electronics

Article

PUF-PSS: A Physically Secure Privacy-Preserving Scheme Using PUF for IoMT-Enabled TMIS

Sungjin Yu [1,2] and Kisung Park [1,*]

[1] Electronics and Telecommunications Research Institute, Daejeon 34129, Korea
[2] School of Electronics and Electrical Engineering, Kyungpook National University, Daegu 41566, Korea
* Correspondence: ks.park@etri.re.kr; Tel.: +82-42-860-5797

Abstract: With the development of telecare medical information system (TMIS), doctors and patients are able to access useful medical services via 5G wireless communications without visiting the hospital in person. Unfortunately, TMIS should have the essential security properties, such as anonymity, mutual authentication, and privacy, since the patient's data is transmitted via a public channel. Moreover, the sensing devices deployed in TMIS are resource-limited in terms of communication and computational costs. Thus, we design a physically secure privacy-preserving scheme using physical unclonable functions (PUF) in TMIS, called PUF-PSS to resolve the security requirements and efficiency of the existing related schemes. PUF-PSS prevents the security threats and also guarantees anonymity, key freshness, and authentication. We evaluate the security of PUF-PSS by performing formal and informal security analyses, including AVISPA implementation and ROR oracle model. We perform the test bed experiments utilizing well-known MIRACL based on a Raspberry PI 4 and compare the communication and computational costs of PUF-PSS with the previous schemes for TMIS. Consequently, PUF-PSS guarantees better efficiency and security than previous schemes and can be applied to TMIS environments.

Keywords: telecare medical information systems; healthcare; physical unclonable function; privacy-preserving; security protocol

Citation: Yu, S.; Park, K. PUF-PSS: A Physically Secure Privacy-Preserving Scheme Using PUF for IoMT-Enabled TMIS. *Electronics* **2022**, *11*, 3081. https://doi.org/10.3390/electronics11193081

Academic Editor: Cheng-Chi Lee

Received: 2 September 2022
Accepted: 23 September 2022
Published: 27 September 2022

1. Introduction

The recent COVID-19 pandemic has posed "one of the most serious threats to patient safety ever recorded, and public health is confronted with one of humanity's and the world's greatest challenges" [1]. The situational factors such as redeployment to unfamiliar roles and health professional shortages due to COVID-19 have all hampered existing care processes in most healthcare systems around the world. If specific precautions are not presented to resolve these problems, potential medical threats are able to result in many deaths. In this regard, numerous researchers have studied applicative and systematic methods for preventing medical deaths and improving patient safety for many years.

With the development of "5G wireless communications" and "internet of medical things (IoMT)" technologies, users can access medical services, including diagnostics and treatments via telecare medical information systems (TMIS). IoMT-based TMIS provides various healthcare services, such as health response, rehabilitation, and health monitoring [2,3]. These applications can greatly help patients and doctors to ensure efficient, robust, and low-cost healthcare services in "low and middle-income countries" and carry out exact medical diagnoses. In general, IoMT-based TMIS have consisted of the TMIS server, user, and sensing device. The sensing devices (e.g., IoMT and wearable devices) collect and monitor the patient's health data, including body temperature and blood pressure, and send health data to the medical systems for treatment. Furthermore, a TMIS server guarantees other medical information and healthcare services to the users. The doctors may access the TMIS server to get patient's real-time health status. Unfortunately, despite the

advantages of TMIS, there are several challenges and problems to be resolved. IoMT-based TMIS may cause serious privacy and security issues [4] because the information is transmitted over an open channel. If the patient's data is exposed, an adversary may attempt potential security threats. Moreover, an adversary can attempt physical sensor capture attacks and extract the secret information from a physically captured sensing devices. In addition, since the sensing device is resource limited with regard to communication and computational overheads, it is not applicable to utilize "symmetric and asymmetric key cryptography" that needs high overheads. Thus, lightweight and robust authentication and key agreement (AKA) schemes are indispensable to providing effective healthcare services for IoMT-based TMIS.

Over the past few years, numerous researchers have designed a lightweight and robust AKA protocol for IoMT-based TMIS [5,6]. They claimed that their AKA protocol can resist potential physical/cyber security attacks, including "sensing device capture, session key disclosure, privileged insider, and impersonation attacks", and also guarantee "user anonymity, mutual authentication, and key freshness". However, the existing AKA schemes for IoMT-based TMIS are vulnerable to potential physical/cyber security threats and also fail to provide essential security features, such as untraceability, anonymity, and mutual authentication. In addition, the existing AKA schemes for TMIS are not suitable for resource-limited sensing devices since it uses public-key cryptosystems (PKC) that require high computational and communication overheads. Hence, we design a physically secure privacy-preserving AKA scheme using physical unclonable functions (PUF) for IoMT-based TMIS, called PUF-PSS, to address the efficiency and security issues of the related schemes.

1.1. Motivations

Recently, the various applications for healthcare in IoMT-based TMIS environments ensure multiple benefits and useful services to legitimate users. However, despite the multiple benefits of TMIS application, the previous AKA schemes for TMIS suffered from cyber security threats, including insider attacks, impersonation, offline password guessing attacks, a lack of security functionalities, and also caused damage and overload to the systems. Besides cyber security threats, the sensing devices in IoMT-based TMIS can be vulnerable to physical security attacks since they are deployed in unattended and hostile environments. This fact motivated us to design a "physically secure privacy-preserving scheme using PUF for IoMT-based MITS" that resolves potential "cyber/physical security attacks" and ensures the "essential security requirements" that exist in IoMT-based TMIS environments.

1.2. Contributions

The detailed contributions of this article can be summarized below:

- We design a "physically secure privacy-preserving scheme using PUF for IoMT-based TMIS" to improve the security weaknesses of the related AKA schemes. PUF-PSS ensures the low overheads suitable for IoMT-based TMIS by performing XOR and hash functions. Moreover, PUF-PSS using PUF ensures that the physical security of the smart devices deployed in IoMT-based TMIS environments.
- We carry out the formal security analysis using "Real-or-Random (ROR) model" [7] and "Automated Validation of Internet Security Protocols and Applications (AVISPA)" simulation [8] to demonstrate the security of PUF-PSS.
- We present the test bed experiments for various forms of cryptography utilizing "Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL)" [9].
- We demonstrate that PUF-PSS guarantees mutual authentication between each entity by performing Burrows–Abadi–Needham (BAN) logic [10].
- We evaluate the performance of PUF-PSS with existing schemes with regard to "security properties, computation cost, and communication cost".

1.3. Organization

The rest of the article is organized as follows. Section 2 introduces the related works for TMIS and Section 3 presents preliminaries. Section 4 designs a physically secure privacy-preserving AKA scheme using PUF for IoMT-based TMIS to enhance the security shortcomings and efficiency of the existing schemes. In Section 5, we attest to the security of PUF-PSS by performing "informal security and formal security analyses". Section 6 indicates the test bed experiments for cryptographic operations utilizing MIRACL and then Section 7 compares the security functionalities, computation, and communication overheads of PUF-PSS with existing schemes. Finally, we summarize the conclusion and future works in Section 8.

2. Related Works

Over the past decades, many AKA schemes [11–13] have been proposed for healthcare in IoMT-based TMIS to ensure privacy and security of legitimate users. Amin et al. [14] presented an elliptic curve cryptography (ECC)-based AKA protocol that allows servers and users to share temporal common keys. Unfortunately, their scheme [14] is vulnerable to offline password guessing and masquerade attacks and has high computational costs. Challa et al. [15] designed an efficient and robust three-factor-based AKA scheme for healthcare using ECC. However, their scheme [15] cannot prevent "forgery and replay attacks and does not guarantee mutual authentication". Li et al. [16] proposed a robust and efficient three-factor AKA scheme using ECC for wireless medical sensor systems (WMSN). Unfortunately, their scheme [16] is fragile to "replay and privileged insider attacks". Furthermore, these AKA schemes [14–16] are not applicable for actual IoMT-enabled TMIS since they use ECC which is generated with high overheads.

Numerous researchers have presented a lightweight AKA scheme for IoMT-based TMIS [17–19] to address the efficiency associated with ECC-based AKA and the security problems. Sharma et al. [20] designed an efficient and reliable AKA protocol for cloud-IoT-enabled healthcare. Nevertheless, Sharma et al.'s scheme [20] is fragile to "sensor node compromise and insider attacks and does not ensure untraceability and anonymity". Wazid et al. [21] presented a reliable AKA protocol for edge-based IoT environments using hash and XOR functions, called LDAKM-EIoT. However, LDAKM-EIoT is insecure to "forgery and desynchronization attacks". Zhou et al. [22] proposed a reliable and lightweight IoT-enabled AKA protocol applicable to cloud-based TMIS. However, Zhou et al.'s scheme [22], similar to LDKAM-EIoT, is resistant to potential security attacks. In addition, these AKA schemes [20–22] guarantee user-friendly and inefficient scalability since it is not included that "user pre-validation and passwords cannot be efficiently changed without server involvement".

In recent years, numerous biometric-based lightweight and robust AKA schemes for TMIS [23–25] have been proposed to address the security, efficiency, and scalability issues. Gupta et al. [26] proposed a robust and lightweight AKA protocol with anonymity for wearable device-based medical services. Gupta et al.'s scheme [26] guarantees high scalability and low computing resources. Unfortunately, Hajian et al. [27] discover that Gupta et al.'s scheme [26] suffers from "potential security threats such as privileged insider, offline guessing, impersonation, desynchronization, and compromise sensing device attacks". Moreover, these AKA schemes [26,27] guarantee "high scalability but may be fragile to physical sensor capture attacks because it does not require secure channel during the sensing device registration process". Thus, we design a physically robust privacy-preserving scheme using PUF for IoMT-based TMIS to resolve the security problems of existing related schemes.

3. Preliminaries

We introduce the preliminaries for this article.

