sensors

MDPI

Article

An Improved Lightweight User Authentication Scheme for the Internet of Medical Things

Keunok Kim [1], Jihyeon Ryu [2], Youngsook Lee [3] and Dongho Won [4,*]

[1] Department of Electrical and Computer Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si 16419, Republic of Korea
[2] Department of Computer Science and Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si 16419, Republic of Korea
[3] Department of IT Software Security, Howon University, 64 Impi-myeon, Howondae 3-gil, Gunsan-si 54058, Republic of Korea
[4] Department of Computer Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si 16419, Republic of Korea
* Correspondence: dhwon@security.re.kr

Abstract: The Internet of Medical Things (IoMT) is used in the medical ecosystem through medical IoT sensors, such as blood glucose, heart rate, temperature, and pulse sensors. To maintain a secure sensor network and a stable IoMT environment, it is important to protect the medical IoT sensors themselves and the patient medical data they collect from various security threats. Medical IoT sensors attached to the patient's body must be protected from security threats, such as being controlled by unauthorized persons or transmitting erroneous medical data. In IoMT authentication, it is necessary to be sensitive to the following attack techniques. (1) The offline password guessing attack easily predicts a healthcare administrator's password offline and allows for easy access to the healthcare worker's account. (2) Privileged-insider attacks executed through impersonation are an easy way for an attacker to gain access to a healthcare administrator's environment. Recently, previous research proposed a lightweight and anonymity preserving user authentication scheme for IoT-based healthcare. However, this scheme was vulnerable to offline password guessing, impersonation, and privileged insider attacks. These attacks expose not only the patients' medical data such as blood pressure, pulse, and body temperature but also the patients' registration number, phone number, and guardian. To overcome these weaknesses, in the present study we propose an improved lightweight user authentication scheme for the Internet of Medical Things (IoMT). In our scheme, the hash function and XOR operation are used for operation in low-spec healthcare IoT sensor. The automatic cryptographic protocol tool ProVerif confirmed the security of the proposed scheme. Finally, we show that the proposed scheme is more secure than other protocols and that it has 266.48% better performance than schemes that have been previously described in other studies.

Keywords: Internet of Medical Things (IoMT); user authentication; biometric; healthcare; healthcare IoT

Citation: Kim, K.; Ryu, J.; Lee, Y.; Won, D. An Improved Lightweight User Authentication Scheme for the Internet of Medical Things. *Sensors* **2023**, *23*, 1122. https://doi.org/10.3390/s23031122

Academic Editors: Richard Chbeir and Taoufik Yeferny

Received: 31 December 2022
Revised: 15 January 2023
Accepted: 15 January 2023
Published: 18 January 2023

1. Introduction

The Internet of Medical Things (IoMT) represents a combination of the healthcare field and the IoT ecosystem that can be used to create, collect, transmit, and analyze medical data through the connection of various healthcare IT systems, healthcare sensors, and healthcare management programs [1,2]. With the continued evolution of IoT technology and the outbreak of COVID-19, IoMT gaining increased interest as it can enable personalized medical information management, real–time health tracking and monitoring but also remote treatment [3].

However, there may be various security problems in the IoMT environment when dealing with sensitive medical information [4,5], such as the following:

1. First, if malicious cyberattacks can take control of healthcare sensors attached to a patient's body, this might not only result in inaccurate data collection but also put the patient's health at risk.
2. Second, malicious cyberattacks can expose sensitive patient data and medical information.
3. Third, since IoMT uses low–power wearable healthcare sensors, the protocol used is not lightweight, and it therefore may be difficult to operate normally or to provide real–time service due to the need for time–consuming computation.

Therefore, further develop the IoMT environment, it is crucial to maintain security for medical systems and IoT devices and to support lightweight security protocols for implementing them.

We implemented and analyzed a suitable scheme for IoMT using the following method. For safety, we not only use a simple password base but also introduce a fuzzy extractor, i.e., it is a biometric-based authentication method. We also propose a system model and an attack model to complement the weaknesses of Masud et al. [6]. We analyze the vulnerabilities of Masud et al. [6] based on the system model and attack model and propose our new IoMT scheme. We analyze the safety of the proposed scheme using a formal analysis and an informal analysis and calculate the cost of computation. Lastly, we analyze how efficient the computational cost is compared to those of other schemes.

1.1. Our Contribution

We proposed a secure and lightweight user authentication scheme for IoMT by improving on Masud et al. [6]'s scheme by addressing the possible threats involved. In summary, we make the following contributions:

1. First, to overcome offline password-guessing attacks, we added biometrics authentication methods that can only authenticate the user when an actual user is present. Further, to protect against replay attacks, we added logic to ensure that the gateway authenticates the user and the freshness of the user's message in the authentication phase. Finally, to overcome privileged insider attacks, we deleted the secret information that shared between the user and the sensor in the registration phase immediately following the registration phase.
2. We proposed a lightweight security protocol that mainly uses a hash function and XOR operation to run low-spec healthcare sensors.
3. The proposed scheme is designed to protect against various security threats such as offline password guessing attacks, privileged insider attacks, user impersonation attacks, replay attacks, and session key disclosure attacks. It also ensures user anonymity.

1.2. Organization of Our Paper

The rest of this paper is organized as follows: Section 3 presents the preliminaries about the fuzzy extractor, system model, and attack model. Section 4 presents the scheme reported by Masud et al. [6]. Section 5 demonstrates the scheme reported by Masud et al. [6]. Section 6 presents our improved scheme. Section 7 provides formal and informal security analysis. Section 8 provides a performance analysis. Section 9 provides discussion of performance. Finally, Section 10 presents our conclusion.

2. Related Work

Even until recently, most healthcare systems could only be accessed using a password. Password-based authentication [7,8] is the most popular method for user authentication. However, it is unfortunately not suitable for use in a sensitive system that requires strong security because it contains various security threats. For example, password-based authentication can lead to unintentional password sharing when someone looks over the user's shoulder and sees the user's password. Anyone who knows this password can access the system on behalf of the user. Moreover, a password guessing attack is possible, in which the attacker guesses the user's password and attempts authentication until finding success.

Password-based authentication schemes can also be exposed to various security threats, so to overcome the security threats of password-based authentication, two–factor authentication using a smart card has been introduced. In 2012, Wu et al. [9] proposed a secure authentication scheme for TMIS using a smart card and a password. However, Debiao et al. [10] pointed out that this scheme was vulnerable to impersonation attacks and insider attacks. Meanwhile, Wei et al. [11] pointed out that the scheme was vulnerable to offline password guessing attacks. To overcome these problems, Debiao et al. [10] proposed a more secure authentication scheme using a smart card and a password. However, Wei et al. [11] pointed out that this scheme also was vulnerable to offline password guessing attacks if the user were to lose his or her smart card. Consequently, a three-factor authentication using a password, a smart card, and biometrics, i.e., fingerprint and face recognition, has been introduced to achieve a higher level of security [12–15]. Wu et al. [12] proposed an improved and provably secure three-factor user authentication scheme for wireless sensor networks. However, Ryu et al. [13] pointed out that this scheme was vulnerable to user impersonation attacks and that it could not preserve the user's anonymity could not be preserved. A summary of each scheme's weaknesses is presented in Table 1.

Table 1. Summary of related works.

Author	Proposed Scheme	Weakness
Wu et al. [9]	For TMIS using a smart card	Impersonation, insider, offline password guessing attacks
Debiao et al. [10]	Using a smart card and a password	Offline password guessing attacks
Wu et al. [12]	Secure three-factor scheme for wireless sensor networks	User impersonation attacks and no user anonymity
Masud et al. [6]	For IoT-based healthcare	Offline-password, replay, privileged insider attacks

Recently, Masud et al. [6] proposed a lightweight and anonymity preserving user authentication scheme for IoT–based healthcare. In the present study, we specifically focused on the use of lightweight protocols to support resource–constrained devices. However, we found various security threats, since this paper only provides user authentication using a password to remain lightweight. While it is important to keep in mind that supporting lightweight security protocols in an IoMT environment is one of the most important considerations, maintaining security is the most foundational requirement. One of the fatal security threats to which Masud et al. [6]'s scheme is vulnerable is an offline-password attack. If an attacker steals a valid authentication message in any authentication phase that is communicated via a public channel, the attacker can guess the user's valid password while offline. Masud et al. [6]'s scheme is also vulnerable to replay attacks and privileged insider attacks. Through such attack, an attacker could log in by stealing the user's password and disguising themselves as the user. An attacker could also enter information on behalf of the user or have access to the user's information as the user.

