC,doc_gw_n1s,N1S')
/\ request(GW,DOC,doc_gw_n1g,N1G')

end role
```

Figure 8. Role specification for the doctor.

7.3.2. Simulation Result

We perform simulations using the OFMC and CL-AtSe back-ends and show the simulation result of the proposed scheme in Figure 9. If the summary message is "SAFE", this indicates that the proposed scheme is secure against replay and man-in-the-middle attacks. As with the simulation result shown in Figure 9, both summaries simulated in the OFMC and CL-AtSe back-ends are "SAFE". Thus, the proposed scheme can prevent replay and man-in-the-middle attacks.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/KDK5.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 5.44s
  visitedNodes: 1432 nodes
  depth: 12 plies
```

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/span/span/testsuite/results/KDK5.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS
  Analysed   : 0 states
  Reachable  : 0 states
  Translation: 0.05 seconds
```

Figure 9. The AVISPA simulation result of the proposed scheme.

7.4. Informal Analysis

In this section, we show the security features of the proposed scheme, including those that protect against offline password guessing, impersonation, replay, man-in-the-middle, physical, cloning, privileged insider, session-specific random number leakage, and verification table leakage attacks. Moreover, the proposed scheme can ensure user anonymity, perfect forward secrecy, and mutual authentication.

7.4.1. User Anonymity

We assume that an adversary $\mathcal{A}$ obtains the stolen smart device of a doctor (user) and extracts $\{\beta, \theta, Ver_D, D_{TID}, P_D\}$. However, $\mathcal{A}$ cannot compute the real identity of the doctor because the pseudo identity of the doctor D_{TID} is masked by the hash function and updated in every session. Since the parameters $\beta = \alpha \oplus h(GPW_D||R^2_{SG})$ and $\theta = h(R_D||GPW_D) \oplus R^2_{SG}$ stored in the smart device are masked in the biometric template of the doctor, the $\mathcal{A}$ has difficulty in guessing the real identity of the doctor. Hence, $\mathcal{A}$ cannot obtain the real identity of the doctor. Therefore, we demonstrate that the proposed scheme can ensure user anonymity.

7.4.2. Offline Password Guessing Attack

$\mathcal{A}$ obtains a doctor's smart device and obtains $\{\beta, \theta, Ver_D, D_{TID}, P_D\}$ from the device using a power analysis attack. Then, $\mathcal{A}$ attempts to guess the password of the doctor using the extracted parameters. Unfortunately, $\mathcal{A}$ cannot guess the password of the doctor because we use the biometrics in the proposed scheme. Since $GPW_D = h(PW_D||R_D)$, $\mathcal{A}$ must guess not only the password PW_D but also the biometrics BIO_D of the doctor at the same time. Note that R_D is the result of the fuzzy extractor, which is expressed as $R_D = Rep(BIO_D, P_D)$. However, this process is a computationally infeasible task. Thus, the proposed scheme can prevent offline password guessing attacks.

7.4.3. Impersonation Attack

Assume that an adversary $\mathcal{A}$ tries to impersonate a legitimate doctor using parameters $\{\beta, \theta, Ver_D, D_{TID}, P_D\}$, which are stored in the doctor's device. Then, $\mathcal{A}$ attempts to calculate the login request message $\{D_{TID}, S_{TID}, M_{D1}, V_{D1}\}$. However, $\mathcal{A}$ cannot calculate $M_{D1} = N^1_D \oplus h(D_{TID}||\alpha)$ and $V_{D1} = h(N^1_D||D_{TID}||\alpha||S_{TID})$ because $\mathcal{A}$ cannot calculate $\alpha = \beta \oplus h(GPW_D||R^2_{SG})$. Hence, the proposed scheme is secure against impersonation attacks.