3.1. Physical Unclonable Functions

PUF is considered as a "solution for protecting smart devices with low computing capabilities from an adversary [28]". In the last few years, many researchers have presented various PUF mechanisms, such as static random access memory (SRAM)-PUF for lightweight property [29], ring oscillator (RO)-PUF for reliability improvement [30], and quantum-PUF [31] for quantum attack resistance [32–34]. Especially, the sensing devices deployed in IoMT-enabled TMIS are suitable to apply SRMA PUF property because it is resource-limited with regard to memory, power, and computing. PUF is widely used to manufacture an "output for an input such as a fingerprint-based on the physical microstructure of the smart devices". PUF does not store a "secret key on the smart devices and is practically difficult to clone successfully identical PUF" because it is "formed by generating nanoscale variations during the integrated circuit (IC) chip's manufacturing process". The ideal PUF offers the functionalities of "unpredictability, uniqueness, and reliability". PUF secures the smart devices deployed in IoMT-based TMIS environments from cloning, tampering, and side-channel attacks. Since PUF depends on the unique physical features of the IC, any alteration to the system will change the PUF output. PUF allows the systems to prove the legality of the smart devices and entities prior to establishing a common session key [35]. The detailed features of the PUF are as described below:

- PUF is easy to evaluate and implement.
- Any try to tamper with the smart devices which contain PUF will update the behavior of the PUF and thus destroy it [36].
- PUF relies on the system's physical microstructure.

As a result, these features combine to make a good solution for the authentication and group proof in IoMT-based TMIS environments.

3.2. Adversary Model

We introduce the adversary models such as the widely accepted "Dolev–Yao (DY)" model [37] and "Canetti and Krawczyk (CK) model" [38].

- In the DY model [37], a malevolent adversary (MA) can block, inject, eavesdrop, and resend the transmitted messages over an open channel.
- In the CK model [38], MA can compromise "secret credentials and session states through session-hijacking attacks". Therefore, a session key must be dependent on both "long-term secret or short-term secret credentials".
- MA can steal a mobile device (MD) of legal users and also has the ability to physically capture sensor devices by performing a differential power analysis [39,40]. Thus, MA extracts the secret parameters stored in MD or sensing devices [41].

3.3. Network Model

Similar to [26], we introduce the network model for healthcare that is a combination of TMIS, IoMT, and WBAN. As shown in Figure 1, the network model is comprised of three entities: the patient, sensing device, and TMIS server.

- TMIS server: This entity is a powerful and trusted authority and includes a secure database that stores medical information for legitimate patients. Moreover, TMIS server is responsible for the registration and mutual authentication of the user/gateway and wearable sensing devices.
- User/Gateway: This entity is a user or gateway terminal, such as an access point and an MD in the ambulance access point or the smart home. The gateway acts as a bridge between mobile/sensing devices and the TMIS server by providing short and long distance communication interfaces that maintain connectivity with internal mobile users and sensing devices. Hence, the gateway provides real-time communication between internal and external environments. In the case of an emergency when a patient is transported to the hospital, the patient needs to be connected to one of

the TMIS servers since he/she cannot have access to the mobile terminal. Thus, we indicate various types of gateways that are not limited to mobile terminals.

- Sensing device: This entity is a wearable sensing device, including a smart watch, heart rate sensor, and smart wristband, which is implanted on a patient's body or deployed by them in homes. *SD*s are resource constrained with regard to computing power, memory, and computation cost.

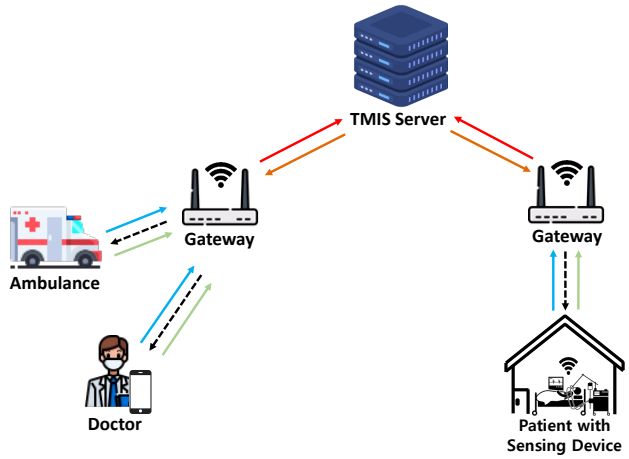

Figure 1. TMIS network model.

4. Proposed Scheme

We design a "physically robust privacy-preserving scheme using PUF for IoMT-based TMIS" to improve the security shortcomings of the existing AKA scheme for IoMT-based TMIS. The proposed scheme provides superior scalability since it uses a public channel in the process of the registration of each *SD*. Moreover, the proposed scheme contains the biometric and password update phase without the involvement of the trusted authority. The notations utilized in this paper are as shown in Table 1.

Table 1. Notations.

Notation	Description
U_i/GW_i	User and Gateway
SD_j	Sensing device
S	TMIS server
GID_i	Identity of GW_i
ID_i	Identity of U_i
SID_j	Identity of SD_j
PW_i	Password of U_i
BIO	Biometric of U_i
SK_{ij}	Session key between GW_i and SD_j
K_i	Secret key of S
X_{GD_i}	Common secret key between GW_i and S
X_{SD_j}	Common secret key between SD_j and S
T_i	Timestamp
ΔT	Maximum transmission delay
$h(\cdot)$	Hash function
$PUF(\cdot)$	Physical unclonable function
$\oplus$	XOR operation
$\|\|$	Concatenation

4.1. System Setup Process

This process consists of two cases such as gateway setup and sensing device setup processes. A trusted authority (TA) or TMIS server must register the SD_j and assign the unique values to it. The TMIS server chooses a secret key X_{SD_j}, a temporal identity $TSID_j$, and an unique identity SID_j for each SD. Then, the TMIS server stores $\{SID_j, TSID_j, X_{SD_j}\}$ in SD_j's memory. In addition, the TMIS server stores $\{SID_j, TSID_j, X_{SD_j}\}$ in secure database. In order to register a gateway, the TMIS server chooses a secret key X_{GD_i}, a temporal identity $TGID_i$, and a unique identity GID_i for each gateway and stores $\{GID_j, TGID_j, X_{GD_i}\}$ in the memory of the gateway. The TMIS server stores $\{GID_j, TGID_j, X_{GD_i}\}$ in a secure database.

4.2. Registration Process

The process consists of two parts: the sensing device and user registration processes.

4.2.1. User Registration Process

U_i should register with S to receive the healthcare services. We introduce the user registration process of PUF-PSS and it is described in detail below as follows:

UR-1: U_i chooses a "ID_i, PW_i and selects a random number n_i". After that, U_i calculates $RPW_i = h(PW_i||n_i)$ and sends $\{ID_i.RPW_i\}$ to S over a secure channel.

UR-2: S computes $HID_i = h(ID_i||X_{GD_i}||K_i)$ and $X_i = h(K_i||X_{GD_i}||HID_i)$, and stores $\{HID_i\}$ in a secure database. Then, S transmits $\{HID_i, X_i\}$ to the U_i through a secure channel.

UR-3: U_i imprints BIO_i and computes $\gamma_i = PUF(BIO_i)$, $(\alpha_i, \beta_i) = Gen(\gamma_i)$, $\beta_i^* = \beta \oplus h(ID_i||X_{GD_i}||PW_i)$, $n_i^* = n_i \oplus h(ID_i||\alpha_i||PW_i)$, $X_i^* = X_i \oplus h(ID_i||\alpha_i||X_{GD_i}||RPW_i)$, $HID_i^* = HID_i \oplus h(\alpha_i||RPW_i||X_i||X_{GD_i})$, and $C_i = h(HID_i||\alpha_i||X_i||X_{GD_i})$. After that, U_i replaces $\{HID_i, X_i\}$ with $\{HID_i^*, X_i^*\}$ and then stores $\{n_i^*, \beta_i^*, C_i\}$ in the memory.

4.2.2. Sensing Device Registration Process

We show the sensing device registration process of PUF-PSS and it is described in detail as follows.

SDR-1: SD_j chooses a random number b_j and calculates $Q_j = b_j \oplus h(SID_j||X_{SD_j})$ and $W_j = h(SID_j||TSID_j||X_{SD_j}||b_j)$. Then, SD_j transmits the message $\{Q_j, W_j, TSID_j\}$ to the S over an insecure channel.

SDR-2: S calculates $b_j = Q_j \oplus h(SID_j||X_{SD_j})$, $W_j^* = h(SID_j|||TSID_j||X_{SD_j}||b_j)$, and checks whether $W_j^* \stackrel{?}{=} W_j$. If the condition is equal, S computes $HSID_j = h(SID_j||X_{SD_j}||K_i)$, $Z_j = h(HSID_j||K_i||X_{SD_j})$, $N_j = (HSID_j||b_j) \oplus h(X_{SD_j}||SID_j||$ $TSID_j)$, and $M_j = Z_j \oplus h(HSID_j||X_{SD_j}||b_j)$. After that, S generates a random challenge set C_j and computes the response set Res_j for the C_j as $Res_j = PUF(C_j)$. Then, the sets R_j and δ_j are computed by passing Res_j via PUF function $Gen(.)$, where $(R_j, \delta_j) = Gen(Res_j)$.

SDR-3: After that, S computes $D_j = \delta_j \oplus h(X_{SD_j}||b_j||HSID_j)$, $F_i = h(b_j||X_{SD_j}||HSID_j||$ $TSID_j||N_j)$, and transmits $\{N_j, M_j, D_j, F_j\}$ to the SD_j. Finally, S computes $V_j = Z_j \oplus K_i \oplus X_{SD_j}$ and then stores $\{V_j, (C_j, R_j)\}$ in the secure database.

SDR-4: SD_j computes $F_j^* = h(b_j||X_{SD_j}||HSID_j||TSID_j||N_j)$ and checks whether $F_j^* \stackrel{?}{=} F_j$. If it is valid, SD_j stores the secret credentials $\{C_j, N_j, M_j, D_j\}$ in the memory.

4.3. Authentication and Key Agreement Process

After performing the registration process, the registered U_i and SD_j carry out mutual authentication with S in order to establish a session key. The messages are transmitted via an open channel. We present the detailed AKA process of the PUF-PSS below.

