*sensors*

MDPI

*Article*

# A Smartcard-Based User-Controlled Single Sign-On for Privacy Preservation in 5G-IoT Telemedicine Systems

**Tzu-Wei Lin [1], Chien-Lung Hsu [1,2,3,4,5,*], Tuan-Vinh Le [1], Chung-Fu Lu [6] and Bo-Yu Huang [2]**

[1] Graduate Institute of Business and Management, Chang Gung University, Taoyuan 333, Taiwan; d0640001@cgu.edu.tw (T.-W.L.); tvle.cgu@gmail.com (T.-V.L.)
[2] Department of Information Management, Chang Gung University, Taoyuan 333, Taiwan; hpy53bert@gmail.com
[3] Healthy Aging Research Center, Chang Gung University, Taoyuan 333, Taiwan
[4] Department of Visual Communication Design, Ming-Chi University of Technology, New Taipei City 243, Taiwan
[5] Department of Nursing, Chang Gung Memorial Hospital, Taoyuan 333, Taiwan
[6] Department of Information Management, Chihlee University of Technology, New Taipei City 220, Taiwan; peter61@mail.chihlee.edu.tw
* Correspondence: clhsu@mail.cgu.edu.tw

**Abstract:** Healthcare is now an important part of daily life because of rising consciousness of health management. Medical professionals can know users' health condition if they are able to access information immediately. Telemedicine systems, which provides long distance medical communication and services, is a multi-functional remote medical service that can help patients in bed in long-distance communication environments. As telemedicine systems work in public networks, privacy preservation issue of sensitive and private transmitted information is important. One of the means of proving a user's identity are user-controlled single sign-on (UCSSO) authentication scheme, which can establish a secure communication channel using authenticated session keys between the users and servers of telemedicine systems, without threats of eavesdropping, impersonation, etc., and allow patients access to multiple telemedicine services with a pair of identity and password. In this paper, we proposed a smartcard-based user-controlled single sign-on (SC-UCSSO) for telemedicine systems that not only remains above merits but achieves privacy preservation and enhances security and performance compared to previous schemes that were proved with BAN logic and automated validation of internet security protocols and applications (AVISPA).

**Keywords:** telemedicine systems; user-controlled; single sign-on; multi-server; BAN logic; AVISPA

check for updates

**Citation:** Lin, T.-W.; Hsu, C.-L.; Le, T.-V.; Lu, C.-F.; Huang, B.-Y. A Smartcard-Based User-Controlled Single Sign-On for Privacy Preservation in 5G-IoT Telemedicine Systems. *Sensors* **2021**, *21*, 2880. https://doi.org/10.3390/s21082880

Academic Editor: Wenjuan Li

Received: 20 March 2021
Accepted: 16 April 2021
Published: 20 April 2021

## 1. Introduction

Healthcare is now an important part of daily life because of rising consciousness of health management. People can check up health conditions by themselves, such as heartbeat rate, quality of sleep, amount of exercise, and so on, by supporting wearable technology, including smart phone, smart watch, smart bracelet, etc., which measures biodata and assists self-health management. Currently, biodata is only transferred to a smartphone and analyzed by applications on a smart phone, without being transferred to other outside systems [1]. Medical professionals can know users' health conditions, if medical professionals are able to access the information immediately [1].

Telemedicine systems provide long distance medical communication and services through which patient and medical professionals can communicate online, and patient benefits from being supported with ambulatory care or other medical services, even in remote areas [1–4]. Telemedicine systems allow health related data and image to be reliably transmitted from one point to another [1]. Many researchers focused on monitoring patient's health with specific diseases using telemedicine, such as diabetes and Parkinson's disease, and telemedicine systems can help a patient recover from illness through this way [5–8]. In other words, telemedicine can help patients improve their quality of life [1]. Moreover, telemedicine systems can provide better solutions in emergency situations and serious disease monitoring [1,8–10]. Telemedicine systems are implemented with wireless communication environments, such as Wi-Fi, Internet of Things (IoT), and fifth generation (5G), to achieve long distance medical communication and services [8,11]. The sensors, such as wearable devices, for example, gather measured data, and measured data are then transmitted through gateways, 5G base stations, and core networks. After this, data is stored or analyzed by applications in back-end servers or cloud servers.