Although Masud et al. [6]'s scheme exhibited better performance to maintain security for IoMT, it still has a security challenge. Therefore, we proposed an improved lightweight user authentication scheme for IoMT that improved upon Masud et al. [6]'s scheme and reduces reduced the computational cost compared to related studies.

3. Preliminaries

In this section, we introduce the fuzzy extractor, the system model which we made, and the attack model. The details are as follows:

3.1. Fuzzy Extractor

Biometric information is the best way to authenticate and verify users [16–18]. In 2004, Dodis et al. [19] proposed the Fuzzy Extractor to obtain a unique bit string extracted from the biometric template. A fuzzy extractor is a tuple $(M, m_e, l, \tau, \epsilon)$ that has two algorithms Gen and Rep, which are expressed as follows [12]:

$$Gen(w_i) = (R_i, P_{bi}) \quad (1)$$

The above result means that the fuzzy extractor can generate the secret string R_i and P_{bi}. w_i is a U_i's original collected biometric data.

$$Rep(w_i^{'}, P_{bi}) = R_i \tag{2}$$

The above result means that the fuzzy extractor can recover the secret string R_i that is generated by the *Gen* algorithm.

3.2. System Model

Figure 1 shows the system model of the proposed scheme. In our scheme, there are three entities: the User, Medical Service Gateway (Gateway), and Healthcare IoT Sensor Node (Sensor Node).

1. Medical Service Gateway (Gateway): Since the user and the sensor node do not communicate with each other directly, the gateway is responsible for authentication and passing the communication between the user and the sensor node.
2. Healthcare IoT Sensor Node (Sensor Node): The Healthcare IoT Sensor Node is attached to the patient's body and collects the patient's medical data. The Healthcare IoT Sensor Node is connected to the healthcare network and transmits the patient's medical data to the user through a Gateway.
3. Doctor (User): The user is a doctor who can access the patient's medical information that has been collected from the sensor node to inform the patient's treatment.

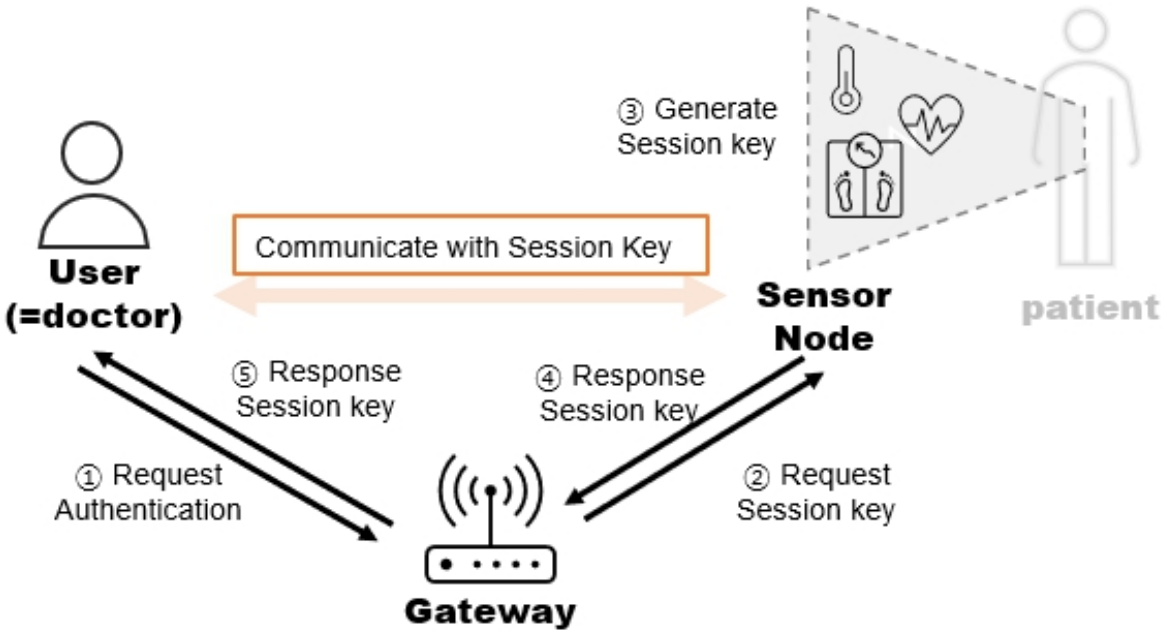

Figure 1. The system model of our scheme in IoMT.

3.3. Attack Model

For security analysis in our scheme, we consider the following attack model [20–22]:

1. The attacker can extract the data in the device that stores some security parameters.
2. The attacker can access the public communication channel, at which point the attacker can interrupt, return, amend and eliminate or transmit the message.
3. The attacker can calculate the identity and password in polynomial time.

4. Review of the Scheme Presented by Masud et al. [6]

In this section, we briefly review Masud et al.'s scheme [6] that only uses password protection to only let legitimate users access the IoT sensor node to obtain the patient's health

information. The notations and Masud et al. [6]'s scheme are described in Tables 2 and 3, respectively.

Table 2. Notations.

Notion	Description
U_iM_*, GM_*, SN_iM_*	*-th message of i-th user, gateway, and j-th sensor node, respectively
ID_i, SID_j	Identity of i-th user, j-th sensor node, respectively
PW_i	Password of i-th user
B_i	Biometric Information of i-th user
r_U^*, r_{GW}^*, r_{SN}^*	$*$-th random number generated by user, gateway, and sensor node, respectively
ts_U	Timestamp of user
S_i^1	Secret information between user and gateway
S_i^2	Secret information between user and sensor node
D_{ID}, S_{ID}	Identity of user, sensor node
PW_D	Device password set by doctor
R_{SG}, R_{SN}	Random secret generated by gateway, sensor node, respectively
N_D, N_G, N_S	Nonce generated by user's device, gateway, sensor node, respectively
K_{GW}	Secret key of gateway
$h(.)$	Hash function
$\|$	Concatenation operator
$\oplus$	Bit wise XOR
SK	Session key

Masud et al. [6]'s scheme consists of three phases: the User Registration Phase, the Sensor Node Registration Phase, and the Mutual Authentication and Key Agreement Phase.

4.1. User Registration Phase

1. The user enters and transmits D_{ID} and PW_D through the secured channel to the gateway.
2. The gateway generates R_{SG}^1 and computes $D_{TID} = R_{SG}^1 \oplus D_{ID}$ and $\alpha = (D_{ID} \oplus R_{SG}^1) \oplus PW_D$. The gateway stores D_{ID}, PW_D, R_{SG}^1 and D_{TID}. Finally, the gateway transmits α through the secured channel to the user.
3. The user derives $R_{SG}^{1*} = (\alpha \oplus PW_D) \oplus D_{ID}$ and computes $D_{TID} = R_{SG}^{1*} \oplus D_{ID}$ and $\beta = h(PW_D \parallel R_{SG}^{1*}) \oplus D_{TID}$. The user then stores R_{SG}^{1*}, D_{TID}, and β.

4.2. Sensor Node Registration Phase

1. The sensor node generates R_{SN}^1 and then transmits R_{SN}^1 and S_{ID} through the secured channel to the gateway.
2. The gateway generates R_{SG}^2 and computes $\delta = (S_{ID} \oplus R_{SG}^2) \oplus R_{SN}^1$ and $S_{TID} = R_{SG}^2 \oplus S_{ID}$. The gateway then stores S_{ID}, R_{SN}^1, R_{SG}^2, and S_{TID}. Finally, the gateway transmits δ through the secured channel to the sensor.
3. The sensor node computes $R_{SG}^{2*} = (\delta \oplus R_{SN}^1) \oplus S_{ID}$ and $S_{TID} = R_{SN}^{2*} \oplus S_{ID}$. The sensor node then stores R_{SN}^1, R_{SG}^{2*}, S_{TID}.

Table 3. Masud et al.'s [6] scheme.