AKP-1: U_i/GW_i inputs a ID_i and PW_i, and imprints BIO_i. After that, MD calculates $\gamma_i = PUF(BIO_i)$, $\beta_i = \beta_i^* \oplus h(ID_i||X_{GD_i}||PW_i)$, $\alpha_i = Rep(\gamma_i, \beta_i)$, $n_i = n_i^* \oplus h(ID_i||\alpha_i||PW_i)$, $RPW_i = h(PW_i||n_i)$, $X_i = X_i^* \oplus h(ID_i||\alpha_i||X_{GD_i}||RPW_i)$, $HID_i = HID_i^* \oplus h(\alpha_i||RPW_i||X_i||X_{GD_i})$, and $C_i^* = h(HID_i||\alpha_i||X_i||X_{GD_i})$, and checks whether $C_i^* \stackrel{?}{=} C_i$. "If it is not valid, U_i terminates this process, otherwise U_i selects a r_1 and calculates $M_1 = r_1 \oplus h(TGID_i||X_i||X_{GD_i})$, and $Auth_u = h(HID_i||r_1||X_i||X_{GD_i}||TGID_i)$, and transmits $\{TGID_i, M_1, Auth_u\}$ to SD_j".

AKP-2: SD_j computes $(HSID_j||b_j) = N_j \oplus h(X_{SD_j}||SID_j||TSID_j)$, $Z_j = M_j \oplus h(HSID_j||X_{SD_j}||b_j)$, and generates a random nonce r_2. After that, SD_j computes $M_2 = (SID_j||r_2) \oplus h(X_{SD_j}||Z_j)$ and $Auth_{SD} = h(HSID_j||X_{SD_j}||Z_j||r_2)$, and transmits $\{TGID_i, M_1, Auth_u, TSID_j, M_2, Auth_{SD}\}$ to S.

AKP-3: S retrieves $\{HID_i\}$ with $TGID_i$ in the secure database and computes $X_i = h(K_i||X_{GD_i}||HID_i)$, $r_1 = M_1 \oplus h(TGID_i||X_i||X_{GD_i})$, and $Auth_u^* = h(HID_i||r_1||X_i||X_{GD_i}||TGID_i)$. Then, S checks whether $Auth_u^* \stackrel{?}{=} Auth_u$. If it is valid, S computes $Z_j = V_j \oplus K_i \oplus X_{SD_j}$, $(SID_j||r_2) = M_2 \oplus h(X_{SD_j}||Z_j)$, $HSID_j = h(SID_j||X_{SD_j}||K_i)$, and $Auth_{SD}^* = h(HSID_j||X_{SD_j}||Z_j||r_2)$, and then checks whether $Auth_{SD}^* \stackrel{?}{=} Auth_{SD}$. If it is correct, S retrieves (C_j, R_j) through SID_j and computes $M_3 = (C_j||r_1) \oplus h(Z_j||HSID_j||r_2||X_{SD_j})$, $TSID_j^{new} = h(r_2||TSID_j)$, $Auth_{TM-SD} = h(TSID_j^{new}||X_{SD_j}||Z_j||R_j||r_1)$, and $Auth_{TM-U} = h(TGID_i||X_{GD_i}||X_i||r_1||r_2)$, and then transmits $\{Auth_{TM-SD}, Auth_{TM-U}, M_3\}$ to SD_j.

AKP-4: SD_j computes $\delta_j = D_j \oplus h(X_{SD_j}||b_j||HSID_j)$, $(C_j||r_1) = M_3 \oplus h(Z_j||HSID_j||r_2||X_{SD_j})$, $R_j = Rep(PUF(C_j), \delta_j)$, $TSID_j^{new} = h(r_2||TSID_j)$, and $Auth_{TM-SD}^* = h(TSID_j^{new}||X_{SD_j}||Z_j||R_j||r_1)$. Then, SD_j checks whether $Auth_{TM-SD}^* \stackrel{?}{=} Auth_{TM-SD}$. If it is valid, SD_j computes $M_4 = (R_j||r_2) \oplus h(r_1||TGID_i||TSID_j)$, $SK = h(r_1||r_2||R_j)$, and $Auth_{SD-U} = h(SK||r_1||r_2||R_j)$. Finally, SD_j transmits $\{TSID_j, Auth_{TM-U}, Auth_{SD-U}, M_4\}$ to U_i, and updates $TSID_j$ to $TSID_j^{new}$ in the memory.

AKP-5: U_i computes "$(R_j||r_2) = M_4 \oplus (r_1||TGID_i||TSID_j)$, $Auth_{TM-U} = h(TGID||X_i||R_j||r_1||r_2)$, and checks whether $Auth_{TM-U}^* \stackrel{?}{=} Auth_{TM-U}$. If it is correct, U_i computes $TSID_j^{new} = h(r_2||TSID_j)$, $SK = h(r_1||r_2||R_j)$, and $Auth_{SD-U}^* = h(SK||r_1||r_2||R_j)$, and verifies whether $Auth_{SD-U}^* \stackrel{?}{=} Auth_{SD-U}$". If it is valid, U_i updates $\{TSID_j^{new}\}$ for the next login.

4.4. Biometric and Password Update Process

If the legitimate users want a new BIO_i and PW_i, U_i can handily change their old BIO_i and PW_i [42].

PBU-1: U_i inputs a "ID_i, a old PW_i, and imprints a old BIO_i into U_i".

PBU-2: U_i calculates $\gamma_i = PUF(BIO_i)$, $\beta_i = \beta_i^* \oplus h(ID_i||X_{GD_i}||PW_i)$, $\alpha_i = Rep(\gamma_i, \beta_i)$, $n_i = n_i^* \oplus h(ID_i||\alpha_i||PW_i)$, $RPW_i = h(PW_i||n_i)$, $X_i = X_i^* \oplus h(ID_i||\alpha_i||X_{GD_i}||RPW_i)$, $HID_i = HID_i^* \oplus h(\alpha_i||RPW_i||X_i||X_{GD_i})$, and $C_i^* = h(HID_i||\alpha_i||X_i||X_{GD_i})$. Then, U_i checks whether $C_i^* \stackrel{?}{=} C_i$. If the condition is not valid, U_i aborts this session, otherwise U_i transmits the authentication message to U_i.

PBU-3: After getting the authentication message, U_i inputs a new password PW_i^{new}, and imprints a new biometric BIO_i^{new} to the U_i via a secure channel.

PBU-4: U_i generates a new biometric token α_i^{new}, and the corresponding secret parameter β_i^{new} as (α_i^{new}, β_i^{new})=$Gen(\gamma_i^{new})$. After that, U_i calculates $\beta_i^{'} = \beta_i^{new} \oplus h(ID_i||X_{GD_i}||PW_i^{new})$, $n_i^{'} = n_i \oplus h(ID_i||\alpha_i^{new}||PW_i^{new})$, $RPW_i^{new} = h(PW_i^{new}||n_i)$, $X_i^{'} = X_i \oplus h(ID_i||\alpha_i^{new}||X_{GD_i}||RPW_i^{new})$, $HID_i^{'} = HID_i \oplus h(\alpha_i^{new}||RPW_i||X_i||X_{GD_i})$, and $C_i^{'} =$

$h(HID_i||\alpha_i^{new}||X_i||X_{GD_i})$. Finally, U_i replaces $\{HID_i^{'}, X_i^{'}, n_i^{'}, \beta_i^{'}, C_i^{'}\}$ with $\{HID_i^*, X_i^*, n_i^*, \beta_i^*, C_i\}$ in the memory.

5. Security Analysis

We carry out the informal/formal security analysis, such as "AVISPA implementation and ROR oracle model". We demonstrate that PUF-PSS can prevent various cyber/physical security threats, including "impersonation, session key disclosure, and MITM attacks" and ensure "anonymity, perfect forward secrecy, and mutual authentications".

5.1. Informal Security Analysis

We perform the "informal security analysis to prove the security of PUF-PSS". We prove that PUF-PSS is able to prevent potential security threats and provide "secure anonymity, perfect forward secrecy, and mutual authentication".

5.1.1. Impersonation Attack

We suppose that MA tries to impersonate by intercepting the exchanged messages of each participant over an open channel. However, MA cannot correctly generate the authentication request message $\{TGID_i, M_1, Auth_u\}$, $\{TGID_i, M_1, Auth_u, TSID_j, M_2, Auth_{SD}\}$ and response message $\{Auth_{TM-SD}, AUTH_{TM-U}, M_3\}$, $\{TSID_j, Auth_{SD-U}, Auth_{TM-U}, M_4\}$ because MA does not receive the "random nonces $\{r_1, r_2\}$ and secret credentials $\{X_i, Z_j\}$". Thus, PUF-PSS is resilient to this attack since MA cannot calculate the valid authentication messages of each entity.

5.1.2. Physical Capture Attack

We assume that MA can physically capture any SD_j, and then extract all the secret credentials in the memory of a physically captured SD_j, compromising of the data $\{N_j, M_j, D_j, C_j\}$ from the SD_j's memory. However, there are independent and distinct factors for all deployed SD_j since SID_j and C_j are randomly generated. Thus, the compromised data does not help in computing a session key SK between U_i and an other non-compromised SD_j. Consequently, PUF-PSS is secure to this attack since the output of PUF challenge and response pair $\{(\alpha_i, \beta_i), (C_j, R_j)\}$ depends upon the intrinsic physical variations in the IC chip.

5.1.3. Replay Attack

If MA eavesdrops the transmitted messages over an open channel, MA tries to authenticate with other participants by retransmitting the intercepted messages from the previous session. However, in PUF-PSS, all of the entities check the freshness of the random nonces r_1 and r_2. Moreover, the transmitted messages are protected with secret credentials X_i and Z_j. Hence, PUF-PSS is resilient against replay attacks.

5.1.4. Session Key Disclosure Attack

MA should obtain the "PUF response and random nonces (short-term secrets) $\{r_1, r_2, R_j\}$ and the long-term secrets credentials $\{X_i, Z_j\}$ to generate the correct $SK = h(r_1||r_2||R_j)$". However, MA cannot calculate because $\{X_i, Z_j\}$ is protected with the "shared secret key $\{X_{GD_i}, X_{SD_j}\}$, random number b_j, and PUF challenge α_i" using the hash function. Moreover, MA cannot obtain $\{r_1, r_2, R_j\}$ since MA does not know the "real identity $\{ID_i, SID_j\}$ of U_i and SD_j, the secret credentials $\{X_i, Z_j\}$, and PUF secret parameter R_j". Thus, PUF-PSS is resilient to this attack under the CK model [38] as the presented threat model in Section 3.2.