Telemedicine systems are implemented with wireless communication environments, which means data are transmitted through public networks. The patient sends healthcare-related information through public networks when using telemedicine technology, and the transmitted information is important, sensitive, and private [1]. Security issues related to data transmission were discussed, such as eavesdropping, man-in-the-middle (MITM) attack, data tempering attack, message modification attack, data interception attack, etc. [12] Although the Health Insurance Portability and Accountability Act (HIPAA), General Data Protection Regulation (GDPR), and Safe Harbor Laws have regulations that provide privacy of personal information, technical support is still not enough [12–14].

Security issues exist in multi-server environments when applying conventional password-based authenticated key exchange schemes. Users have to maintain pairs of identifiers and passwords that increase computational cost and security risks. Moreover, a trustworthy third party is required, while users utilize a single pair of identity and password in multi-server telemedicine systems. However, a malicious third party is able to impersonate users to access other services, with knowledge of shared keys. The same can be proved if a malicious server exists. In terms of performance, the cost of establishing a session key for users is related to the number of servers in conventional schemes. A single sign-on (SSO) mechanism can overcome the above issues that allow users a single action to achieve authentication with a single pair of identity and password, rather than with multiple passwords [15–17].

In the proposed scheme, we apply a user-controlled SSO (UCSSO) authenticated key agreement. The key allows patients access to services in multi-server telemedicine systems with a user-defined password, and establishes a secret shared key among servers for securing subsequent communications and designing a smartcard-based user-controlled single sign-on (SC-UCSSO) scheme that can be applied in 5G-IoT multi-server telemedicine systems. The patients has data ownership as they can control and decide the data's destination and time of transmission. The proposed scheme establishes a secure communication channel using authenticated session keys between patients and services, while meeting general security requirements. Moreover, computational complexity is better than the compared previous schemes. We sketch the remaining organization of paper below. We introduce telemedicine systems and the Chebyshev chaotic maps in Section 2. We introduce our scheme in Section 3, and security and performance analysis are detailed in Sections 4 and 5. We present our results of implementation in Section 6. Finally, the conclusion is drawn in Section 7.

## 2. Related Works

### 2.1. Telemedicine Systems

Telemedicine systems is a technology of electronic message and telecommunication related to healthcare [1,18]. The National Health Services (NHS) of United Kingdom defines changes to NHS service model as "out of hospital care", "reducing emergency hospital services", "personalized care", "digitally enables care", and "integrated care systems models", which can correspond to the features of telemedicine systems [19,20]. Thanks to the IoT technology, which enable medical professionals to monitor patients who are outside of medical institutes in real-time, medical professionals can know users' health condition, if they are able to view the information immediately [1,20]. In other words, telemedicine systems with IoT can enhance functions of patient's health monitor and proactive and preventive healthcare interventions [20].

A general telemedicine system can be divided into three level [21]. Level 1 (primary healthcare unit) consists of users with webcam, smart phone, or wearable devices, which is enables communications of measured biodata through wireless communications, including radio frequency identification (RFID), near field communication (NFC), Bluetooth, Wi-Fi, Zigbee, etc. [20]. Measured biodata are transmitted to the user's smartphone without being transferred to other systems [1]. Level 2 (city or district hospital) is clinic or local hospital that the patient might visit before being transferred to a large hospital or medical center. Level 3 (specialty center) takes part in telemedicine in case of a rare disease or an incurable disease [21]. Figure 1 illustrates a general telemedicine system including two scenarios—asynchronous telemedicine and synchronous telemedicine [21,22]. Asynchronous telemedicine allows patients to decide a proper time to send medical image and health record to medical service providers for detailed examination. Synchronous telemedicine, also called synchronous video conferencing or interactive telemedicine, provides real-time communication between patient and medical professional [22].