User	**Gateway**	**Sensor Node**
{User Registration Phase}		
Enter: D_{ID}, PW_D		
Transmit: D_{ID}, $PW_D \longrightarrow$	Generate: R^1_{SG}	
	Compute: $D_{TID} = R^1_{SG} \oplus D_{ID}$	
	$\alpha = (D_{ID} \oplus R^1_{SG}) \oplus PW_D$	
	Store: D_{ID}, PW_D, R^1_{SG}, D_{TID}	
Derives: $R^{1*}_{SG} = (\alpha \oplus PW_D) \oplus D_{ID}$	$\longleftarrow$ Transmit: α	
Compute: $D_{TID} = R^{1*}_{SG} \oplus D_{ID}$,		
$\beta = h(PW_D \parallel R^{1*}_{SG}) \oplus D_{TID}$		
Store: R^{1*}_{SG}, D_{TID}, β		
{Sensor Node Registration Phase}		
		Generate: R^1_{SN}
	Generate: R^2_{SG}	$\longleftarrow$ Transmit: R^1_{SN}, S_{ID}
	Compute: $\delta = (S_{ID} \oplus R^2_{SG}) \oplus R^1_{SN}$,	
	$S_{TID} = R^2_{SG} \oplus S_{ID}$	
	Store: S_{ID}, R^1_{SN}, R^2_{SG}, S_{TID}	
	Transmit: $\delta \longrightarrow$	Compute: $R^{2*}_{SG} = (\delta \oplus R^1_{SN}) \oplus S_{ID}$,
		$S_{TID} = R^{2*}_{SN} \oplus S_{ID}$
		Store: R^1_{SN}, R^{2*}_{SG}, S_{TID}
{Mutual Authentication and Key agreement}		
(1)		
Enter: PW_D	**(2)**	
Compute: $Q = h(PW_D \parallel R^{1*}_{SG}) \oplus D_{TID}$	Retrieves: $N^1_D = N^{1*}_D \oplus PW_D$	
Verify: Q = ? β	Verify: N^1_D's freshness	
Generate: N^1_D	Check: D_{TID}, S_{TID} are stored	
Compute: $N^{1*}_D = N^1_D \oplus PW_D$,	Compute: $\lambda^* = h(R^1_{SG} \parallel PW_D)$	
$\lambda = h(R^{1*}_{SG} \parallel PW_D)$	Verify: λ^* = ? λ	
Transmit: $N^{1*}_D, D_{TID}, \lambda, S_{TID} \longrightarrow$	Generate: N^1_G, SK, R^3_{SG}	
	Compute: $G^1_W = N^1_G \oplus S_{TID}$,	**(3)**
	$G^2_W = h(R^1_{SN} \parallel R^2_{SG})$,	Retrieve: $N^1_G = G^1_W \oplus S_{TID}$
	$SK_S = (SK \oplus R^1_{SN}) \oplus N^1_G$,	Verify: N^1_G's freshness
	$G^3_W = R^3_{SG} \oplus R^1_{SN}$	Compute: $S^1_N = h(R^1_{SN} \parallel R^{2*}_{SG})$
	Store: G^3_W	Verify: S^1_N = ? G^2_W
	Transmit: G^1_W, G^2_W, D_{TID},	Retrieve: $SK = (SK_S \oplus R^1_{SN}) \oplus N^1_G$
	SK_S, $G^3_W \longrightarrow$	Generate: N^1_S, R^2_{SN}
		Retrieve: $R^3_{SG} = G^3_W \oplus R^1_{SN}$
	(4)	Compute: $S^2_N = N^1_S \oplus S_{TID}$,
	Retrieve: $N^1_S = S^2_N \oplus S_{TID}$,	$S^3_N = h(R^{2*}_{SG} \parallel R^1_{SN} \parallel SK)$,
	$R^2_{SN} = S^4_N \oplus R^2_{SG}$	$S^4_N = R^{2*}_{SG} \oplus R^2_{SN}$,
	Verify: N^1_S's freshness	$S^{new}_{TID} = R^3_{SG} \oplus S_{ID}$
	Compute: $G^4_W = h(R^2_{SG} \parallel R^1_{SN} \parallel SK)$,	Store: R^2_{SN}, R^3_{SG}, S^{new}_{TID}, S^3_N
	$S^{new}_{TID} = S^3_{SG} \oplus S_{ID}$	$\longleftarrow$ Transmit: S^2_N, S^3_N, S^4_N
	Verify: G^4_W = ? S^3_N	
(5)	Store: R^2_{SN}, R^3_{SG}, S^{new}_{TID}	
Retrieve: $N^2_G = \mu \oplus D_{ID}$	Generate: N^2_G, R^4_{SG}	
Verify: N^2_G's freshness	Compute: $\mu = D_{ID} \oplus N^2_G$,	
Compute: $\phi = h(D_{ID} \parallel PW_D \parallel SK \parallel N^2_G)$	$SK_U = (SK \oplus PW_D) \oplus N^2_G$,	
Verify: ϕ = ? η	$\eta = h(D_{ID} \parallel PW_D \parallel SK \parallel N^2_G)$,	
Retrieve: $SK = (SK_U \oplus N^2_G) \oplus PW_D$,	$G^5_W = R^4_{SG} \oplus PW_D$,	
$R^4_{SG} = G^5_W \oplus PW_D$	$D^{new}_{TID} = R^4_{SG} \oplus D_{ID}$	
Compute: $D^{new}_{TID} = R^4_{SG} \oplus D_{ID}$	Store:R^4_{SG}, D^{new}_{TID}	
Store: R^4_{SG}, D^{new}_{TID}	$\longleftarrow$ Transmit:μ, SK_u, η, and G^5_W	

4.3. Mutual Authentication and Key Agreement Phase

1. The user enters PW_D and then computes $Q = h(PW_D \parallel R^{1*}_{SG}) \oplus D_{TID}$, where R^{1*}_{SG} and D_{TID}. If Q and β are not equal, the procedure is stopped; if they are equal, the user generates N^1_D and computes $N^{1*}_D = N^1_D \oplus PW_D$ and $\lambda = h(R^{1*}_{SG} \parallel PW_D)$. Finally, the user transmits $N^{1*}_D, D_{TID}, \lambda$, and S_{TID} to the gateway.
2. The gateway retrieves $N^1_D = N^{1*}_D \oplus PW_D$, after which the gateway checks the freshness of N^1_D. If N^1_D is not fresh, the procedure is stopped; else the gateway compares D_{TID} and S_{TID} with the received values. If the values are not identical, the procedure is

stopped; else the gateway computes $\lambda^* = h(R^1_{SG} \parallel PW_D)$. If λ^* and λ are equal, the user authentication is successful. If they are not equal, the procedure is stopped. To share SK with the sensor node, the gateway generates N^1_G, SK, and R^3_{SG}. The gateway computes $G^1_W = N^1_G \oplus S_{TID}$, $G^2_W = h(R^1_{SN} \parallel R^2_{SG})$, $SK_S = (SK \oplus R^1_{SN}) \oplus N^1_G$ and $G^3_W = R^3_{SG} \oplus R^1_{SN}$. Finally, the gateway stores G^3_W and transmits G^1_W, G^2_W, D_{TID}, SK_S, and G^3_W through the public channel to the sensor node.

3. The sensor node retrieves $N^1_G = G^1_W \oplus S_{TID}$ and then checks the freshness of N^1_G. If N^1_G is not fresh, the procedure is stopped; else the sensor node computes $S^1_N = h(R^1_{SN} \parallel R^{2*}_{SG})$. If S^1_N and G^2_W are equal, the gateway authentication is successful. Then, the sensor node retrieves $SK = (SK_S \oplus R^1_{SN}) \oplus N^1_G$ and the sensor node generates N^1_S and R^2_{SN}. The sensor node then computes $S^2_N = N^1_S \oplus S_{TID}$, $S^3_N = h(R^{2*}_{SG} \parallel R^1_{SN} \parallel SK)$, and $S^4_N = R^{2*}_{SG} \oplus R^2_{SN}$. Moreover, the sensor node retrieves $R^3_{SG} = G^3_W \oplus R^1_{SN}$ and derives $S^{new}_{TID} = R^3_{SG} \oplus S_{ID}$. The sensor node stores R^2_{SN}, R^3_{SG}, S^{new}_{TID}, and S^3_N. Finally, the sensor node transmits S^2_N, S^3_N, and S^4_N to the gateway.
4. The gateway retrieves $N^1_S = S^2_N \oplus S_{TID}$ and $R^2_{SN} = S^4_N \oplus R^2_{SG}$. The gateway then checks the freshness of N^1_S. If N^1_S is not fresh, the procedure is stopped; else the gateway computes $G^4_W = h(R^2_{SG} \parallel R^1_{SN} \parallel SK)$ and $S^{new}_{TID} = S^3_{SG} \oplus S_{ID}$. If G^4_W and S^3_N are equal, the mutual authentication between the sensor node and the gateway is successful. Subsequently, the gateway stores R^2_{SN}, R^3_{SG}, and S^{new}_{TID}. To share SK with the user, the gateway generates N^2_G and R^4_{SG}. The gateway then computes $\mu = D_{ID} \oplus N^2_G$, $SK_U = (SK \oplus PW_D) \oplus N^2_G$, $\eta = h(D_{ID} \parallel PW_D \parallel SK \parallel N^2_G)$, $G^5_W = R^4_{SG} \oplus PW_D$ and $D^{new}_{TID} = R^4_{SG} \oplus D_{ID}$. Finally the gateway stores R^4_{SG} and D^{new}_{TID} and ultimately transmits μ, SK_u, η, and G^5_W to the user.
5. The user retrieves $N^2_G = \mu \oplus D_{ID}$. The user then checks the freshness of N^2_G. If N^2_G is not fresh, the procedure is stopped; else the user computes $\phi = h(D_{ID} \parallel PW_D \parallel SK \parallel N^2_G)$. If ϕ and η are equal, the mutual authentication between the gateway and the user is successful. Then the user retrieves $SK = (SK_U \oplus N^2_G) \oplus PW_D$ and $R^4_{SG} = G^5_W \oplus PW_D$. The user computes $D^{new}_{TID} = R^4_{SG} \oplus D_{ID}$ and stores R^4_{SG} and D^{new}_{TID}.