5.1.5. Offline Password Guessing Attack

We suppose that MA attempts to guess the U_i's password PW_i, and also extract all secret credentials $\{HID_i^*, X_i^*, n_i^*, \beta_i^*, C_i\}$ in MD_i's memory using the differential power analysis. If MA can guess U_i's PW_i, MA can calculate "several equations and the correct credentials with the guessed PW_i". However, MA should know a "unique biometric BIO_i

and a random number n_i of U_i" to calculate the correct credentials and equations. Hence, MA is difficult to correctly guess U_i's PW_i because MA cannot obtain the biometric BIO_i and random number n_i.

5.1.6. MITM Attack

We suppose that if MA can eavesdrop the transmitted messages through an open channel, then this attack may be possible. However, MA is unable to successfully calculate the authentication request and confirmation messages since MA cannot obtain the "random nonces $\{r_1, r_2\}$, PUF secret parameter $\{\beta_i, R_j\}$, biometric BIO_i, real identity $\{ID_i, SID_j\}$". Consequently, PUF-PSS can prevent this attack since MA cannot get the secret credentials of the legal entities.

5.1.7. Stolen Verifier Attack

We assume that MA steals the secret credential stored in S's database and then tries to impersonate the legitimate participant. Even if MA obtains the secret credentials $\{HID_i\}$ for U_i and $\{V_j, (C_j, R_j)\}$ for SD_j stored in database of SP, MA cannot obtain sensitive information and impersonate as legitimate entities. Even if the secret credential $\{HID_i\}$ for U_i is revealed, MA does not obtain the sensitive information without the fresh random nonce r_1, the correct shared secret key X_{GD_i} for U_i and S. Moreover, the secret credential $\{V_j\}$ for SD_j is protected with the secret private key K_i of S and the shared secret key X_{SD_j} by performing XOR and hash functions. PUF challenge/response pairs $\{C_j, R_j)\}$ for SD_j are computationally difficult to compromise the PUF secret value because the output of PUF relies on the unique physical characteristics. Hence, PUF-PSS is resilient to this attack since MA cannot impersonate the legitimate participant because MA does not receive the sensitive data for SD_j and U_i/GW.

5.1.8. Ephemeral Secret Leakage (ESL) Attack

According to Section 3.2, we assume that MA can compromise the session states and secret credentials under the CK adversary model. If the short-term secrets $\{r_1, r_2\}$ are revealed, an SK is protected since MA cannot obtain the sensitive information, such as the random nonces $\{n_i, b_j\}$ and the real identities $\{ID_i, SID_j\}$. On the other hand, if the long-term secrets $\{X_i, Z_j\}$ are compromised, an SK is still protected since MA does not obtain the shared secret keys $\{X_{GD_i}, X_{SD_j}\}$, the biometric secret value β_i, and the PUF secret value R_j. Thus, PUF-PSS resists an ESL attack based on the CK model [38].

5.1.9. Perfect Forward Secrecy

We assume that MA can obtain TMIS server S's secret key K_i. After that, MA attempts to compute a session key $SK = h(r_1||r_2||R_j)$ between U_i and SD_j. However, MA cannot compute an SK because MA does not obtain the random nonces $\{r_1, r_2\}$ and the PUF value R_j. Therefore, PUF-PSS scheme ensures perfect forward secrecy.

5.1.10. Mutual Authentication

During the authentication and key agreement process, all of the participants successfully perform mutual authentication. After getting the messages $\{TGID_i, M_1, Auth_u\}$ from the U_i, S checks whether $Auth_u^* \stackrel{?}{=} Auth_u$. If it is correct, S authenticates U_i. After obtaining the messages $\{TSID_j, M_2, Auth_{SD}\}$ from SD_j, S verifies whether $Auth_{SD}^* \stackrel{?}{=} Auth_{SD}$. If the condition is correct, S authenticates SD_j. After getting the messages $\{Auth_{TM-SD}, C_j, M_4\}$ from S, SD_j checks whether $Auth_{TM-SD}^* \stackrel{?}{=} Auth_{TM-SD}$. If it is valid, SD_j authenticates S. After obtaining the message $\{Auth_{TM-U}, Auth_{SD-U}, M_3, M_5\}$ from SD_j and S, U_i verifies whether $Auth_{TM-U}^* \stackrel{?}{=} Auth_{TM-U}$ and $Auth_{SD-U}^* \stackrel{?}{=} Auth_{SD-U}$. If it is correct, U_i authenticates SD_j and S and establishes an SK. Consequently, all of the participants are "mutually

authenticated because MA cannot calculate the authentication request and confirmation messages successfully".

5.1.11. Anonymity

According to Section 3.2, MA can extract secret parameters stored in MD_i and intercept the transmitted messages in each session. However, MA cannot retrieve the "real identity $\{ID_i, SID_j\}$ of U_i and SD_j because the transmitted messages are masked with random nonce $\{r_1, r_2\}$, secret credentials $\{X_i, Z_j\}$, biometric $\{BIO_i\}$ and shared secret key $\{X_{GD_i}, X_{SD_j}\}$" using the PUF function, hash function, and XOR operation. Therefore, PUF-PSS guarantees the anonymity of U_i and SD_j.

5.2. Formal Security Analysis Using BAN Logic

We demonstrate that PUF-PSS guarantees secure mutual authentication among U_i, SD_j, and S by performing BAN logic [10]. We introduce the symbols in Table 2 and also define rules, idealized forms, security goals, and assumptions for BAN logic.

Table 2. BAN logic symbols.

Symbol	Description
ζ, φ	Principals
X, Y	Statements
SK	Session key
$\zeta \mid \equiv X$	ζ **believes** X
$\zeta \mid \sim X$	ζ once **said** X
$\zeta \Rightarrow X$	ζ **controls** X
$\zeta \lhd X$	ζ **receives** X
$\#X$	X is **fresh**
$\{X\}_K$	X is **encrypted** with K
$\zeta \overset{K}{\leftrightarrow} \varphi$	ζ and φ have shared secret key K

1. Message meaning rule (MMR) :

$$\frac{\zeta \mid \equiv \zeta \overset{K}{\leftrightarrow} \varphi, \quad \zeta \lhd \{X\}_K}{\zeta \mid \equiv \varphi \mid \sim X}$$

2. Nonce verification rule (NVR) :

$$\frac{\zeta \mid \equiv \#(X), \quad \zeta \mid \equiv \varphi \mid \sim X}{\zeta \mid \equiv \varphi \mid \equiv X}$$

3. Jurisdiction rule (JR) :

$$\frac{\zeta \mid \equiv \varphi \Rightarrow X, \quad \zeta \mid \equiv \varphi \mid \equiv X}{\zeta \mid \equiv X}$$

4. Freshness rule (FR) :

$$\frac{\zeta \mid \equiv \#(X)}{\zeta \mid \equiv \#(X, Y)}$$

5. Belief rule (BR) :

$$\frac{\zeta \mid \equiv (X, Y)}{\zeta \mid \equiv X}$$

5.2.1. Security Goals

We present the security goals of PUF-PSS to prove the BAN logic.

Goal 1: $U_i| \equiv U_i \overset{SK}{\longleftrightarrow} SD_j$

Goal 2: $SD_j| \equiv U_i \overset{SK}{\longleftrightarrow} SD_j$

Goal 3: $U_i| \equiv SD_j| \equiv U_i \overset{SK}{\longleftrightarrow} SD_j$

Goal 4: $SD_j| \equiv U_i| \equiv U_i \overset{SK}{\longleftrightarrow} SD_j$

5.2.2. Idealized Forms

The idealized forms of the messages in PUF-PSS are as follows.

MIF_1**:** $U_i \rightarrow SD_j : \{r_1, HID_i, TGID_i, X_i\}_{X_{GD_i}}$

MIF_2**:** $SD_j \rightarrow S : \{HID_i, TGID_i, r_1, X_i, r_2, SID_j, TSID_j, Z_j\}_{X_{SD_j}}$

MIF_3**:** $S \rightarrow SD_j : \{HID_i, SID_j, r_1, r_2, R_j\}_{X_{SD_j}}$

MIF_4**:** $SD_j \rightarrow U_i : \{(U_i \overset{SK}{\longleftrightarrow} SD_j), TGID_i, TSID_j, r_2, X_i\}_{X_{GD_i}}$

5.2.3. Assumptions

We present the assumptions of PUF-PSS as follows.

AS_1**:** $SD_j| \equiv (U_i \overset{X_{GD_i}}{\longleftrightarrow} SD_j)$

AS_2**:** $SD_j| \equiv \#(r_1)$

AS_3**:** $S| \equiv (S \overset{X_{SD_j}}{\longleftrightarrow} SD_j)$

AS_4**:** $S| \equiv \#(r_1, r_2)$

AS_5**:** $SD_j| \equiv (S \overset{X_{SD_j}}{\longleftrightarrow} SD_j)$

AS_6**:** $SD_j| \equiv \#(r_2)$

AS_7**:** $U_i| \equiv (U_i \overset{X_{GD_i}}{\longleftrightarrow} SD_j)$

AS_8**:** $U_i| \equiv \#(r_1)$

AS_9**:** $U_i| \equiv SD_j \mapsto (U_i \overset{SK}{\longleftrightarrow} SD_j)$

AS_{10}**:** $SD_j| \equiv U_i \mapsto (U_i \overset{SK}{\longleftrightarrow} SD_j)$

5.2.4. BAN Logic Proof

We then present the BAN logic proof as follows.