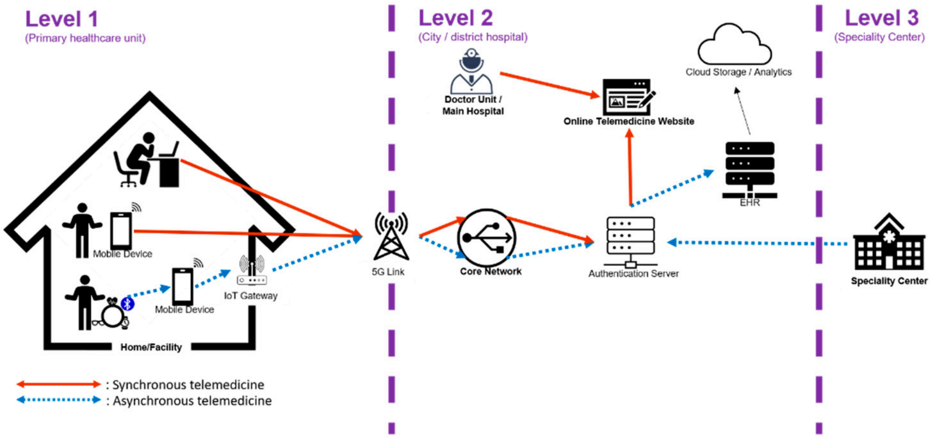


**Figure 1.** A general telemedicine system with asynchronous and synchronous telemedicine.

## 2.2. Medical Privacy

Telemedicine systems have many challenges, such as infrastructure, connections, professional requirements, data management, and real-time monitoring [23,24]. Medical privacy is of the utmost importance, and damage of medical privacy not only brings huge economic losses and losses of credibility to hospitals and other related institutions but does potential harm to patients and endangers lives of patients [24,25]. Unfortunately, thus far, healthcare-related industries did not achieve users' expectations [24]. Trust management (TM) is important for allowing reliable data collection and transmission, to provide qualified services and enhance user privacy and information security [26]. Gambetta first defined two widely accepted definitions of trust called reliability trust and decision trust [26,27]. Recently, researchers had discussions about TM of IoT [28–32]. Fortino et al. summarized and discussed main trust concepts, including behavior trust, reputation, honesty, and accuracy [26].

As we mentioned, telemedicine is implemented in public networks, so privacy preservation is one of notable security issues, which has caught researchers' attention. Mishra et al. [33] and Renuka et al. [34] utilized a biometric feature to design authentication schemes for telemedicine systems. Zriqat and Altamimi discussed issues through data collection, data transmission, and data storage and access level [12]. Dharminder et al. discussed authorized access to healthcare services [35]. Zhang et al. [36], Zhang et al. [37], and Sureshkumar et al. [38] designed authentication and key agreement for telemedicine system. Baker et al. [8], Guo et al. [39], and Anwar et al. [11] focused on telemedicine using IoT, blockchain, and 5G technology and proposed framework or scheme. In summary, three keys to the question must be solved for assuring telemedicine environments. First, image storage should be highly efficient. Second, transmitting sensitive image should satisfy confidence, integrity, and accessibility. Finally, encryption progress should be efficient, especially for the end-point.

## 3. Proposed Scheme

In the proposed system, there are $i$ users and $j$ servers. User $U_i$ can use a smartcard or a smart token to log in to whichever server $S_j$ user $U_i$ wishes to access, as shown in Figure 2. The proposed scheme includes four phases—system initialization phase, registration phase, authenticated key exchange phase, and offline password change phase. In the system initialization phase, Server $S_j$ generates essential parameters and functions for the whole scheme. User $U_i$ becomes a legitimate member in the system through the registration phase. In the authenticated key exchange phase, User $U_i$ and server $S_j$ authenticate each other and establish a session key for symmetric encryption for communication and transmitted measured biodata. The proposed scheme provides offline password change phase such that user $U_i$ can change the password periodically, without the participation of server $S_j$. Notations are defined in Table 1.