5. Weaknesses of Masud et al. [6]'s Scheme

These weaknesses are listed under the assumption that the attacker has recorded the message $(N^{1*}_D, D_{TID}, \lambda, S_{TID})$ from a successful mutual authentication and key agreement of the user A.

5.1. Offline Password Guessing Attack

In an offline password guessing attack, the attacker is never actually attempting to login to the gateway server. Suppose the attacker steals the device of user A and obtains R^{1*}_{SG} from the device. Then, the attacker repeatedly guesses a password PW^*_D and computes $\lambda^* = h(R^{1*}_{SG} \parallel PW^*_D)$ offline. If λ^* is equal to λ, the attacker can obtain the correct password PW_D. Until the attacker determines a valid user's password PW_D, the gateway does not notice this attack at all because the attacker does not try to login.

Then, in the Mutual Authentication and Key Agreement Phase, the attacker retrieves $D_{ID} = R^{1*}_{SG} \oplus D_{TID}$, $N^2_G = \mu \oplus D_{ID}$ and $SK = (SK_U \oplus N^2_G) \oplus PW_D$. Eventually, the attacker can obtain an SK that can be used to access the resource of the gateway and the sensor node.

5.2. Privileged Insider Attack

If a privileged insider of the gateway has obtained the user A's password PW_D, D_{ID} and R^1_{SG} from the gateway's database, he is trying to impersonate user A. In the Mutual Authentication and Key Agreement Phase, the privileged insider retrieves $N^2_G = \mu \oplus D_{ID}$ and $SK = (SK_U \oplus N^2_G) \oplus PW_D$. Eventually, the privileged insider can obtain an SK that can access the resource of the gateway and the sensor node.

5.3. Replay Attack

Masud et al. [6] claim their scheme is safe from replay attacks because the gateway checks the freshness of nonce $N_D^{1*} = N_D \oplus PW_D$ and that the nonce cannot be modified since it is secretly enclosed in the password PW_D. However, suppose the attacker generates the nonce N_A and then transmits $(N_A, D_{TID}, \lambda, S_{TID})$ instead of $(N_D, D_{TID}, \lambda, S_{TID})$. Upon receiving $(N_A, D_{TID}, \lambda, S_{TID})$, the gateway retrieves $N_D^{1*} = N_A \oplus PW_D$ and then the gateway checks the freshness of N_D^{1*}. However, since the retrieved N_D^{1*} is a random number if only freshness is guaranteed, the gateway cannot confirm whether N_D^{1*} is valid. Briefly, if the attacker can make a nonce that can guarantee freshness, Masud et al. [6]'s scheme cannot resist replay attacks. Further, if freshness is proven since the new identities of user and device D_{TID}^{new}, S_{TID}^{new} are changed, valid users who do not know the changed identity D_{TID}^{new} will no longer be able to authenticate themselves after the replay attack.

6. Proposed Scheme

In this section, we propose a three-factor mutual authentication scheme for the Internet of Medical Things (IoMT) that is intended to overcome the weaknesses of the scheme reported by Masud and colleagues. The proposed scheme only consists of three phases: user registration, sensor node registration, and authentication and key distribution. The proposed scheme is described in Table 4.

6.1. User Registration Phase

1. The user enters ID_i and PW_D and generates a random number r_U^1. The user imprints B_i on a device for biometric collection and computes $Gen(B_i) = (R_i, R_{bi})$ and $HPW_i = h(PW_i \parallel R_i \parallel r_U^1)$. For registration, the user transmits ID_i through a secured channel to the gateway.
2. The gateway generates random numbers r_{GW}^1, r_{GW}^2 and r_{GW}^3 and computes $TID_i = h(ID_i \parallel r_{GW}^1 \parallel K_{GW})$, $S_i^1 = h(ID_i \parallel r_{GW}^2 \parallel K_{GW})$ and $S_i^2 = h(ID_i \parallel r_{GW}^3 \parallel K_{GW})$. The gateway stores ID_i, TID_i, S_i^1 and S_i^2. S_i^2 is temporarily stored by the gateway until the sensor node registration phase, is transferred from the gateway to the sensor node during the sensor node registration phase, and is then deleted from the gateway. Finally, the gateway transmits TID_i, S_i^1, and S_i^2 through a secured channel to the user.
3. The user computes $U_iM_1 = TID_i \oplus HPW_i$, $U_iM_2 = S_i^1 \oplus HPW_i$ and $U_iM_3 = S_i^2 \oplus HPW_i$, $U_iM_4 = h(PW_i \parallel R_i \parallel ID_i) \oplus r_U^1$ and $U_iM_5 = h(r_U^1 \parallel TID_i \parallel S_i^1 \parallel S_i^2)$ and stores U_iM_1, U_iM_2, U_iM_3, U_iM_4, and U_iM_5.

6.2. Sensor Node Registration Phase

1. For registration, the sensor node transmits SID_j through a secured channel to the gateway.
2. The gateway generates a random number r_{GW}^4 and computes $TSID_j = h(SID_j \parallel r_{GW}^4 \parallel K_{GW})$ and stores SID_j, $TSID_j$. Finally, the gateway transmits $TSID_j$, TID_i, and S_i^2 through a secured channel to the sensor node and deletes S_i^2.
3. The sensor node stores SID_j, $TSID_j$, TID_i, and S_i^2.

Table 4. The proposed scheme.