Step 1: Based on MIF_1, we obtain

$$(P_1) : SD_j \lhd \{r_1, HID_i, TGID_i, X_i\}_{X_{GD_i}}$$

Step 2: Using P_1 and AS_1 with the MMR, we obtain

$$(P_2) : SD| \equiv U| \sim \{r_1, HID_i, TGID_i, X_i\}_{X_{GD_i}}$$

Step 3: Based on the P_2 and AS_2 with the FR, we obtain

$$(P_3) : SD| \equiv \#\{r_1, HID_i, TGID_i, X_i\}_{X_{GD_i}}$$

Step 4: Using P_2 and P_3 with the NVR, we obtain

$$(P_4) : SD| \equiv U| \equiv \{r_1, HID_i, TGID_i, X_i\}_{X_{GD_i}}$$

Step 5: Based on the P_4 with the BR, we obtain

$$(P_5) : SD| \equiv U| \equiv (r_1)$$

Step 6: According to MIF_2, we obtain

$$(P_6) : S \lhd \{HID_i, TGID_i, r_1, X_i, r_2, SID_j, TSID_j, Z_j\}_{X_{SD_j}}$$

Step 7: Using P_6 and AS_3 with the MMR, we obtain

$$(P_7) : S| \equiv SD_j| \sim \{HID_i, TGID_i, r_1, X_i, r_2, SID_j, TSID_j, Z_j\}_{X_{SD_j}}$$

Step 8: Based on the P_7 and AS_4 with the FR, we obtain

$$(P_8) : S| \equiv \#\{HID_i, TGID_i, r_1, X_i, r_2, SID_j, TSID_j, Z_j\}_{X_{SD_j}}$$

Step 9: Using P_7 and P_8 with the NVR, we obtain

$$(P_9) : S| \equiv SD_j| \equiv \{HID_i, TGID_i, r_1, X_i, r_2, SID_j, TSID_j, Z_j\}_{X_{SD_j}}$$

Step 10: According to MIF_3, we obtain

$$(P_{10}) : SD_j \lhd \{HID_i, SID_j, r_1, r_2, R_j\}_{X_{SD_j}}$$

Step 11: Using P_{10} and AS_5 with the MMR, we obtain

$$(P_{11}) : SD_j| \equiv S| \sim \{HID_i, SID_j, r_1, r_2, R_j\}_{X_{SD_j}}$$

Step 12: Based on the P_{11} and AS_6 with the FR, we obtain

$$(P_{12}) : SD_j| \equiv \#\{HID_i, SID_j, r_1, r_2, R_j\}_{X_{SD_j}}$$

Step 13: Using P_{11} and P_{12} with the NVR, we obtain

$$(P_{13}) : SD_j| \equiv S| \equiv \{HID_i, SID_j, r_1, r_2, R_j\}_{X_{SD_j}}$$

Step 14: According to MIF_4, we obtain

$$(P_{14}) : U_i \lhd \{(U_i \overset{SK}{\longleftrightarrow} SD_j), TGID_i, TSID_j, r_2, X_i\}_{X_{GD_i}}$$

Step 15: Using P_{14} and AS_7 with the MMR, we obtain

$$(P_{15}) : U_i| \equiv SD_j| \sim \{(U_i \overset{SK}{\longleftrightarrow} SD_j), TGID_i, TSID_j, r_2, X_i\}_{X_{GD_i}}$$

Step 16: Based on the P_{15} and AS_8 with the FR, we obtain

$$(P_{16}) : U_i| \equiv \#\{(U_i \overset{SK}{\longleftrightarrow} SD_j), TGID_i, TSID_j, r_2, X_i\}_{X_{GD_i}}$$

Step 17: Using P_{15} and P_{16} with the NVR, we obtain

$$(P_{17}) : U_i| \equiv SD_j| \equiv \{(U_i \overset{SK}{\longleftrightarrow} SD_j), TGID_i, TSID_j, r_2, X_i\}_{X_{GD_i}}$$

Step 18: Based on the P_{17} with the BR, we obtain

$$(P_{18}) : U_i| \equiv SD_j| \equiv (U_i \overset{SK}{\longleftrightarrow} SD_j) \qquad \textbf{(Goal 3)}$$

Step 19: Using P_{18} and AS_9 with the JR, we obtain

$$(P_{19}) : U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} SD_j) \qquad \textbf{(Goal 1)}$$

Step 20: Because of $SK = h(r_1||r_2||R_j)$ from P_5, P_9, P_{13} and P_{17}, we obtain

$$(P_{20}) : SD_j| \equiv U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} SD_j) \qquad \textbf{(Goal 4)}$$

Step 21: Based on the P_{19} and AS_{10} with the JR, we obtain

$$(P_{21}) : SD_j| \equiv (U_i \overset{SK}{\longleftrightarrow} SD_j) \qquad \textbf{(Goal 2)}$$

Consequently, we prove that U_i, SD_j, and S are mutually authenticated because they achieve security goals 1–4.

5.3. Formal Security Analysis Using ROR Oracle Model

We evaluate a session key (SK) security of PUF-PSS from MA under the ROR oracle model [7]. We define the necessary queries for the ROR model [7] as follows.

In PUF-PSS, there are three entities: the users $P_U^{t_1}$, the sensing devices $P_{SD}^{t_2}$, and the TMIS server $P_S^{t_3}$, where $P_U^{t_1}$, $P_{SD}^{t_2}$, and $P_S^{t_3}$ are instances t_1^{th} of U_i, t_2^{th} of SD_j, and t_3^{th} of S, respectively. Table 3 shows the necessary queries, including "$Execute()$, $CorruptMD()$, $Send()$, $Test()$ and $Reveal()$ to perform security analysis". Furthermore, we use a "hash function $Hash$, and a PUF function PUF as a random oracle". We utilize Zipf's law [43] to prove the SK security of PUF-PSS.

Table 3. Queries and purposes.

Queries	Purpose
$Execute(\mathcal{P}_U^{t_1}, \mathcal{P}_{SD}^{t_2}, \mathcal{P}_S^{t_3})$	Based on $Execute()$, MA performs the passive/active attacks by eavesdropping the exchanged messages between each entity over a insecure channel.
$CorruptMD(\mathcal{P}_U^{t_1})$	This query indicates as the mobile device stolen attacks, where MA can extract the secret credentials stored in MD.
$CorruptSD(\mathcal{P}_{SD}^{t_2})$	This query indicates as the physical capture attacks, where MA can obtain the secret parameters stored in SD.
$Send(\mathcal{P}^t, Msg)$	Based on this query, MA can transmit the message Msg to the P^t, and obtain the response message accordingly.
$Reveal(\mathcal{P}^t)$	Under the this query, MA reveals a SK generated between $P_U^{t_1}$ and $P_{SD}^{t_2}$.
$Test(\mathcal{P}^t)$	An unbiased coin c is tossed prior to game start. If MA gets the $c = 1$ under the $Test()$, it indicates a SK between $P_U^{t_1}$ and $P_{SD}^{t_2}$ is fresh. If MA obtains the $c = 0$, it indicate a SK is not fresh; otherwise, MA obtains a null value ($\perp$).

Theorem 1. *$Adv_{MA}^{PUF-PSS}$ presents the "advantages of MA in violating SK security for PUF-PSS". Hence, we derive the following:*

$$Adv_{MA}^{PUF-PSS} \leq \frac{q_h^2}{|Hash|} + \frac{q_P^2}{|PUF|} + 2\{C \cdot q_{send}^s, \frac{q_s}{2^{l_1}}, \frac{q_s}{2^{l_2}}\}$$

q_P, q_h, q_{send}, and Hash are "the range space of PUF $PUF(\cdot)$, the range space of hash function $h(\cdot)$, $Send(\cdot)$ query, and the number of Hash query", respectively. In addition, l_n, s, l_m, and C are the Zipf's credentials [43].

Proof. We introduce the five games GM_i ($i \in [0,4]$). We present that $Adv_{MA,GM_i}^{PUF-PSS}$ is the "probability of MA winning the GM_i". □

Game GM_0: "GM_0 is considered as an actual attack executed by MA in PUF-PSS. The bit c is randomly selected prior to the beginning of GM_0". Based on GM_0, the result is as follows:

$$Adv_{MA}^{PUF-PSS} = |2 \cdot Adv_{MA,GM_0}^{PUF-PSS} - 1| \quad (1)$$

Game GM_1: "GM_1 presents that MA executes an eavesdropping attack using $Execute()$ query. MA perform $Test()$ and $Reveal()$ queries to reveal SK. The output of the $Test()$ and $Reveal()$ queries decide if MA gets the secret credentials and $SK = h(r_1||r_2||R_j)$. To reveal SK, MA needs the PUF value R_j and random nonces $\{r_1, r_2\}$. Thus, MA's probability of winning GM_1 by eavesdropping on the exchanged messages does not increase". Based on GM_1, the result is as presented below:

$$Adv_{MA,GM_1}^{PUF-PSS} = Adv_{MA,GM_0}^{PUF-PSS} \quad (2)$$

Game GM_2: GM_2 is considered as the "passive/active attacks by using $Send()$ and $Hash$ queries". MA can intercept the messages $\{TGID_i, M_1, Auth_u\}$, $\{TGID_i, M_1, Auth_u, TSID_j M_2, Auth_{SD}\}$, $\{Auth_{TM-SD}, Auth_{M-U}, M_3\}$, and $\{TSID_j, Auth_{SD-U}, Auth_{TM-U}, M_4\}$ during the AKA process. All of the messages are not compromised by MA since it is protected by using $h(\cdot)$ with the random nonces r_1 and r_2. Based on the birthday paradox [44], the GM_2's result is as follows:

$$|Adv_{MA,GM_2}^{PUF-PSS} - Adv_{MA,GM_1}^{PUF-PSS}| \leq \frac{q_h^2}{2|Hash|} \quad (3)$$

Game GM_3: GM_3 is an "extended game to GM_2 which the simulation of PUF query is included in this game". By utilizing an analogous argument presented in GM_2, this game's results is presented below:

$$|Adv_{MA,GM_3}^{PUF-PSS} - Adv_{MA,GM_2}^{PUF-PSS}| \leq \frac{q_P^2}{2|PUF|} \quad (4)$$

Game GM_4: GM_4 is modeled on the simulation of the $CorruptMD()$ and $CourruptSD()$ qeries. MA is able to extract the secret parameters $\{HID_i^*, X_i^*, n_i^*, \beta_i^*, C_i\}$ in MD memory by performing the differential power analysis. Note that $HID_i^* = HID_i \oplus h(\alpha_i||RPW_i||X_i||X_{GD_i})$, $X_i^* = X_i \oplus h(ID_i||\alpha_i||X_{GD_i}||RPW_i)$, $n_i^* = n_i \oplus h(ID_i||\alpha_i||PW_i)$, $\beta_i^* = \beta \oplus h(ID_i||X_{GD_i}||PW_i)$, and $C_i = h(HID_i||\alpha_i||X_i||X_{GD_i})$. In addition, MA can obtain the secret credentials $\{C_j, N_j, M_j, D_j\}$ in SD memory by performing physical capture attacks. Note that the PUF random challenge set C_j, $N_j = (HSID_j||b_j) \oplus h(X_{SD_j}||SID_j||TSID_j)$, $M_j = Z_j \oplus h(HSID_j||X_{SD_j}||b_j)$, and $D_j = \delta_j \oplus h(X_{SD_j}||b_j||HSID_j)$. However, this game is computationally infeasible for MA to compromise PW_i over the $Send()$ query without the ID_i, n_i, and α_i. Moreover, MA cannot distinguish the biometric and PUF value since the "probability of guessing the biometric credential of l_1 bits and the PUF secret value of l_2 by MA

is $\frac{1}{2^{l_1}}$ and $\frac{1}{2^{l_2}}$". Consequently, GM_3 and GM_4 are "indistinguishable if the off-line biometric or password guessing attacks are not implemented". The GM_4's result is as follows:

$$|Adv_{MA,GM_4}^{PUF-PSS} - Adv_{MA,GM_3}^{PUF-PSS}| \leq \{C \cdot q_{send}^{s}, \frac{q_s}{2^{l_b}}\} \tag{5}$$