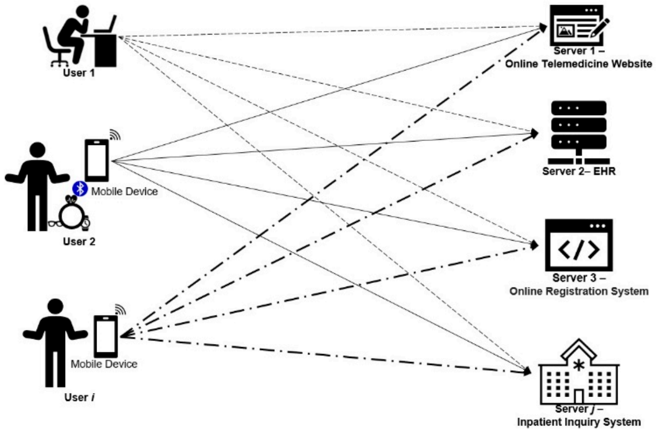


**Figure 2.** System structure of the proposed scheme.

**Table 1.** Notations of the proposed scheme.

| Notations | Definitions |
|---|---|
| $ID_i$ | Identity of user $U_i$. |
| $SID_j$ | Identity of server $S_j$. |
| $\oplus$ | Exclusive OR (XOR) operation. |
| $H(.)$ | Collision-resistant one-way hash function. |
| $PWi$ | Password of user $U_i$. |
| $x_{S_j}$ | Secret value of server $S_j$. |
| $k$ | Encryption/decryption key $k$. |
| $E_k(.)/D_k(.)$ | A symmetric encryption/decryption algorithm with secret key $k$. |
| $x$, $yi$, $\rho i$ | Random numbers. |
| $h_k(.)$ | Collision-resistance secure one-way chaotic hash function. |
| USB | Portable USB device. |
| $sj$ | Server $S_j$'s new chaotic random number. |

### 3.1. Preliminary

We briefly introduce Chebyshev chaotic maps in this section. The chaotic system has properties that can correspond to the cryptosystem's properties. First, the result is unpredictable if small changes in initial values occur [40–42]. Second, the chaotic system is a complex oscillation [40–42]. Third, the chaotic system shows qualitative change of character of solutions [40–42]. The above features can correspond to confusion and diffusion of the cryptosystem, which was discussed for decades [24,40–50]. Mathematical definitions of the Chebyshev chaotic maps are introduced as below [24,46–50].

- Polynomials of Chebyshev chaotic maps $T_n(x) : [-1, 1] \rightarrow [-1, 1]$ is formed as $T_n(x) = \cos(ncos^{-1}(x))$ in $x$ of degree $n$.
- If $n \geq 2$, polynomials of the Chebyshev chaotic maps is formed as $T_n(x) = 2xT_{n-1}(x) -T_{n-2}(x)$. However, results of the Chebyshev chaotic maps are 1 and $x$ when $n$ is 0 and 1, respectively.
- If $(s,\ r) \in Z$ and $s \in [-1, 1]$, $T_r(T_s(x)) = T_{rs}(x) = T_s(T_r(x))$, which is the so-called semi-group property.
- Zhang [51] proved that semi-group property can hold if Chebyshev polynomials are extended on interval $[-\infty, +\infty]$. In the situation, polynomials of Chebyshev chaotic maps become $T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x))$ mod $N$ where $n \geq 2$, $x \in (-\infty, +\infty)$, and $N$ is a large prime number, and $T_r(T_s(x))$ mod $N = T_{rs}(x)$ mod $N = T_s(T_r(x))$ mod $N$.
- Even only with the knowledge of $x$ and $y$, $n$ is computationally infeasible to be obtained such that $T_n(x)$ mod $N = y$, which is the so-called Chaotic maps-based discrete logarithm problem (CMDLP).
- Even only with the knowledge of ($x$ $T_r(x)$ mod $N$, $T_s(x)$ mod $N$), $T_{rs}(x)$ mod $N$ is computationally infeasible to be obtained, which is the so-called Chaotic maps-based Diffie-Hellman problem (CMDHP).