7.4.4. Replay Attack

Assume that an adversary $\mathcal{A}$ intercepts authentication request messages $\{D_{TID}, S_{TID}, M_{D1}, V_{D1}\}$, $\{D_{TID}, S_{TID}, M_{G1}, M_{G2}, V_{G1}\}$, and sends messages to authenticate the gateway and the sensor node at other sessions. However, each entity checks the freshness of N^1_D, N^1_G, and N^1_S, which are random nonces generated by the doctor, the gateway, and the sensor node, respectively. Therefore, the proposed scheme is secure against replay attacks.

7.4.5. Man-in-the-Middle Attack

We show that $\mathcal{A}$ cannot generate the login request message $\{D_{TID}, S_{TID}, M_{D1}, V_{D1}\}$, according to Section 7.4.3. Moreover, $\mathcal{A}$ cannot compute $\{D_{TID}, S_{TID}, M_{G1}, M_{G2}, V_{G1}\}$, $\{M_{S1}, V_{S1}\}$, and $\{M_{G3}, V_{G2}\}$ because each message is masked in the shared secret parameter α and δ. Thus, the proposed scheme can prevent man-in-the-middle attacks.

7.4.6. Physical and Cloning Attacks

We can assume that $\mathcal{A}$ physically captures a sensor node SN_1 and tries to authenticate the gateway as SN_1. To do this, $\mathcal{A}$ obtains the parameters of SN_1 $\{\delta, P_{SN}, S_{TID}\}$ using a power analysis attack. Then, $\mathcal{A}$ attempts to authenticate as a legitimate sensor node SN_1 using parameters $\{\delta, P_{SN}, S_{TID}\}$ or by cloning the sensor node SN_1. When $\mathcal{A}$ receives $\{D_{TID}, S_{TID}, M_{G1}, M_{G2}, V_{G1}\}$ from the gateway, $\mathcal{A}$ computes $CH^*_1 = M_{G3} \oplus$

$h(\delta||D_{TID}||S_{TID})$. However, $\mathcal{A}$ cannot compute RE_1 because the function $PUF(.)$ is a physically unclonable circuit and cannot duplicate, according to Section 3.3. Therefore, $\mathcal{A}$ cannot compute $R^1_{SN} = Rep(RE_1, P_{SN})$ and $AS_{ID} = h(R^1_{SN}||S_{ID})$ to calculate M_{S1} and V_{S1}. Thus, the proposed scheme is secure against physical and cloning attacks.

7.4.7. Privileged Insider Attack

Assume that a privileged insider $\mathcal{A}$ obtains the registration request message $\{D_{ID}, GPW_D, R_{req}\}$ of a doctor and obtains parameters $\{\beta, \theta, Ver_D, D_{TID}, P_D\}$, extracted from the stolen smart device of the doctor using a power analysis attack, and $\mathcal{A}$ attempts to impersonate as the doctor. To compute the login request message $\{D_{TID}, S_{TID}, M_{D1}, V_{D1}\}$, $\mathcal{A}$ must calculate the shared secret parameter α. However, $\mathcal{A}$ cannot calculate $\alpha = h(GPW_D||R^2_{SG})$ because the parameter R_D in $GPW_D = h(PW_D||R_D)$ is generated by the biometrics of the doctor. Moreover, $\mathcal{A}$ must guess the password PW_D of the doctor to calculate $GPW_D = h(PW_D||R_D)$, and it is a computationally infeasible task to guess R_D and PW_D at the same time. Therefore, the proposed scheme can prevent privileged insider attacks.

7.4.8. Session-Specific Random Number Leakage Attack

Suppose that $\mathcal{A}$ obtains random nonces N^1_D, N^1_G, and N^1_S. Then, $\mathcal{A}$ tries to calculate the session key $SK = h(h(N^1_D||\alpha) \oplus N^1_G \oplus N^1_S)$. However, $\mathcal{A}$ cannot compute the session key SK without knowing the shared secret parameter α. Since α is masked by the hash functions, $\mathcal{A}$ cannot calculate α. Thus, the proposed scheme has resistance against session-specific random number leakage attacks.