After $GM_0 - GM_4$ are successfully executed, MA tries to guess the "bit c to win the games by performing $Test()$ query". Hence, we obtain the following:

$$Adv_{MA,GM_4}^{PUF-PSS} = \frac{1}{2} \tag{6}$$

Combining Formulas (1), (2) and (6), we obtain the following:

$$\begin{aligned} \frac{1}{2}Adv_{MA}^{PUF-PSS} &= |Adv_{MA,GM_0}^{PUF-PSS} - \frac{1}{2}| \\ &= |Adv_{MA,GM_1}^{PUF-PSS} - \frac{1}{2}| \\ &= |Adv_{MA,GM_1}^{PUF-PSS} - Adv_{MA,GM_4}^{PUF-PSS}| \end{aligned} \tag{7}$$

Based on the "triangular inequality with the Formulas (3)–(5) and (7)", we obtain the following:

$$\begin{aligned} \frac{1}{2}Adv_{MA}^{PUF-PSS} &= |Adv_{MA,GM_1}^{PUF-PSS} - Adv_{MA,GM_4}^{PUF-PSS}| \\ &\leq |Adv_{MA,GM_1}^{PUF-PSS} - Adv_{MA,GM_3}^{PUF-PSS}| \\ &\quad + |Adv_{MA,GM_3}^{PUF-PSS} - Adv_{MA,GM_4}^{PUF-PSS}| \\ &\leq |Adv_{MA,GM_1}^{PUF-PSS} - Adv_{MA,GM_2}^{PUF-PSS}| \\ &\quad + |Adv_{MA,GM_2}^{PUF-PSS} - Adv_{MA,GM_3}^{PUF-PSS}| \\ &\quad + |Adv_{MA,GM_3}^{PUF-PSS} - Adv_{MA,GM_4}^{PUF-PSS}| \\ &\leq \frac{q_h^2}{2|Hash|} + \frac{q_P^2}{2|PUF|} + \{C \cdot q_{send}^{s}, \frac{q_s}{2^{l_1}}, \frac{q_s}{2^{l_2}}\}. \end{aligned} \tag{8}$$

Finally, by multiplying both sides of Equation (8) by a factor of 2, we obtain the following: $Adv_{MA}^{PUF-PSS} \leq \frac{q_h^2}{|Hash|} + \frac{q_P^2}{|PUF|} + 2\{C \cdot q_{send}^{s}, \frac{q_s}{2^{l_1}}, \frac{q_s}{2^{l_2}}\}$

5.4. Formal Security Analysis Using AVISPA Simulation

AVISPA is a "formal security verification simulation that demonstrates whether the cryptographic protocol is resilient against various security threats such as MITM and replay attacks. AVISPA simulation is implemented by utilizing High-Level Protocol Specification Language (HLPSL) [45] to generate input format (IF) of the backends such as On-the-Fly Model Checker (OFMC), Constraint Logic-based Attack Searcher (CL-AtSE), Tree Automata based on Automatic Approximations for Analysis of Security Protocol (TA4SP), and SAT-based Model Checker (SATMC)".

To evaluate the security of PUF-PSS, we first "express utilizing a rule-oriented HLPSL. The various specification roles for the U/GW, SD, and S, and for the mandatory roles for the sessions, environments and security goals are implemented in HLPSL for PUF-PSS. Since XOR operation is not provided for the SATMC and TA4SP backends, AVISPA implementation results for these backends are not included".

Under the HLPSL, we simulated "PUF-PSS using the Security Protocol ANimator (SPAN) [46] for AVISPA. The simulation result for MA utilizing SPAN is shown in Figure 2. Furthermore, the implementation results by performing CL-AtSe and OFMC back-ends are as shown in Figure 3". Consequently, we demonstrate that PUF-PSS is resistant to the cyber security attacks.

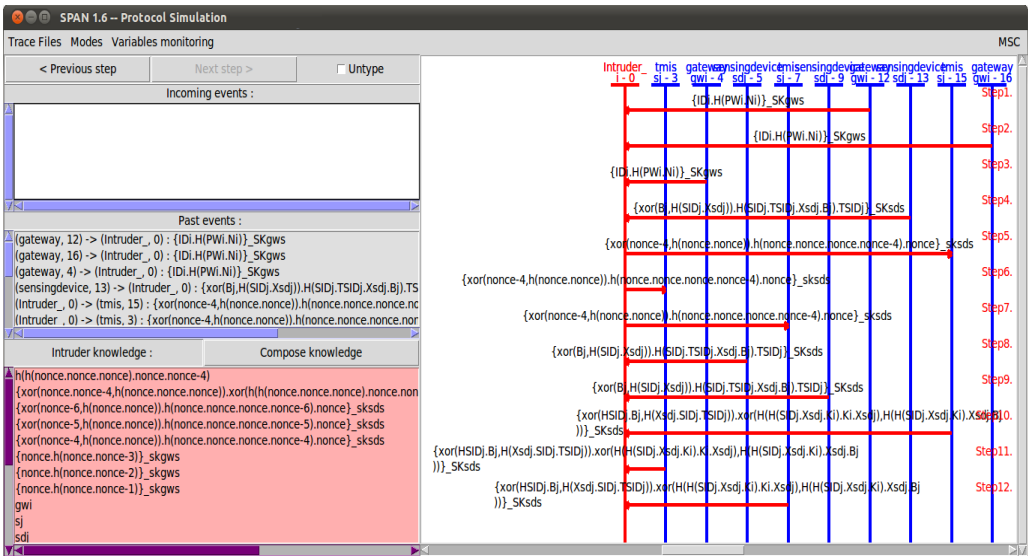

Figure 2. AVISPA result based on SPAN.

SUMMARY **SAFE** DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/span/span/testsuite/results/sjyu.if GOAL As_specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00s searchTime: 1.21s visitedNodes: 1168 nodes depth: 9 plies	SUMMARY **SAFE** DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/span/span/testsuite/results/sjyu.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 15 states Reachable : 15 states Translation: 0.13 seconds Computation: 0.02 seconds

Figure 3. AVISPA results based on CL-AtSe and OFMC.

6. Test Bed Experiments using MIRACL

We present the test bed experiments to estimate the computational time required for essential cryptographic operations utilized in PUF-PSS and previous schemes using the broadly utilized MIRACL [9]. In the following, we utilize two scenarios to estimate the computational time of the cryptographic operations. We denote "T_{bp}, T_{ecpm}, T_h, and T_{sed} to estimate the execution times (in milliseconds) required for a bilinear pairing, an elliptic curve scalar point multiplication, a hash function (for example, Secure Hash Algorithm (SHA-256) [47]), and a symmetric key encryption/decryption (for example, Advanced Encryption Standard (AES) [48])", respectively.

- **Scenario I.** In this case, we have modeled a desktop server setting as follows: "Model: Desktop, CPU Architecture: 64 bits, Processor: Intel Core i5-10400 @2.90 GHz, Six-core, OS: Ubuntu 18.04.4 LTS with 16 GB memory. Each primitive has run for 10,000 times. The maximum and minimum time in milliseconds are observed for each primitive. At the same, the average time (in milliseconds) is also measured out of these 100 runs". The experimental results under sever setting are tabulated in Table 4.

Table 4. Execution time for a server.

Operation	Max. Time (ms)	Min. Time (ms)	Average Time (ms)
T_{bp}	5.157	2.940	3.002
T_{ecpm}	2.737	0.472	0.522
T_h	0.149	0.024	0.055
T_{sed}	0.002	0.001	0.001

- **Scenario II.** In this case, we have modeled a "Raspberry PI setting as follows: Model: Raspberry PI 4B (2019), CPU Architecture: 64 bit, Processor: 1.5 GHz Quad-core, OS: Ubuntu 20.04.2 LTS with 8 GB memory. Similar to Scenario I, each primitive has also run for 10,000 times and then measured the average, minimum and maximum time in milliseconds for the primitives". The experimental results based on a Raspberry PI 4 are presented in Table 5.

Table 5. Execution time for a Raspberry PI 4.

Operation	Max. Time (ms)	Min. Time (ms)	Average Time (ms)
T_{bp}	18.722	18.132	18.294
T_{ecpm}	2.920	2.766	2.848
T_h	0.643	0.274	0.309
T_{sed}	0.021	0.011	0.012

7. Comparative Analysis

We demonstrate the comparative analysis for the performance of PUF-PSS with previous AKA schemes for TMIS [20–22] with regard to "communication costs", "computation costs", and "security properties".

7.1. Communication Costs

This section evaluates the communication cost comparison analysis of our AKA scheme and the related schemes [20–22]. According to [27], we assume that the lengths (bits) for the "timestamp, identity, random nonce, hash function, and ECC are 32, 128, 128, 256, and 320 bits", respectively. During the AKA process of PUF-PSS, the transmitted messages "$\{TGID_i, M_1, Auth_u\}$, $\{TGID_i, M_1, Auth_u, TSID_j, M_2, Auth_{SD}\}$, $\{Auth_{TM-S}, Auth_{TM-U}, M_3\}$, and $\{TSID_j, Auth_{SD-U}, Auth_{TM-U}, M_4\}$ require (128 + 256 + 256 = 640 bits), (128 + 256 + 256 + 128 + 256 + 256 = 1,280 bits), (256 + 256 + 256 = 768 bits), and (128 + 256 + 256 + 256 = 896 bits)", respectively. We show the analysis result for communication overhead comparison in Figure 4 and Table 6. Although PUF-PSS has a somewhat greater communication overhead than Wazid et al.'s scheme [21], it offers more efficient communication costs compared with the existing related schemes [20–22]. Therefore, PUF-PSS is suitable for IoMT-based TMIS environments.