The proposed scheme applies the extended Chebyshev chaotic maps, which satisfies the above definitions.

### *3.2. System Initialization Phase*

User $U_i$ sets up smartcard by entering an identifier and password in the system initialization phase. Server $S_j$ sets up the system's parameters by performing the following steps.

Step 1. Server $S_j$ generates a secret value $x_{S_j}$, a big prime $p$, and a random number $x \in (-\infty, +\infty)$.

Step 2. Server $S_j$ choses a symmetric encryption algorithm $E_k(.)$, a symmetric decryption algorithm $D_k(.)$, a collision-resistance one-way hash function $H(.)$, and a collision-resistance secure one-way chaotic hash function $h_k(.)$.

### *3.3. Registration Phase*

User $U_i$ and server $S_j$ perform the following steps to complete the registration phase to become a legitimate member, as illustrated in Figure 3.

Step 1. User $U_i$ enters $ID_i$ and $PW_i$.

Step 2. User $U_i$ uses the smartcard to choose a random number $y_i \in Z_p^*$. After that, smartcard computes $(\alpha_i,\ A_i)$ as below. Finally, smartcard stores $y_i$ and sends $(ID_i,\ A_i)$ to server $S_j$.

$$\alpha_i = T_{y_i}(x) \bmod p \tag{1}$$

$$A_i = h_{\alpha_i}(PW_i) \oplus h_{\alpha_i}(y_i,\ SID_j) \tag{2}$$

Step 3. After receiving $(ID_i,\ A_i)$, server $S_j$ computes elements below. Then, server $S_j$ returns $(B_i,\ B_j)$ to user $U_i$.

$$\beta_j = T_{x_{S_j}}(x) \bmod p \tag{3}$$

$$u_i = h_{\beta_j}(ID_i) \tag{4}$$

$$u_j = h_{\beta_j}(SID_i) \tag{5}$$

$$B_i = U_i \oplus A_i \tag{6}$$

$$B_j = U_j \oplus A_i \tag{7}$$

Step 4. Upon receiving ($B_i$, $B_j$), user $U_i$ stores ($B_i$, $B_j$) in USB or smartcard.