7.4.9. Verification Table Leakage Attack

If $\mathcal{A}$ obtains the verification table $\{\omega, R^1_{SG}, D_{TID}\}$, $\{AS_{ID}, S_{TID}, CH_1\}$ of the gateway, $\mathcal{A}$ attempts to calculate the session key SK or impersonate a doctor. However, $\mathcal{A}$ cannot calculate the shared secret parameter $\alpha = \omega \oplus h(R^1_{SG}||s)$ and $\delta = h(AS_{ID}||s)$ without the master key s of the gateway. Therefore, it is difficult for $\mathcal{A}$ to compute the session key $SK = h(h(N^1_D||\alpha) \oplus N^1_G \oplus N^1_S)$ or impersonate a doctor. Therefore, the proposed scheme can prevent verification table leakage attacks.

7.4.10. Perfect Forward Secrecy

If $\mathcal{A}$ obtains the master key s of the gateway, $\mathcal{A}$ attempts to compute the session key $SK = h(h(N^1_D||\alpha) \oplus N^1_G \oplus N^1_S)$. However, $\mathcal{A}$ cannot compute $\alpha = h(D_{ID}||s)$ without the real identity of the doctor, and all random nonces are masked by hash functions. Therefore, $\mathcal{A}$ cannot calculate SK. For this reason, the proposed scheme ensures perfect forward secrecy.

7.4.11. Mutual Authentication

To ensure mutual authentication, each entity checks the validity of $V^*_{D1} \stackrel{?}{=} V_{D1}$, $V^*_{G1} \stackrel{?}{=} V_{G1}$, $V^*_{S1} \stackrel{?}{=} V_{S1}$, and $V^*_{G2} \stackrel{?}{=} V_{G2}$. Furthermore, all participants check the freshness of random nonces N^1_D, N^1_G, and N^1_S. When the verification processes are successful, we can demonstrate that the participants of the proposed scheme authenticate each other. Therefore, the proposed scheme ensures mutual authentication.

8. Performance

In this section, we compare the security features of the proposed scheme with other related schemes [7,18–20,25]. Moreover, we show the communication costs, computation costs, and energy consumption of the proposed scheme.

8.1. Security Features Comparison

We present the security features of the proposed scheme compared with related schemes [7,18–20,25]. In Table 3, we consider various security attacks and functionalities. The security features and the functionalities are as follows: $SP1$: resistance against smart

device theft attack, $SP2$: resistance against offline password guessing attack, $SP3$: resistance against impersonation attack, $SP4$: resistance against replay attack, $SP5$: resistance against privileged insider attack, $SP6$: resistance against physical and cloning attacks, $SP7$: resistance against session-specific random number leakage attack, $SP8$: resistance against verification table leakage attack, $SP9$: ensuring user anonymity, $SP10$: ensuring perfect forward secrecy, $SP11$: ensuring mutual authentication, $SP12$: performing RoR model, $SP13$: performing AVISPA simulation, $SP14$: performing BAN logic proof. Therefore, our scheme can provide a secure authentication process compared with [7,18–20].

Table 3. Security and functionality features comparison.

Security Properties	[18]	[19]	[20]	[25]	[7]	Proposed
$SP1$	×	✓	✓	✓	×	✓
$SP2$	✓	✓	✓	✓	×	✓
$SP3$	✓	✓	✓	✓	×	✓
$SP4$	✓	✓	✓	✓	✓	✓
$SP5$	×	✓	✓	✓	×	✓
$SP6$	×	×	×	✓	×	✓
$SP7$	×	×	✓	✓	✓	✓
$SP8$	–	–	–	–	–	✓
$SP9$	×	✓	✓	✓	×	✓
$SP10$	✓	✓	✓	✓	✓	✓
$SP11$	✓	✓	✓	✓	✓	✓
$SP12$	✓	–	–	✓	–	✓
$SP13$	✓	✓	✓	–	✓	✓
$SP14$	–	✓	✓	–	–	✓

✓: Provides the security/functionality feature. ×: Does not provide the security/functionality feature. –: Does not consider the security/functionality feature.