User(U_i)	**Gateway**	**Sensor Node(SN_j)**
{User Registration}		
(1) Enter: ID_i, PW_i, B_i		
Generate: r_U^1	**(2)** Generate: r_{GW}^1, r_{GW}^2, r_{GW}^3	
Compute: $Gen(B_i) = (R_i, R_{bi})$,	Compute:	
$HPW_i = h(PW_i \parallel R_i \parallel r_U^1)$	$TID_i = h(ID_i \parallel r_{GW}^1 \parallel K_{GW})$,	
Transmit: $ID_i \longrightarrow$	$S_i^1 = h(ID_i \parallel r_{GW}^2 \parallel K_{GW})$,	
	$S_i^2 = h(ID_i \parallel r_{GW}^3 \parallel K_{GW})$	
(3) Compute:	Store: ID_i, TID_i, S_i^1, S_i^2	
$U_iM_1 = TID_i \oplus HPW_i$,	$\longleftarrow$ Transmit: TID_i, S_i^1, S_i^2	
$U_iM_2 = S_i^1 \oplus HPW_i$		
$U_iM_3 = S_i^2 \oplus HPW_i$		
$U_iM_4 = h(PW_i \parallel R_i \parallel ID_i) \oplus r_U^1$		
$U_iM_5 = h(r_U^1 \parallel TID_i \parallel S_i^1 \parallel S_i^2)$		
Store: U_iM_1, U_iM_2, U_iM_3, U_iM_4, U_iM_5		
{Sensor Node Registration}		
	(2) Generate: r_{GW}^4	$\longleftarrow$ **(1)** Transmit: SID_j
	Compute:	
	$TSID_j = h(SID_j \parallel r_{GW}^4 \parallel K_{GW})$	
	Store: SID_j, $TSID_j$	
	Delete: S_i^2	
	Transmit: $TSID_j$, TID_i, $S_i^2 \longrightarrow$	**(3)** Store: SID_j, $TSID_j$, TID_i, S_i^2
{Authentication and Key distribution}		
(1) Verify Password		
Enter: ID_i, PW_i, B_i		
Compute: $R_i = Rep(B_i, R_{bi})$,	**(2)** Verify the user	
$r_U^1 = h(PW_i \parallel R_i \parallel ID_i) \oplus U_iM_4$,	Retrieve: $r_U^2 = U_iM_6 \oplus S_i^1$	
$HPW_i = h(PW_i \parallel R_i \parallel r_U^1)$	Compute:	**(3)** Verify the gateway
$TID_i = U_iM_1 \oplus HPW_i$,	$U_iM_8^* = h(r_U^2 \parallel TID_i \parallel ts_U)$	Retrieve: $r_{GW}^5 = GM_1 \oplus TSID_j$
$S_i^1 = U_iM_2 \oplus HPW_i$,	Verify: r_U^2's freshness, ts_U	$TID_i = GM_2 \oplus r_{GW}^5$
$S_i^2 = U_iM_3 \oplus HPW_i$,	U_iM_8= ?$U_iM_8^*$	$TID_i^{new} = GM_3 \oplus r_{GW}^5$
$U_iM_5^* = h(r_U^1 \parallel TID_i \parallel S_i^1 \parallel S_i^2)$		$r_U^3 = U_iM_7 \oplus S_i^2$
Verify: U_iM_5 = ? $U_iM_5^*$	Generate: r_{GW}^5	Compute:
	$TID_i^{new} = h(ID_i \parallel r_{GW}^5 \parallel K_{GW})$,	GM_4^*
Generate: r_U^2, r_U^3, ts_U	$GM_1 = TSID_j \oplus r_{GW}^5$,	$= h(r_{GW}^5 \parallel TSID_j \parallel TID_i \parallel TID_i^{new})$
Compute:	$GM_2 = TID_i \oplus r_{GW}^5$,	Verify: GM_4= ?GM_4^*
$U_iM_6 = r_U^2 \oplus S_i^1$,	$GM_3 = TID_i^{new} \oplus r_{GW}^5$	
$U_iM_7 = r_U^3 \oplus S_i^2$	GM_4	
$U_iM_8 = h(r_U^2 \parallel TID_i \parallel ts_U)$	$= h(r_{GW}^5 \parallel TSID_j \parallel TID_i \parallel TID_i^{new})$	
Transmit:	Transmit: GM_1, GM_2, GM_3	
U_iM_6, U_iM_7, U_iM_8, TID_i, $ts_U \longrightarrow$	GM_4, $U_iM_7 \longrightarrow$	Generate: SK
		Compute:
	(4) Verify the sensor node	$SN_jM_1 = SK \oplus S_i^2$
(5) Verify the gateway	Compute:	$SN_jM_2 = h(SK \parallel TID_i^{new} \parallel r_U^3)$
Compute:	$SN_jM_3^*$	SN_jM_3
$TID_i^{new} = GM_6 \oplus S_i^1$,	$= h(SN_jM_1 \parallel SN_jM_2 \parallel TSID_j)$	$= h(SN_jM_1 \parallel SN_jM_2 \parallel TSID_j)$
$GM_5^* = h(TID_i \parallel TID_i^{new})$	Verify: SN_jM_3 = ? $SN_jM_3^*$	Replace: $TID_i \leftarrow TID_i^{new}$
$U_iM_1^{new} = TID_i^{new} \oplus HPW_i$		Transmit: SN_jM_1, SN_jM_2
Verify: GM_5 = ? GM_5^*	Compute:	$\longleftarrow SN_jM_3$
Replace: $U_iM_1 \leftarrow U_iM_1^{new}$	$GM_5 = h(TID_i \parallel TID_i^{new})$,	
	$GM_6 = TID_i^{new} \oplus S_i^1$	
Retrieve: $SK = SN_jM_1 \oplus S_i^2$	Replace: $TID_i \leftarrow TID_i^{new}$	
Compute:	Transmit: GM_5, GM_6, SN_jM_1,	
$SN_jM_2^* = h(SK \parallel TID_i^{new} \parallel r_U^3)$	$\longleftarrow SN_jM_2$	
Verify: SN_jM_2 = ? $SN_jM_2^*$		

6.3. Mutual Authentication and Key Distribution Phase

1. The user enters ID_i and PW_i and imprints B_i on a device for biometric collection and computes $R_i = Rep(B_i, R_{bi})$, $r_U^1 = h(PW_i \parallel R_i \parallel ID_i) \oplus U_iM_4$, $HPW_i = h(PW_i \parallel R_i \parallel r_U^1)$, $TID_i = U_iM_1 \oplus HPW_i$, $S_i^1 = U_iM_2 \oplus HPW_i$, $S_i^2 = U_iM_3 \oplus HPW_i$ and $U_iM_5^* = h(r_U^1 \parallel TID_i \parallel S_i^1 \parallel S_i^2)$. To check the user's password, the user checks if U_iM_5 = ? $U_iM_5^*$. If the equation is equal, the password check is passed; if not, the procedure is stopped. To generate an authentication message, the user generates

r_U^2, r_U^3 and ts_U and computes $U_iM_6 = r_U^2 \oplus S_i^1$, $U_iM_7 = r_U^3 \oplus S_i^2$ and $U_iM_8 = h(r_U^2 \parallel TID_i \parallel ts_U)$. Finally the user transmits U_iM_6, U_iM_7, U_iM_8, TID_i and ts_U through a public channel to the gateway.

2. To authenticate the user, the gateway retrieves $r_U^2 = U_iM_6 \oplus S_i^1$ and computes $U_iM_8^* = h(r_U^2 \parallel TID_i \parallel ts_U)$. The gateway checks r_U^2's freshness and if $U_iM_8 = ?\ U_iM_8^*$. The gateway also checks for whether or not ts_U is a valid range. If $U_iM_8^*$ and ts_U are valid, the user verification is passed; if not, the procedure is stopped. To generate the authentication message, the gateway generates r_{GW}^5 and computes $TID_i^{new} = h(ID_i \parallel r_{GW}^5 \parallel K_{GW})$, $GM_1 = TSID_j \oplus r_{GW}^5$, $GM_2 = TID_i \oplus r_{GW}^5$, $GM_3 = TID_i^{new} \oplus r_{GW}^5$ and $GM_4 = h(r_{GW}^5 \parallel TSID_j \parallel TID_i \parallel TID_i^{new})$. Finally the gateway transmits GM_1, GM_2, GM_3, GM_4, and U_iM_7 through a public channel to the sensor node.
3. To authenticate the gateway, the sensor node retrieves $r_{GW}^5 = GM_1 \oplus TSID_j$, $TID_i = GM_2 \oplus r_{GW}^5$, $TID_i^{new} = GM_3 \oplus r_{GW}^5$ and $r_U^3 = U_iM_7 \oplus S_i^2$ then computes $GM_4^* = h(r_{GW}^5 \parallel TSID_j \parallel TID_i \parallel TID_i^{new})$. The sensor node checks $GM_4 = ?\ GM_4^*$. If the equation is equal, the gateway verification is passed; if not, the procedure is stopped. To generate the session key SK, the sensor node generates SK and computes $SN_jM_1 = SK \oplus S_i^2$, $SN_jM_2 = h(SK \parallel TID_i^{new} \parallel r_U^3)$ and $SN_jM_3 = h(SN_jM_1 \parallel SN_jM_2 \parallel TSID_j)$. The sensor node replaces TID_i by TID_i^{new}. Finally, the sensor node transmits SN_jM_1, SN_jM_2 and SN_jM_3 through a public channel to the gateway.
4. To authenticate the sensor node, the gateway computes $SN_jM_3^* = h(SN_jM_1 \parallel SN_jM_2 \parallel TSID_j)$ and checks if $SN_jM_3 = ?\ SN_jM_3^*$. If the equation is equal, the sensor node verification is passed. To generates an authentication message, the gateway computes $GM_5 = h(TID_i \parallel TID_i^{new})$ and $GM_6 = TID_i^{new} \oplus S_i^1$ and replaces TID_i by TID_i^{new}. Finally, the gateway transmits GM_5, GM_6, SN_jM_1 and SN_jM_2 through a public channel to the user.
5. To authenticate the gateway, the user computes $TID_i^{new} = GM_6 \oplus S_i^1$, $GM_5^* = h(TID_i \parallel TID_i^{new})$ and $U_iM_1^{new} = TID_i^{new} \oplus HPW_i$. The user checks if $GM_5 = ?\ GM_5^*$. If the equation is equal, the gateway verification is passed; if not, the procedure is stopped. To obtain the session key SK, the user retrieves $SK = SN_jM_1 \oplus S_i^2$ and computes $SN_jM_2^* = h(SK \parallel TID_i^{new} \parallel r_U^3)$. The user checks $SN_jM_2 = ?\ SN_jM_2^*$. If the equation is equal, the user obtains the valid session key SK; if not, the procedure is stopped.

7. Security Analysis of the Proposed Scheme

In this section, we demonstrate formal and informal security analysis. We use the security verification tool ProVerif to demonstrate that the proposed scheme can satisfy security and authentication features. As an informal security analysis, we show how our proposed scheme meets the security requirements for an IoMT sensor protocol.