Table 6. A communication cost summary.

Scheme	1st Message	2nd Message	3rd Message	4th Message	Total Costs
Sharma and Karla [20]	928 bits	1472 bits	1056 bits	832 bits	4288 bits
Wazid et al. [21]	672 bits	672 bits	800 bits	1088 bits	3232 bits
Zhou et al. [22]	1152 bits	2304 bits	1536 bits	768 bits	4760 bits
Our scheme	640 bits	1280 bits	768 bits	896 bits	3584 bits

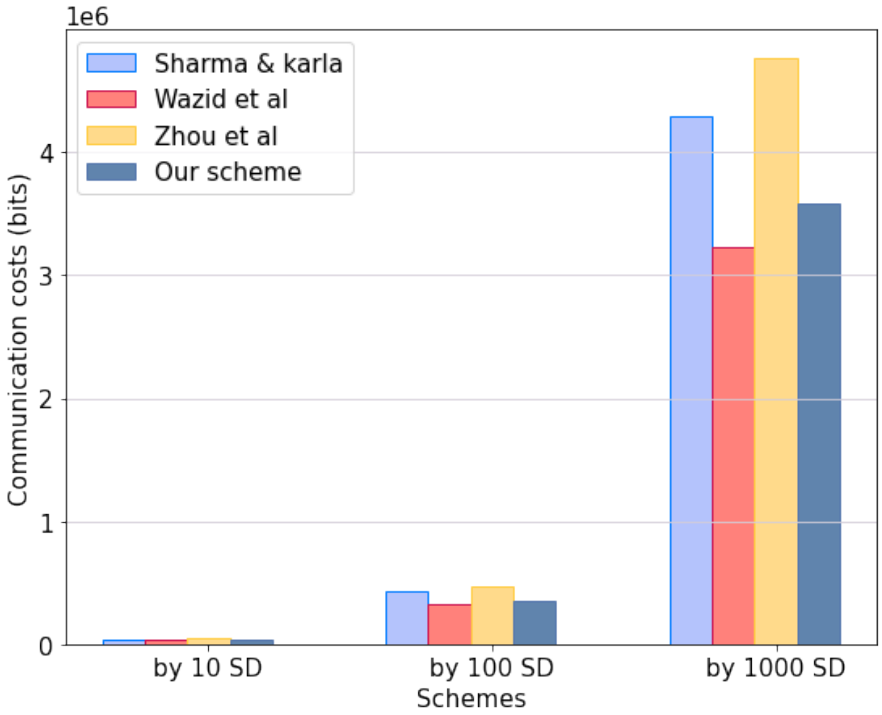

Figure 4. Communication cost comparison in sensing devices [20–22].

7.2. Computation Costs

We perform the computation cost comparison analysis of PUF-PSS with the existing schemes [20–22] during the AKA process. We use the "test-bed experimental results for a server setting and the Raspberry PI 4 setting, which are measured for the execution time needed for various cryptographic operations in Section 6". We utilize the "experimental results for the average execution time needed for cryptographic operations under S environment is considered with a server setting (as shown in Table 4)". In this scenario, we have presented "$T_{bp} \approx 3.002$ ms, $T_{ecpm} \approx 0.522$ ms, $T_h \approx 0.055$ ms and $T_{sed} \approx 0.001$ ms". On the other side, we have used the "experimental results for the average execution time needed for cryptographic operations under MU_i or SD_j environment with a Raspberry PI 4 setting (as shown in Table 5)". Under this scenario, we have presented "$T_{bp} \approx 18.294$ ms, $T_{ecpm} \approx 2.848$ ms, $T_h \approx 0.309$ ms and $T_{sed} \approx 0.012$ ms". Finally, we show the performance results for the computation overhead comparison in Figure 5 and Table 7. PUF-PSS better offers the necessary security requirements and features, and also provides a similar computational costs compared with previous schemes [20–22]. Hence, PUF-PSS is applicable for IoMT-based TMIS.

Table 7. A computation cost summary.

Scheme	User	Sensing Device	TMIS Server	Total Costs
Sharma and Karla [20]	$11T_h \approx 3.399$ ms	$7T_h \approx 2.163$ ms	$12T_h \approx 0.66$ ms	$30T_h \approx 6.222$ ms
Wazid et al. [21]	$9T_h \approx 2.781$ ms	$12T_h \approx 3.708$ ms	$7T_h \approx 0.385$ ms	$28T_h \approx 6.874$ ms
Zhou et al. [22]	$10T_h \approx 3.09$ ms	$7T_h \approx 2.163$ ms	$15T_h \approx 0.825$ ms	$32T_h \approx 6.078$ ms
Our scheme	$12T_h \approx 3.708$ ms	$9T_h \approx 2.781$ ms	$9T_h \approx 0.495$ ms	$30T_h \approx 6.984$ ms

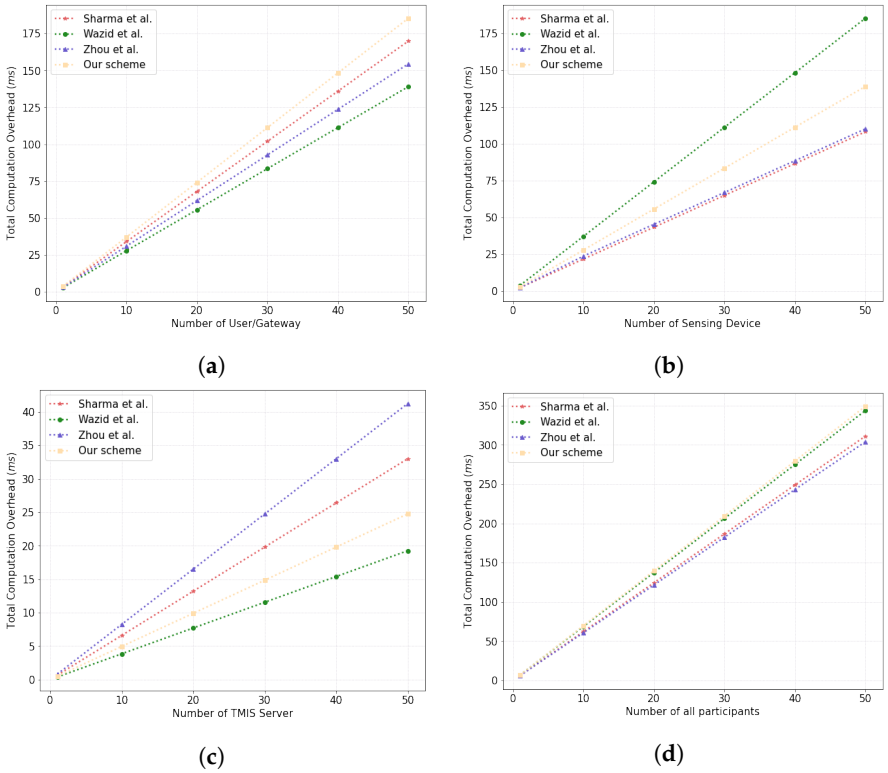

Figure 5. Computation overhead comparison of (**a**) users, (**b**) sensing devices, (**c**) TMIS servers, and (**d**) all participants [20–22].

7.3. Security Properties

We perform the security functionalities comparison analysis of PUF-PSS with the existing related schemes [20–22]. Referring to Table 8, the related schemes are fragile to potential security threats and cannot withstand anonymity and mutual authentication. In contrast, we prove that PUF-PSS is resilient against potential security threats, and guarantees anonymity and authentication. Consequently, PUF-PSS provides many essential security properties compared with the existing related schemes [20–22].

Table 8. A comparative summary: security properties.

Properties	Sharma and Karla [20]	Wazid et al. [21]	Zhou et al. [22]	Ours
SPN1	√	√	√	√
SPN2	√	√	√	√
SPN3	×	√	√	√
SPN4	√	×	×	√
SPN5	√	√	√	√
SPN6	√	√	√	√
SPN7	√	√	√	√
SPN8	×	√	√	√
SPN9	√	×	×	√
SPN10	√	√	√	√
SPN11	√	√	√	√
SPN12	×	×	√	√
SPN13	×	×	×	√

SPN1: "Mobile device stolen attack"; SPN2: "Impersonation attack"; SPN3: "Stolen verifier attack"; SPN4: "Off-line password guessing attack"; SPN5: "Session key disclosure attack"; SPN6: "Replay attack"; SPN7: "MITM attack"; SPN8: "Physical capture attack"; SPN9: "Privileged insider attack"; SPN10: "Perfect forward secrecy"; SPN11: "Mutual authentication"; SPN12: "User anonymity"; SPN13: "Formal (mathematical) analysis".

8. Conclusions and Future Works

We prove that the previous AKA schemes for IoMT-based TMIS suffer from potential security- and privacy-related issues because they are fragile to passive/active security threats, such as impersonation, physical capture, and stolen verifier attacks. We design a physically secure privacy-preserving scheme using PUF for IoMT-based TMIS to improve the security flaws of the previous AKA scheme. We demonstrate that PUF-PSS prevents potential security attacks and provides the essential security properties. We then show that PUF-PSS is secure against various security threats by using well-known formal security analysesm such as AVISPA implementation and the ROR oracle model. Furthermore, we present the test bed experiments of our AKA scheme on the MIRACL-based Raspberry PI 4. Furthermore, PUF-PSS ensures efficient computational and communication costs and also offers superior security functionality compared with previous schemes. Consequently, PUF-PSS is suitable for IoMT-based TMIS because it is more secure compared with previous schemes for IoMT-based TMIS.

In future works, we have planned to develop a new architecture and protocol using blockchain technology to integrate PUF-PSS into a more complete IoMT-enabled TMIS.