8.2. Communication Costs Comparison

In this section, we compare the communication costs of the proposed scheme with existing schemes [7,18–20,25]. According to [35], we suppose that the SHA-1 hash digest, identity, random number, PUF challenge–response pair, timestamp, and ECC point are 160, 160, 128, 128, 32, and 320 bits, respectively. Therefore, the communication costs of the proposed scheme can be described as follows.

- Message 1 : The message $\{D_{TID}, S_{TID}, M_{D1}, V_{D1}\}$ requires $(160 + 160 + 160 + 160) = 640$ bits.
- Message 2 : The message $\{D_{TID}, S_{TID}, M_{G1}, M_{G2}, V_{G1}\}$ needs $(160 + 160 + 160 + 160 + 160) = 800$ bits.
- Message 3 : The message $\{M_{S1}, V_{S1}\}$ needs $(160 + 160) = 320$ bits.
- Message 4 : The message $\{M_{G3}, V_{G2}\}$ requires $(160 + 160) = 320$ bits.

Therefore, the total communication costs of our scheme are $640 + 800 + 320 + 320 = 2080$ bits. In Table 4, we show the total communication costs of our scheme and other related schemes. Consequently, we demonstrate that our scheme has more efficient communication costs than other related schemes [7,18–20,25].

Table 4. Comparison of communication costs.

Schemes	Total Communication Costs	Messages
Li et al. [18]	2880 bits	4 messages
Shin et al. [19]	3328 bits	4 messages
Ali et al. [20]	2240 bits	4 messages
Chen et al. [25]	2880 bits	5 messages
Masud et al. [7]	2176 bits	4 messages
Proposed	2080 bits	4 messages

8.3. Computation Costs Comparison

We compare the computation costs of the proposed scheme with [7,18–20,25]. According to [42,43], we define T_{RNG}, T_H, T_{EM}, T_{EA}, T_F, and T_{PUF} as the random number generation (≈0.0539 s), hash function (≈0.00023 s), ECC multiplication (≈0.2226 s), ECC addition (≈0.00288 s), fuzzy extractor (≈0.268 s), and PUF operation time (≈0.012 s), respectively. Furthermore, we ignore the execution time of exclusive-OR ($\oplus$) operations because it is computationally negligible.

The total computation costs of our scheme are slightly higher than those of Masud et al.'s scheme [7] as shown in Table 5. However, our scheme has a much higher security level than [7] using the fuzzy extractor and PUF. Moreover, our scheme is more efficient and lightweight than previous schemes [18–20,25] that utilize ECC, the fuzzy extractor, and PUF.

Table 5. Comparison of computational costs.