7.1. Formal Security Analysis

In this section, the ProVerif tool [23] is used to evaluate the security of the proposed protocol. ProVerif tool is an automatic cryptographic protocol verifier that was developed by Bruno Blanchet [13]. Several studies have used this tool to demonstrate the safety of their protocols [24,25].

We use two types, and four channels in total. Private channel1 and Private channel2 transmit sensitive data between the user and the gateway and between the gateway and the sensor node, respectively. Public channel1 and Public channel2 transmit general data between the user and the gateway and between the gateway and the sensor node, respectively. Table 5 presents the definitions of the channels, variables, and other related parameters. The processes performed by the user, the gateway, and the sensor node are presented in Tables 6–8, respectively. Lastly, the queries and main process are detailed in Table 9.

The results of our proposed scheme are presented in Table 10. It can be seen that the proposed protocol kept the session key *SK* safe from the attacker.

Table 5. Definitions of channels, variables and other related parameters.

```
(*—-channels—-*)
free privateChannel1:channel [private].
free privateChannel2:channel [private].
free publicChannel1:channel.
free publicChannel2:channel.
(*—-constants—-*)
free Ri:bitstring [private].
free PWi:bitstring [private].
free IDi:bitstring [private].
free kgw:bitstring [private].
free IDg:bitstring.
free SIDj:bitstring.
(*—-shared key—-*)
free SK:bitstring [private].
(*—-functions—-*)
fun xor(bitstring, bitstring):bitstring.
fun concat(bitstring, bitstring):bitstring.
fun h(bitstring):bitstring.
(*—-events—-*)
event startUi(bitstring).
event endUi(bitstring).
event startGW(bitstring).
event endGW(bitstring).
event startSNj(bitstring).
event endSNj(bitstring).
```

Table 6. User's process.

```
(*—-Ui process—-*)
let Ui =
new ru1:bitstring;
let HPWi = h(concat(concat(PWi, Ri), ru1)) in
out(privateChannel1,(IDi));
in(privateChannel1, (XTIDi:bitstring, XSi1:bitstring, XSi2: bitstring));
let UiM1= xor(XTIDi, HPWi) in
let UiM2= xor(XSi1, HPWi) in
let UiM3= xor(XSi2, HPWi) in
let UiM4 = xor(h(concat(concat(PWi, Ri), IDi)), ru1) in
let UiM5 = h(concat(concat(concat(ru1, XTIDi), XSi1), XSi2)) in
event startUi(IDi);
let ru1 = xor(h(concat(concat(PWi, Ri), IDi)), UiM4) in
let HPWi = h(concat(concat(PWi, Ri), ru1)) in
let XTIDi = xor(UiM1, HPWi) in
let XSi1 = xor(UiM2, HPWi) in
let XSi2 = xor(UiM3, HPWi) in
if h(concat(concat(concat(ru1, XTIDi), XSi1), XSi2)) = UiM5 then
new ru2:bitstring;
new ru3:bitstring;
new tsU:bitstring;
let UiM6 = xor(ru2, XSi1) in
let UiM7 = xor(ru3, XSi2) in
let UiM8 = h(concat(ru2, concat(XTIDi, tsU))) in
out(publicChannel1, (UiM6, UiM7, UiM8, XTIDi, tsU));
in(publicChannel1, (XGM5:bitstring, XGM6:bitstring, XXSNjM1:bitstring, XXSNjM2:bitstring));
let XTIDinew = xor(XGM6, XSi1) in
let UiM1new = xor(XTIDinew, HPWi) in
if h(concat(XTIDi, XTIDinew)) = XGM5 then
let UiM1 = UiM1new in
let SK = xor(XXSNjM1, XSi2) in
if(concat(concat(SK, XTIDinew), ru3)) = XXSNjM2 then
event endUi(IDi).
```

Table 7. Gateway's process.

```
(*—-GW process—-*)
let GW =
in(privateChannel1, XIDi:bitstring);
new rgw1:bitstring;
new rgw2:bitstring;
new rgw3:bitstring;
let TIDi = h(concat(concat(IDi, rgw1), kgw)) in
let Si1 = h(concat(concat(IDi, rgw2), kgw)) in
let Si2 =h(concat(concat(IDi, rgw1), kgw)) in
out(privateChannel1, (TIDi, Si1, Si2));
in(privateChannel2, XSIDj:bitstring);
new rgw4:bitstring;
let TSIDj = h(concat(concat(XSIDj, rgw4), kgw)) in
out(privateChannel2, (TSIDj, TIDi, Si2));
event startGW(IDg);
in(publicChannel1, (XUiM6:bitstring, XUiM7:bitstring, XUiM8:bitstring, XTID:bitstring));
let Xru2 = xor(XUiM6, Si1) in
if h(concat(Xru2, TIDi)) = XUiM8 then
new rgw5:bitstring;
let TIDinew = h(concat(concat(XIDi, rgw5), kgw)) in
let GM1 = xor(TSIDj, rgw5) in
let GM2 = xor(TIDi, rgw5) in
let GM3 = xor(TIDinew, rgw5) in
let GM4 = h(concat(concat(concat(rgw5, TSIDj),TIDi), TIDinew)) in
out(publicChannel2, (GM1, GM2, GM3, GM4, XUiM7));
in(publicChannel2, (XSNjM1:bitstring, XSNjM2:bitstring, XSNjM3:bitstring));
if h(concat(concat(XSNjM1, XSNjM2), TSIDj)) =XSNjM3 then
let GM5 = h(concat(TIDi, TIDinew)) in
let GM6 = xor(TIDinew, Si1) in
let TIDi = TIDinew in
out(publicChannel1, (GM5, GM6, XSNjM1, XSNjM2));
event endGW(IDg).
```

Table 8. Sensor node's process.

```
(*—-SNj process—-*)
let SNj =
out(privateChannel2, SIDj);
in(privateChannel2, (XTSIDj:bitstring, XXTIDi:bitstring, XXSi2:bitstring));
event startSNj(SIDj);
in(publicChannel2, (XGM1:bitstring, XGM2:bitstring, XGM3:bitstring, XGM4:bitstring,
XXUiM7:bitstring));
let Xrgw5 = xor(XGM1, XTSIDj) in
let XXTIDi = xor(XGM2, Xrgw5) in
let XXTIDinew = xor(XGM3, Xrgw5) in
let Xru3 = xor(XXUiM7, XXSi2) in
if h(concat(concat(concat(Xrgw5, XTSIDj), XXTIDi), XXTIDinew)) = XGM4 then
new SK:bitstring;
let SNjM1 = xor(SK, XXSi2) in
let SNjM2 = h(concat(concat(SK, XXTIDinew), Xru3)) in
let SNjM3 = h(concat(concat(SNjM1, SNjM2), XTSIDj)) in
let XXTIDi = XXTIDinew in
out(publicChannel2, (SNjM1, SNjM2, SNjM3));
event endSNj(SIDj).
```

Table 9. Queries and main process.

```
(*—-queries—-*)
query idi:bitstring; inj-event(endUi(idi)) ==> inj-event(startUi(idi)).
query idg:bitstring; inj-event(endGW(idg)) ==> inj-event(startGW(idg)).
query snj:bitstring; inj-event(endSNj(snj)) ==> inj-event(startSNj(snj)).
query attacker(SK).
(*—-process—-*)
process
((!Ui) | (!GW) | (!SNj))
```

Table 10. Result.

Verification summary:
Query inj-event(endUi(idi)) ==> inj-event(startUi(idi)) is true.
Query inj-event(endGW(idg)) ==> inj-event(startGW(idg)) is true.
Query inj-event(endSNj(snj)) ==> inj-event(startSNj(snj)) is true.
Query not attacker(SK[]) is true.

When we run the following query in Table 9, we can obtain the following results:

1. Query inj–event(endEVENTA) ==> inj-event(startEVENTA) is true.
2. Query inj–event(endEVENTB) ==> inj-event(startEVENTB) is false.
3. Query not attacker(M) is true.
4. Query not attacker(M) is false.

"Query inj-event (endEVENTA) == > inj-event (startEVENTA) is true." means that the process from endEVENTA to startEVENTA has been authenticated. By contrast, "Query inj-event (endEVENTB) == > inj-event (startEVENTB) is false." means that the authentication from endEVENTB to startEVENTB is not successful. "Query not attacker (M) is true." means that an attacker cannot acquire a free name M. Finally, "Query not attacker (M) is false." means that an attacker can trace the M.

The query results from Table 9 are listed in Table 10.