Author Contributions: Conceptualization, S.Y; methodology, S.Y.; validation, S.Y.; formal analysis, S.Y.; writing—original draft preparation, S.Y.; writing—review and editing, K.P.; supervision, K.P.; project administration, K.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. World Health Organization (WHO). Global Patient Safety Action Plan 2021–2030. Towards Zero Patients Harm in Healthcare. 2021. Available online: https://www.who.int/docs/default-source/patient-safety/1st-draft-global-patient-safety-action-plan-august-2020.pdf?sfvrsn=9b1552d2_4 (accessed on 15 February 2022).
2. Xiong, H.; Tao, J.; Yuan, C. Enabling Telecare Medical Information Systems With Strong Authentication and Anonymity. *IEEE Access* **2017**, *5*, 5648–5661.
3. Lara, E.; Aguilar, L.; Garcla, J.A. Lightweight Authentication Protocol Using Self-Certified Public Keys for Wireless Body Area Networks in Health-Care Applications. *IEEE Access* **2021**, *9*, 79196–79213.
4. Ermakova, T.; Fabian, B.; Kornacka, M.; Thiebes, S.; Sunyaev, A. Security and Privacy Requirements for Cloud Computing in Healthcare: Elicitation and Prioritization from a Patient Perspective. *ACM Trans. Manag. Inf. Syst.* **2020**, *11*, 1–29.
5. Das, A.K. A Secure User Anonymity Preserving Three-Factor Remote User Authentication Scheme for the Telecare Medicine Information Systems. *J. Med. Syst.* **2015**, *39*, 1–20.
6. Qiu, S.; Xu, G.; Ahmad, H.; Wang, L. A Robust Mutual Authentication Scheme Based on Elliptic Curve Cryptography for Telecare Medical Information Systems. *IEEE Access* **2017**, *6*, 7452–7463.
7. Abdalla, M.; Fouque, P.A.; Pointcheval, D. Password-based authentication key exchange in the three-party setting. In *Public Key Cryptography*; Springer: Les Diablerets, Switzerland, 2005; pp. 65–84.
8. AVISPA. Automated Validation of Internet Security Protocols and Applications. 2001. Available online: http://www.avispa-project.org/ (accessed on 16 March 2021).
9. MIRACL Cryptographic SDK. Multiprecision Integer and Rational Arithmetic Cryptographic Library. 2019. Available online: https://github.com/miracl/MIRACL (accessed on 15 April 2021).
10. Burrows, M.; Abadi, M.; Needham, R. A Logic of Authentication. *ACM Trans. Comput. Syst.* **1990**, *183*, 18–36.
11. Giri, D.; Maitra, T.; Amin, R.; Srivastava, P.D. An Efficient and Robust RSA-Based Remote User Authentication for Telecare Medical Information Systems. *J. Med. Syst.* **2015**, *39*, 1–9.
12. Chatterjee, S.; Roy, S.; Das, A.K.; Chattopadhyay, S.; Kumar, N.; Reddy, A.G.; Park, K.S.; Park, Y.H. On the Design of Fine Grained Access Control With User Authentication Scheme for Telecare Medicine Information Systems. *IEEE Access* **2017**, *5*, 2169–3536.
13. Salem, F.M.; Amin, R. A Privacy-Preserving RFID Authentication Protocol Based on El-Gamal Cryptosystem for Secure TMIS. *Inf. Sci.* **2020**, *527*, 382–393.
14. Amin, R.; Islam, S.K.H.; Biswas, G.P.; Khan, M.K.; Kumar, N. An Efficient and Practical Smart Card Based Anonymity Preserving User Authentication Scheme for TMIS using Elliptic Curve Cryptography. *J. Med. Syst.* **2015**, *39*, 1–18.
15. 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.

16. 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.* **2020**, *14*, 39–50.
17. Zhang, L.; Zhu, S.; Tang, S. Privacy Protection for Telecare Medicine Information Systems Using a Chaotic Map-Based Three-Factor Authenticated Key Agreement Scheme. *IEEE J. Biomed. Health Inform.* **2017**, *21*, 465–475.
18. Jiang, Q.; Chen, Z.; Li, B.; Shen, J.; Yang, L.; Ma, J. Security Analysis and Improvement of Bio-Hashing Based Three-Factor Authentication Scheme for Telecare Medical Information Systems. *J. Ambient. Intell. Humaniz. Comput.* **2018**, *9*, 1061–1073.
19. Deebak, B.D.; Turjman, F.A. Smart Mutual Authentication Protocol for Cloud Based Medical Healthcare Systems Using Internet of Medical Things. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 346–360.
20. Sharma, G.; Kalra, S. A Lightweight User Authentication Scheme for Cloud-IoT Based Healthcare Services. *Iran. J. Sci. Technol. Trans. Electr. Eng.* **2019**, *43*, 619–636.
21. Wazid, M.; Das, A.K.; Shetty, S.; Rodrigues, J.J.P.C.; Park, Y.H. LDAKM-EIoT: Lightweight Device Authentication and Key Management Mechanism for Edge-Based IoT Deployment. *Sensors* **2019**, *19*, 5539.
22. Zhou, L.; Li, X.; Yeh, K.H.; Su, C.; Chiu, W. Lightweight IoT-Based Authentication Scheme in Cloud Computing Circumstance. *Future Gener. Comput. Syst.* **2019**, *91*, 244–251.
23. Guo, D.; Wen, Q.; Li, W.; Zhang, H.; Jin, Z. An Improved Biometrics-Based Authentication Scheme for Telecare Medical Information Systems. *J. Med. Syst.* **2015**, *39*, 1–10.
24. Lei, C.L.; Chuang, Y.H. Privacy Protection for Telecare Medicine Information Systems With Multiple Servers Using a Biometric-Based Authenticated Key Agreement Scheme. *IEEE Access* **2019**, *7*, 186480–186490.
25. Hsu, C.L.; Le, T.V.; Hsieh, M.C.; Tsai, K.Y.; Lu, C.F.; Lin, T.W. Three-Factor USCCO Scheme With Fast Authentication and Privacy Protection for Telecare Medicine Information Systems. *IEEE Access* **2020**, *8*, 196553–196566.
26. Gupta, A.; Tripathi, M.; Shaikh, T.J.; Sharma, A. A Lightweight Anonymous User Authentication and Key Establishment Scheme for Wearable Devices. *IEEE Access* **2019**, *149*, 29–42.
27. Hajian, R.; ZakeriKia, S.; Erfani, S.H.; Mirabi, M. SHAPARAK: Scalable Healthcare Authentication Protocol With Attack-Resilience and Anonymous Key-Agreement. *Comput. Netw.* **2020**, *183*, 1–18.
28. 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.
29. Kusters, L.; Willems, F.M.J. Secret-Key Capacity Regions for Multiple Enrollments with An SRAM-PUF. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2276–2287.
30. Rahman, M.T.; Rahman, F.; Forte, D.; Tehranipoor, M. An Aging-Resistant RO-PUF for Reliable Key Generation. *IEEE Trans. Emerg. Top. Comput.* **2016**, *4*, 335–348.
31. Phalak, K.; Saki, A.A.; Alam, M.; Topaloglu, R.O.; Ghosh, S. Quantum PUF for Security and Trust in Quantum Computing. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2021**, *11*, 333–342.
32. Gu, J.; Cao, X.Y.; Yin, H.L.; Chen, Z.B. Differential Phase Shift Quantum Secret Sharing Using a Twin Field. *Opt. Express* **2021**, *29*, 9165–9173.
33. Lu, Y.S.; Cao, X.Y.; Weng, C.X.; Gu, J.; Xie, Y.M.; Zhou, M.G.; Yin, H.L.; Chen, Z.B. Efficient Quantum Digital Signatures without Symmetrization Step. *Opt. Express* **2021**, *29*, 10162–10171.
34. Xie, Y.M.; Lu, Y.S.; Weng, C.X.; Cao, X.Y.; Jia, Z.Y.; Bao, Y.; Wang, Y.; Fu, Y.; Lei, F.H. Breaking the Rate-Loss Bound of Quantum Key Distribution with Asynchronous Two-Photon Interference. *PRX Quantum* **2022**, *3*, 1–15.
35. Gao, Y.; Sarawi, S.F.A.; Abbott, D. Physical Unclonable Functions. *Nat. Electron.* **2020**, *3*, 81–91.
36. Frikken, K.B.; Blanton, M.; Atallah, M.J. Robust Authentication Using Physically Unclonable Functions. In Proceedings of the International Conference on Information Security, Pisa, Italy, 7–9 September 2009; pp. 262–277.
37. Dolev, D.; Yao, A.C. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208.
38. Canetti, R.; Krawczyk, H. Universally Composable Notions of Key Exchange and Secure Channels. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'02), Amsterdam, The Netherlands, 28 April–2 May 2002; pp. 337–351.
39. 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.
40. Yu, S.J.; Lee, J.Y.; Park, Y.H.; Park, Y.H.; Lee, S.W.; Chung, B.H. A Secure and Efficient Three-Factor Authentication Protocol in Global Mobility Networks. *Appl. Sci.* **2020**, *10*, 3565–3588.
41. Das, A.K.; Wazid, M.; Kumar, N.; Vasilakos, A.V.; Rodrigues, J.J.P.C. Biometrics-Based Privacy-Preserving User Authentication Scheme for Cloud-Based Industrial Internet of Things Deployment. *IEEE Internet Things J.* **2018**, *5*, 4900–4913.
42. Li, J.; Zhang, N.; Ni, J.; Chen, J.; Du, R. Secure and Lightweight Authentication With Key Agreement for Smart Wearable Systems. *IEEE Internet Things J.* **2020**, *7*, 7334–7344.
43. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf's Law in Passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791.
44. Boyko, V.; Mackenzie, P.; Patel, S. Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman. In Proceedings of the International conference on the theory and applications of cryptographic techniques, Bruges, Belgium, 14–18 May 2000; pp. 156–171.
45. Oheimb, D.V. The High-Level Protocol Specification Language HLPSL Developed in the EU Project AVISPA. In Proceedings of the APPSEM 2005 Workshop, Tallinn, Finland, 12–15 September 2005; pp, 1–17.

46. SPAN. A Security Protocol Animator for AVISPA. 2001. Available online: http://www.avispa-project.org/ (accessed on 16 March 2021).
47. Secure Hash Standard. FIPS PUB 180-1. National Institute of Standards and Technology (NIST). U.S. Department of Commerce. 1995. Available online: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf (accessed on 13 January 2021).
48. Advanced Encryption Standard (AES). FIPS PUB 197. National Institute of Standards and Technology (NIST). U.S. Department of Commerce. 2001. Available online: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf (accessed on 13 January 2021).