Schemes	User	Gateway	Sensor Node	Total	Total Cost (s)
Li et al. [18]	$1T_{RNG} + 8T_H + 3T_{EM}$	$1T_{RNG} + 8T_H + T_{EM}$	$1T_{RNG} + 4T_H + 2T_{EM}$	$3T_{RNG} + 20T_H + 6T_{EM}$	1.502
Shin et al. [19]	$1T_{RNG} + 1T_F + 14T_H + 2T_{EM}$	$12T_H + 1T_{EM}$	$1T_{RNG} + 5T_H + 1T_{EM}$	$2T_{RNG} + 1T_F + 31T_H + 4T_{EM}$	1.232
Ali et al. [20]	$1T_{RNG} + 1T_F + 3T_H + 2T_{EM}$	$1T_{RNG} + 4T_H + 2T_{EM}$	$1T_H$	$2T_{RNG} + 1T_F + 8T_H + 4T_{EM}$	1.268
Chen et al. [25]	$1T_{RNG} + 2T_F + 14T_H + 1T_{PUF}$	$8T_H$	$1T_{RNG} + 1T_F + 8T_H$	$2T_{RNG} + 3T_F + 30T_H + 1T_{PUF}$	0.919
Masud et al. [7]	$1T_{RNG} + 3T_H$	$4T_{RNG} + 3T_H$	$2T_{RNG} + 2T_H$	$7T_{RNG} + 8T_H$	0.379
Proposed	$1T_{RNG} + 1T_F + 11T_H$	$1T_{RNG} + 15T_H$	$1T_{RNG} + 1T_F + 8T_H + 1T_{PUF}$	$3T_{RNG} + 2T_F + 34T_H + 1T_{PUF}$	0.717

8.4. Energy Consumption Comparison

In this section, we compare the energy consumption of our scheme with [7,18–20,25]. We follow the battery consumption model used in [44], where the energy consumption for sending and receiving a bit are taken as 4.602 mJ and 2.34 mJ, respectively [45]. Therefore, the total energy consumption of our scheme is 4867 mJ. Table 6 shows the total energy consumption of the proposed scheme and [7,18–20,25]. The result indicates that our scheme is more efficient in terms of energy consumption than other related schemes.

Table 6. Comparison of energy consumption.

Schemes	Total Energy Consumption
Li et al. [18]	6739 mJ
Shin et al. [19]	7788 mJ
Ali et al. [20]	5242 mJ
Chen et al. [25]	6739 mJ
Masud et al. [7]	5092 mJ
Proposed	4867 mJ

9. Conclusions

In this paper, we review Masud et al.'s scheme and prove that their scheme is vulnerable to offline password guessing, impersonation, and privileged insider attacks. We also discover that Masud et al.'s scheme cannot ensure user anonymity and has a device update problem. To improve the security level and overcome the security weaknesses of Masud et al.'s scheme, we propose a provably secure three-factor-based mutual authentication and key agreement scheme for WMSNs. Our scheme has light weight, using

only hash functions and exclusive-OR operators; it provides a secure login process to the doctor using the fuzzy extractor, and it provides resistance against cloning and physical attacks using PUF. We ensure the mutual authentication utilizing BAN logic and prove the session key security of our scheme using the RoR model. We also show that our scheme offers resistance against replay and man-in-the-middle attacks by utilizing the AVISPA simulation tool. We prove that our scheme is secure against various attacks, including offline password, impersonation, sensor node capture, and verification table leakage attacks, through informal analysis. Furthermore, we demonstrate that our scheme can provide user anonymity, perfect forward secrecy, and mutual authentication. Finally, we estimate the computation costs, communication costs, and energy consumption of our scheme and compare it with other related schemes. Our result shows that the proposed scheme can provide doctors and patients with more secure services for WMSNs. In the future, we will develop and implement our scheme, considering performance evaluation and result analysis, confirming its suitability for practical WMSN environments.

Author Contributions: Conceptualization, D.K.; Formal analysis, D.K. and Y.P (Yohan Park); Methodology, D.K. and Y.P.(Yohan Park); Software, D.K.; Validation, Y.P. (Yohan Park) and Y.P. (Youngho Park); Formal Proof, Y.P. (Youngho Park); Writing—original draft, D.K.; Writing—review and editing, Y.P. (Yohan Park), and Y.P. (Youngho Park); Supervision, Y.P. (Youngho Park). All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education under grant 2020R1I1A3058605, and in part by the BK21 FOUR project, funded by the Ministry of Education, Korea under grant 4199990113966.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Lara, E.; Aguilar, L.; Sanchez, M.A.; García, J.A. Lightweight authentication protocol for M2M communications of resource-constrained devices in industrial Internet of Things. *Sensors* **2020**, *20*, 501. [CrossRef] [PubMed]
2. Park, K.; Noh, S.; Lee, H.; Das, A.K.; Kim, M.; Park, Y.; Wazid, M. LAKS-NVT: Provably secure and lightweight authentication and key agreement scheme without verification table in medical internet of things. *IEEE Access* **2020**, *20*, 119387–119404. [CrossRef]
3. Oh, J.; Yu, S.; Lee, J.; Son, S.; Kim, M.; Park, Y. A secure and lightweight authentication protocol for IoT-based smart homes. *Sensors* **2021**, *21*, 1488. [CrossRef] [PubMed]
4. Abdulsalam, Y.; Hossain, M.S. COVID-19 networking demand: An auction-based mechanism for automated selection of edge computing services. *IEEE Trans. Netw. Sci. Eng.* **2020**, 1–11. [CrossRef]
5. Aileni, R.M.; Suciu, G. IoMT: A blockchain perspective. In *Decentralised Internet of Things*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 199–215.