7.2. Informal Security Analysis

We performed a formal analysis. However, a formal analysis by itself is not sufficient to prove safety [13,26,27]. Therefore, we further analyzed our scheme using an informal analysis. We present a theoretical analysis of the proposed scheme. The results of the informal security analysis are then briefly described.

1. Offline Password Guessing Attack: Since our scheme uses biometric information B_i with the unique biological characteristics of individuals that are not stored for user authentication, it is impossible to guess a user's password without a real user. Therefore our scheme can protect against the offline password guessing attack.
2. Privileged Insider Attack: Even if the privileged insider steals ID_i, TID_i, SID_j, $TSID_j$, and S_i^1 from the gateway's database, the privileged insider can not obtain the session key SK without secret information S_i^2 that is shared between the user and the sensor node. Therefore our scheme can protect against privileged insider attacks.
3. User Impersonation Attack: Even if the attacker steals and replaces the user's TID_i, the attacker can not generate valid U_iM_6 and U_iM_7 without secret information S_i^1 and S_i^2. When the gateway and the sensor node verify U_iM_6 and U_iM_7, respectively, they can find the invalid user. Therefore our scheme can protect against user impersonation attacks.
4. Server Impersonation Attack: Even if the attacker impersonates the gateway, the attacker does not generate valid GM_4 and GM_5 without $TSID_j$ and S_i^1. When the sensor node and the user verify GM_4 and GM_5, respectively, they can find the invalid gateway. Therefore our scheme can protect against server impersonation attacks.
5. Replay Attack: Even if the attacker steals U_iM_6, U_iM_7, U_iM_8, and TID_i from a successful mutual authentication and key distribution phase and then resends it to the gateway, the gateway can find whether or not the message is reused because the gateway checks r_U^2's freshness. Moreover, the attacker can not generate and modify r_U^2 and U_iM_6 without S_i^1. Therefore our scheme can protect against replay attacks.
6. Man-in-the-Middle Attack: In a man-in-the-middle attack, an attacker puts themselves in the middle of two parties so that they can intercept and modify some communicated data to masquerade as the entities. In the mutual authentication and key distribution phase, the attacker intercepts communicated data between the user and the gateway and attempts to modify the message to retrieve the session key. However, in our scheme, the attacker can not modify communicated messages

without the secret information S_i^1 and S_i^2. Therefore our scheme can protect against man-in-the-middle attacks.

7. Session Key Disclosure Attack: Even if the attacker obtains SN_jM_1 which includes the session key SK, the attacker can not obtain the session key without the secret information S_i^2. Therefore, our scheme can protect against session key disclosure attacks.
8. Forward Secrecy and Backward Secrecy: Even if someone gains the session key SK, they can not know the old session key or the new session key because each session key is generated randomly with no relation to the other session keys. Therefore our scheme can preserve forward secrecy and backward secrecy.
9. Mutual Authentication: The gateway and the user can authenticate each other by verifying U_iM_8 and GM_5 respectively, using the secret information S_i^1. Therefore our scheme provides mutual authentication.
10. User Anonymity: Our scheme identifies users using TID_i and then replaces it every time with TID_i^{new} regardless of the old TID_i. Therefore our scheme preserves user anonymity.

The results of the security analysis with comparisons to related papers are presented in Table 11.

Table 11. Comparisons of the security features.

Security Features	Wu et al. [12]	Li et al. [15]	Masud et al. [6]	Ours
1. Resist Offline Password Guessing Attack	O	O	X	O
2. Resist Privileged Insider Attack	O	O	X	O
3. Resist User Impersonation Attack	X	O	O	O
4. Resist Server Impersonation Attack	O	X	O	O
5. Resist Replay Attack	X	X	X	O
6. Resist Man-in-the-Middle Attack	O	O	O	O
7. Resist Session Key Disclosure Attack	X	O	O	O
8. Preserve Forward Secrecy and Backward Secrecy	O	O	O	O
9. Provide Mutual Authentication	O	O	O	O
10. Preserve Anonymity	X	X	X	O

8. Performance Analysis of the Proposed Scheme

Many authentication studies have analyzed their performance in the following manner [9–15,20–22,24–32]. We compared each computation amount in terms of the research methods used. We analyze the performance of our scheme as follows.

Our study analyzes the computational cost using the time measurement presented in Table 12 [28,29]. T_M stands for the computational cost of multiplication in the field. T_{bh} stands for the computational cost of the biohash function operation, and T_h stands for the computational cost of the one-way hash function operation. It is assumed that the XOR operation does not affect the cost of operation. Table 13 and Figure 2 compare the computational cost of our scheme with those of other schemes according to Table 12 [6,12,15].

Table 12. Computational cost of cryptographic calculations (ms).

Symbol	Meaning	Time
T_m	The computational cost of multiplication in the field.	7.3529 [28]
T_{bh}	The computational cost of the biohash function operation.	7.3529 [29]
T_h	The computational cost of the one-way hash function operation.	0.0004 [28]

Table 13. Comparisons of computational cost.

Schemes	Wu et al. [12]	Li et al. [15]	Masud et al. [6]	Ours
User	$T_{bh} + 10T_h + 2T_m$ $= 22.0627$	$T_{bh} + 10T_h + 2T_m$ $= 22.0627$	$3T_h$ $= 0.0012$	$T_{bh} + 6T_h$ $= 7.3553$
GWN	$8T_h$ $= 0.0032$	$8T_h + T_m$ $= 7.3561$	$4T_h$ $= 0.0016$	$5T_h$ $= 0.002$
Sensor Node	$2T_h + 2T_m$ $= 14.7066$	$4T_h + 2T_m$ $= 14.7074$	$2T_h$ $= 0.008$	$3T_h$ $= 0.0012$
Total	$T_{bh} + 20T_h + 4T_m$ $= 36.7725$	$T_{bh} + 22T_h + 5T_m$ $= 44.1262$	$9T_h$ $= 0.0036$	$T_{bh} + 14T_h$ $= 7.3585$

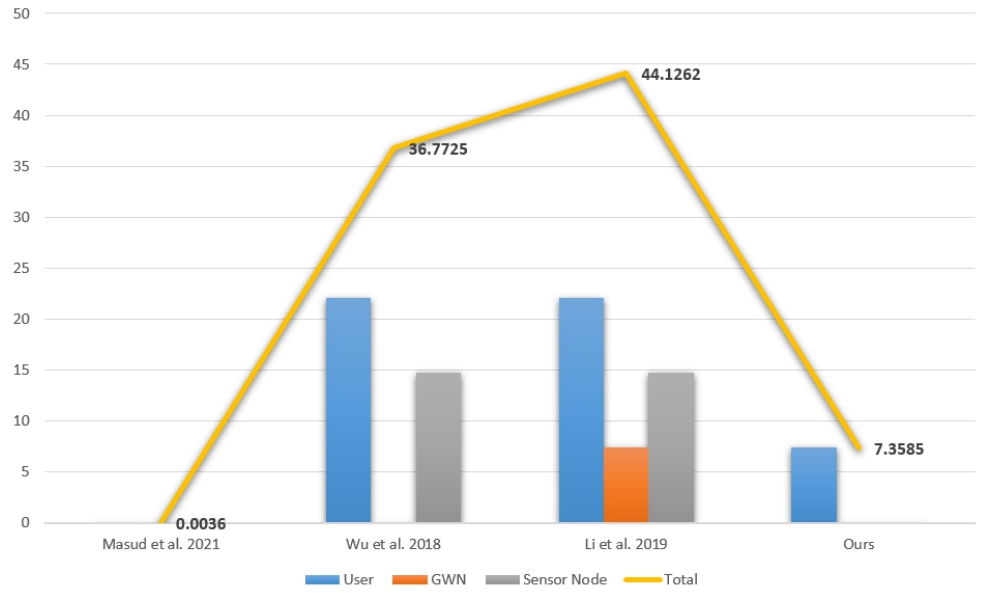

Figure 2. Graph comparisons of computational cost [6,12,15].

We calculate the computational efficiency of our scheme as follows:

$$(t_1 - t_2)/t_2 \tag{3}$$

In Formula (3), t_1 represents the average cost of computation of the different schemes. Moreover, t_2 represents the cost of operation of our scheme.

According to the above formula, the operation of our scheme is 266.48% more efficient in terms of computational cost than the other schemes, and Table 11 shows that our scheme is more secure than the other methods.

9. Discussion of Performance

We proposed a secure and lightweight user authentication scheme for IoMT by improving Masud et al. [6]'s scheme. We compared the performance of three schemes [6,12,15] in Section 8. Our scheme outperforms [12,15] by 399.73% and 499.66% respectively. The performance of [6] is lightweight, but it does not meet basic security requirements such as offline password guessing attacks, privileged insider attacks, and replay attacks. Therefore, our scheme is a suitable lightweight user authentication scheme for IoMT because our scheme not only is improved by addressing the security threats of [6] but also outperforms 266.48% more efficiently than the other schemes.