6. Rahman, M.; Jahankhani, H. Security vulnerabilities in existing security mechanisms for IoMT and potential solutions for mitigating cyber-attacks. In *Information Security Technologies for Controlling Pandemics*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 307–334.
7. 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**. [CrossRef]
8. 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.
9. Burrows, M.; Abadi, M.; Needham, R.M. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [CrossRef]
10. Abdalla, M.; Fouque, P.; 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.
11. AVISPA. Automated Validation of Internet Security Protocols and Applications. Available online: http://www.avispa-project.org/ (accessed on 20 July 2021).
12. SPAN: A Security Protocol Animator for AVISPA. Available online: http://www.avispa-project.org/ (accessed on 20 July 2021).
13. 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]

14. 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]
15. Mir, O.; Munilla, J.; Kumari, S. Efficient anonymous authentication with key agreement protocol for wireless medical sensor networks. *Peer-to-Peer Netw. Appl.* **2017**, *10*, 79–91. [CrossRef]
16. Wu, F.; Li, X.; Sangaiah, A.K.; Xu, L.; Kumari, S.; Wu, L.; Shen, J. A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks. *Future Gener. Comput. Syst.* **2018**, *82*, 727–737. [CrossRef]
17. Challa, S.; Das, A.K.; Odelu, V.; Kumar, N.; Kumari, S.; Khan, M.K.; Vasilakos, A.V. An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks. *Comput. Electr. Eng.* **2018**, *69*, 534–554. [CrossRef]
18. 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]
19. Shin, S.; Kwon, T. A privacy-preserving authentication, authorization, and key agreement scheme for wireless sensor networks in 5G-integrated Internet of Things. *IEEE Access* **2020**, *8*, 67555–67571. [CrossRef]
20. Ali, Z.; Ghani, A.; Khan, I.; Chaudhry, S.A.; Islam, S.H.; Giri, D. A robust authentication and access control protocol for securing wireless healthcare sensor networks. *J. Inf. Secur. Appl.* **2020**, *52*, 102502. [CrossRef]
21. Hsu, C.L.; Le, T.V.; Hsieh, M.C.; Tsai, K.Y.; Lu, C.F.; Lin, T.W. Three-factor UCSSO scheme with fast authentication and privacy protection for telecare medicine information systems. *IEEE Access* **2020**, *8*, 196553–196566. [CrossRef]
22. 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]
23. Byun, J.W. End-to-end authenticated key exchange based on different physical unclonable functions. *IEEE Access* **2019**, *7*, 102951–102965. [CrossRef]
24. Fang, D.; Qian, Y.; Hu, R.Q. A flexible and efficient authentication and secure data transmission scheme for IoT applications. *IEEE Internet Things J.* **2020**, *7*, 3474–3484. [CrossRef]
25. Chen, Y.; Chen, J. An efficient mutual authentication and key agreement scheme without password for wireless sensor networks. *J. Supercomput.* **2021**, 1–23. [CrossRef]
26. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
27. Canetti, R.; Krawczyk, H. Universally composable notions of key exchange and secure channels. In *Lecture Notes in Computer Science, Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques—Advances in Cryptology (EUROCRYPT'02), Amsterdam, The Netherlands, 28 April–2 May 2002*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 337–351.
28. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 1999; pp. 388–397.
29. Park, K.; Park, Y.; Park, Y.; Das, A.K. 2PAKEP: Provably secure and efficient two-party authenticated key exchange protocol for mobile environment. *IEEE Access* **2018**, *6*, 30225–30241. [CrossRef]
30. Park, Y.; Park, K.; Park, Y. Secure user authentication scheme with novel server mutual verification for multiserver environments. *Int. J. Commun. Syst.* **2019**, *32*, e3929. [CrossRef]
31. Lee, J.; Yu, S.; Kim, M.; Park, Y.; Lee, S.; Chung, B. Secure key agreement and authentication protocol for message confirmation in vehicular cloud computing. *Appl. Sci.* **2020**, *10*, 6268. [CrossRef]
32. Shashidhara, R.; Nayak, S.K.; Das, A.K.; Park, Y. On the design of lightweight and secure mutual authentication system for global roaming in resource-limited mobility networks. *IEEE Access* **2021**, *9*, 12879–12895. [CrossRef]
33. 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. Eng.* **2021**, *2021*, 9954089. [CrossRef]
34. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf's law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791. [CrossRef]
35. Banerjee, S.; Odelu, V.; Das, A.K.; Chattopadhyay, S.; Rodrigues, J.J.; Park, Y. Physically secure lightweight anonymous user authentication protocol for internet of things using physically unclonable functions. *IEEE Access* **2019**, *7*, 85627–85644. [CrossRef]
36. Wazid, M.; Bagga, P.; Das, A.K.; Shetty, S.; Rodrigues, J.J.; Park, Y. AKM-IoV: Authenticated key management protocol in fog computing-based Internet of vehicles deployment. *IEEE Internet Things J.* **2019**, *6*, 8804–8817 [CrossRef]
37. 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**. [CrossRef]
38. Boyko, V.; MacKenzie, P.; Patel, S. Provably secure password-authenticated key exchange using Diffie-Hellman. In *Lecture Notes in Computer Science, Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Bruges, Belgium, 14–18 May 2000*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 156–171.
39. Son, S.; Lee, J.; Kim, M.; Yu, S.; Das, A.K.; Park, Y. Design of secure authentication protocol for cloud-assisted telecare medical information system using blockchain. *IEEE Access* **2020**, *8*, 192177–192191. [CrossRef]
40. Lee, J.; Yu, S.; Kim, M.; Park, Y.; Das, A.K. On the design of secure and efficient three-factor authentication protocol using honey list for wireless sensor networks. *IEEE Access* **2020**, *8*, 107046–107062. [CrossRef]
41. Kim, M.; Lee, J.; Park, K.; Park, Y.; Park, K.H.; Park, Y. Design of secure decentralized car-sharing system using blockchain. *IEEE Access* **2021**, *9*, 54796–54810. [CrossRef]

42. Kilinc, H.H.; Yanik, T. A survey of SIP authentication and key agreement schemes. *IEEE Commun. Surv. Tutor.* **2013**, *16*, 1005–1023. [CrossRef]
43. Gope, P.; Sikdar, B. Lightweight and privacy-preserving two-factor authentication scheme for IoT devices. *IEEE Internet Things J.* **2019**, *6*, 580–589. [CrossRef]
44. Das, A.K.; Sutrala, A.K.; Kumari, S.; Odelu, V.; Wazid, M.; Li, X. An efficient multi-gateway-based three-factor user authentication and key agreement scheme in hierarchical wireless sensor networks. *Secur. Commun. Netw.* **2016**, *9*, 2070–2092. [CrossRef]
45. Shnayder, V.; Hempstead, M.; Chen, B.R.; Allen, G.W.; Welsh, M. Simulating the power consumption of large-scale sensor network applications. In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, Baltimore, MD, USA, 3–5 November 2004; pp. 188–200.