10. Conclusions

The purpose of our paper was to propose a secure and lightweight user authentication scheme for IoMT by addressing the security threats to which Masud et al. [6]'s scheme is vulnerable. In particular, our scheme can protect against well-known attacks in IoMT i.e.,

offline password guessing attacks, privileged insider attacks, user impersonation attacks, replay attacks, and session key disclosure attacks, and it ensures user anonymity. We also proved that our scheme is a suitable user authentication scheme for IoMT through formal security analysis by ProVerif. Moreover, we proposed a lightweight security protocol that mainly uses a hash function and XOR operation considering low-spec healthcare sensors. As a result, we showed 266.48% better performance than the average computational cost of the considered schemes [6,12,15]. Our scheme outperforms [12,15], but it does not outperform [6]. Our scheme shows higher safety than the compared schemes [6,12,15]. Our security and performance analysis shows that our scheme is a suitable lightweight user authentication scheme for IoMT. Further studies will be able to improve convenience by combining behavioral biometrics authentication. Behavioral biometric authentication is expected to achieve further improved convenience over biometrics authentication because it uses keystroke dynamics, gait analysis, mouse use characteristics, signature analysis, and cognitive biometrics.

Author Contributions: Conceptualization, K.K.; Methodology, K.K., J.R. and Y.L.; Software, K.K.; Validation, J.R.; Formal analysis, K.K. and Y.L.; Writing—original draft, K.K. and J.R.; Writing—review and editing, J.R., Y.L. and D.W.; Supervision, D.W.; Funding acquisition, D.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Thiswork was supported by an Institute of Information & Communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00558, Development of National Statistical Analysis System using Homomorphic Encryption Technology).

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

References

1. Jha, N.K. Internet-of-Medical-Things. In Proceedings of the Great Lakes Symposium on VLSI (GLSVLSI), Banff, AB, Canada, 10–12 May 2017; p. 7.
2. Hatzivasilis, G.; Soultatos, O.; Ioannidis, S.; Verikoukis, C.; Demetriou, G.; Tsatsoulis, C. Review of security and privacy for the Internet of Medical Things (IoMT). In Proceedings of the 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), Santorini Island, Greece, 29–31 May 2019; pp. 457–464.
3. Dilibal, C.; Davis, B.L.; Chakraborty, C. Generative design methodology for internet of medical things (IoMT)-based wearable biomedical devices. In Proceedings of the 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 11–13 June 2021; pp. 1–4.
4. Aman, A.H.M.; Hassan, W.H.; Sameen, S.; Attarbashi, Z.S.; Alizadeh, M.; Latiff, L.A. IoMT amid COVID-19 pandemic: Application, architecture, technology, and security. *J. Netw. Comput. Appl.* **2021**, *174*, 102886. [CrossRef] [PubMed]
5. Khadidos, A.O.; Shitharth, S.; Khadidos, A.O.; Sangeetha, K.; Alyoubi, K.H. Healthcare Data Security Using IoT Sensors Based on Random Hashing Mechanism. *J. Sens.* **2022**, *2022*, 8457116. [CrossRef]
6. 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]
7. Lamport, L. Password authentication with insecure communication. *Commun. ACM* **1981**, *24*, 770–772. [CrossRef]
8. Liao, I.E.; Lee, C.C.; Hwang, M.S. A password authentication scheme over insecure networks. *J. Comput. Syst. Sci.* **2006**, *72*, 727–740. [CrossRef]
9. Wu, Z.Y.; Lee, Y.C.; Lai, F.; Lee, H.C.; Chung, Y. A secure authentication scheme for telecare medicine information systems. *J. Med. Syst.* **2012**, *36*, 1529–1535. [CrossRef]
10. Debiao, H.; Jianhua, C.; Rui, Z. A more secure authentication scheme for telecare medicine information systems. *J. Med. Syst.* **2012**, *36*, 1989–1995. [CrossRef]
11. Wei, J.; Hu, X.; Liu, W. An improved authentication scheme for telecare medicine information systems. *J. Med. Syst.* **2012**, *36*, 3597–3604. [CrossRef]
12. Wu, F.; Xu, L.; Kumari, S.; Li, X. An improved and provably secure three-factor user authentication scheme for wireless sensor networks. *Peer-Peer Netw. Appl.* **2018**, *11*, 1–20. [CrossRef]

13. Ryu, J.; Lee, H.; Kim, H.; Won, D. Secure and efficient three-factor protocol for wireless sensor networks. *Sensors* **2018**, *18*, 4481. [CrossRef]
14. Mao, D.; Liu, H.; Zhang, W. An enhanced three-factor authentication scheme with dynamic verification for medical multimedia information systems. *IEEE Access* **2019**, *7*, 167683–167695. [CrossRef]
15. 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]
16. Ebrahimi, S.; Bayat–Sarmadi, S. Lightweight fuzzy extractor based on LPN for device and biometric authentication in IoT. *IEEE Internet Things J.* **2021**, *8*, 10706–10713. [CrossRef]
17. Satamraju, K.P.; Malarkodi, B. A PUF-based mutual authentication protocol for internet of things. In Proceedings of the 2020 5th International Conference on Computing, Communication and Security (ICCCS), Patna, India, 14–16 October 2020; pp. 1–6.
18. Abdaoui, A.; Erbad, A.; Al-Ali, A.K.; Mohamed, A.; Guizani, M. Fuzzy Elliptic Curve Cryptography for Authentication in Internet of Things. *IEEE Internet Things J.* **2021**, *9*, 9987–9998. [CrossRef]
19. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; pp. 523–540.
20. Ryu, J.; Lee, H.; Lee, Y.; Won, D. SMASG: Secure Mobile Authentication Scheme for Global Mobility Network. *IEEE Access* **2022**, *10*, 26907–26919. [CrossRef]
21. Kang, D.; Lee, H.; Lee, Y.; Won, D. Lightweight user authentication scheme for roaming service in GLOMONET with privacy preserving. *PLoS ONE* **2021**, *16*, e0247441. [CrossRef]
22. Ryu, J.; Kang, D.; Lee, H.; Kim, H.; Won, D. A secure and lightweight three-factor-based authentication scheme for smart healthcare systems. *Sensors* **2020**, *20*, 7136. [CrossRef]
23. Blanchet, B.; Smyth, B.; Cheval, V.; Sylvestre, M. ProVerif 2.04: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial. Available online: https://proverif.inria.fr/manual.pdf (accessed on 30 November 2021).
24. Kang, D.; Jung, J.; Lee, D.; Kim, H.; Won, D. Security analysis and enhanced user authentication in proxy mobile IPv6 networks. *PLoS ONE* **2017**, *12*, e0181031. [CrossRef]
25. Roy, S.; Chatterjee, S.; Das, A.K.; Chattopadhyay, S.; Kumari, S.; Jo, M. Chaotic map–based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing Internet of Things. *IEEE Internet Things J.* **2017**, *5*, 2884–2895. [CrossRef]
26. Lee, H.; Lee, D.; Moon, J.; Jung, J.; Kang, D.; Kim, H.; Won, D. An improved anonymous authentication scheme for roaming in ubiquitous networks. *PLoS ONE* **2018**, *13*, e0193366. [CrossRef]
27. Jung, J.; Kim, J.; Choi, Y.; Won, D. An anonymous user authentication and key agreement scheme based on a symmetric cryptosystem in wireless sensor networks. *Sensors* **2016**, *16*, 1299. [CrossRef] [PubMed]
28. Xu, L.; Wu, F. Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care. *J. Med. Syst.* **2015**, *39*, 1–9. [CrossRef] [PubMed]
29. Das, A.K. A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks. *Peer-Peer Netw. Appl.* **2016**, *9*, 223–244. [CrossRef]
30. Sahoo, S.S.; Mohanty, S.; Majhi, B. An efficient three-factor user authentication scheme for industrial wireless sensor network with fog computing. *Int. J. Commun. Syst.* **2022**, *35*, 3. [CrossRef]
31. Bahache, A.N.; Chikouche, N.; Mezrag, F. Authentication Schemes for Healthcare Applications Using Wireless Medical Sensor Networks: A Survey. *SN Comput. Sci.* **2022**, *3*, 1–25. [CrossRef]
32. Li, Y.; Tian, Y. A Lightweight and Secure Three-Factor Authentication Protocol With Adaptive Privacy-Preserving Property for Wireless Sensor Networks. *IEEE Syst. J.* **2022**, *16*, 6197–6208. [CrossRef]