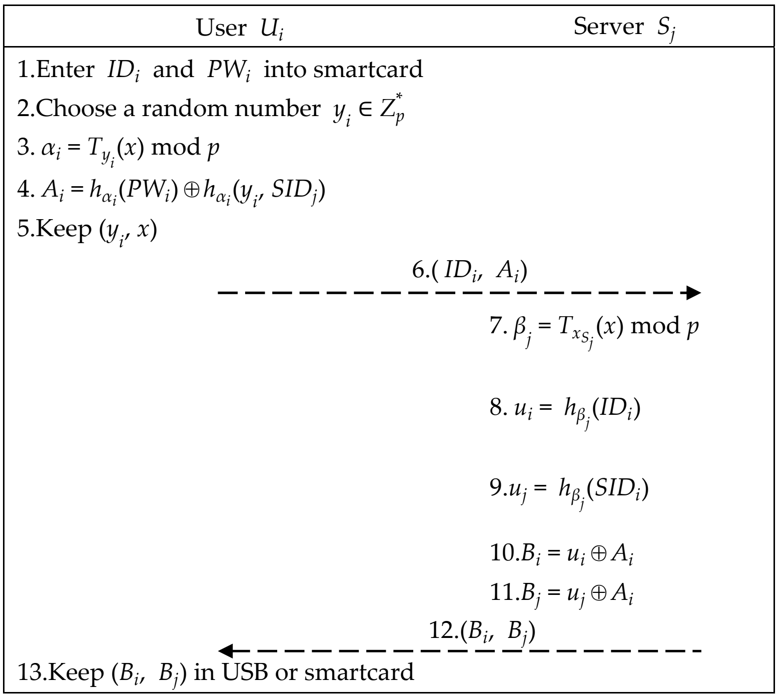

User $U_i$
Server $S_j$
1.Enter $ID_i$ and $PW_i$ into smartcard
2.Choose a random number $y_i \in Z_p^*$
3. $\alpha_i = T_{y_i}(x) \bmod p$
4. $A_i = h_{\alpha_i}(PW_i) \oplus h_{\alpha_i}(y_i, SID_j)$
5.Keep $(y_i, x)$
6.$(ID_i, A_i)$
7. $\beta_j = T_{x_{S_j}}(x) \bmod p$
8. $u_i = h_{\beta_j}(ID_i)$
9.$u_j = h_{\beta_j}(SID_i)$
10.$B_i = u_i \oplus A_i$
11.$B_j = u_j \oplus A_i$
12.$(B_i, B_j)$
13.Keep $(B_i, B_j)$ in USB or smartcard

**Figure 3.** Registration phase of the proposed scheme.

### 3.4. Authenticated Key Exchange Phase

To complete the mutual authentication and session key confirmation and obtain the remote server's services, user $U_i$, user $U_i$'s smartcard, and a server $S_j$ perform the following steps, as illustrated in Figure 4.

Step 1. User $U_i$ enters $ID_i$ and $PW_i$.

Step 2. Smartcard checks $PW_i$, utilizes ($y_i$, $x$) to compute $A_i$, retrieves ($B_i$, $B_j$) to recover $u_i$, and computes ($K_i$, $R_i$), as below.

$$u_i = B_i \oplus A_i \tag{8}$$

$$K_i = A_i \oplus h_{\alpha_i}(y_i) \tag{9}$$

$$R_i = B_j \oplus h_{\alpha_i}(y_i) \tag{10}$$

Step 3. Smartcard chooses integer $\rho_i \in (-\infty, +\infty)$ and a big prime $N_i$ to compute ($\mu_i$, $b_i$, $C_i$) as below, and sends ($R_i$, $C_i$, $N_i$) to server $S_j$.

$$\mu_i = T_{y_i}(\rho_i) \bmod N_i \tag{11}$$

$$b_i = E_{u_i}(N_i || \mu_i) \tag{12}$$

$$C_i = E_{K_i}(ID_i, b_i, \rho_i) \tag{13}$$

Step 4. After receiving ($R_i$, $C_i$, $N_i$), server $S_j$ computes the equations below. If server $S_j$ can decrypt $b_i$ successfully, server $S_j$ successfully authenticates user $U_i$.

$$K_i = R_i \oplus h_{\beta_j}(SID_j) \tag{14}$$

$$(ID_i, b_i, \rho_i) = D_{K_i}(C_i) \tag{15}$$

$$u_i = h_{\beta_j}(ID_i) \tag{16}$$

$$(N_i || \mu_i) = D_{u_i}(b_i) \tag{17}$$

Step 5. For establishing a shared session key, server $S_j$ chooses a random number $s_j \in Z_p^*$, utilizes $\rho_i$, $N_i$, and $\mu_i$ retrieved from Step 4 to compute $\omega_j$, $k_{ji}$, and $MAC_{S_j}$, and sends $\left(MAC_{S_j},\ \omega_j\right)$ to user $U_i$.

$$\omega_j = T_{s_j}(\rho_i) \bmod N_i \tag{18}$$

$$k_{ji} = H\left(T_{s_j}(\mu_i) \bmod N_i\right) \tag{19}$$

$$MAC_{S_j} = h_{k_{ji}}(SID_j,\ ID_i,\ \mu_i) \tag{20}$$

Step 6. Upon receiving $\left(MAC_{S_j},\ \omega_j\right)$, user $U_i$'s smartcard computes $k_{ij}$ and checks whether $MAC_{S_j}$ is correct. If it holds, the mutually shared session key is correct. Then, user $U_i$'s smartcard computes $MAC_{U_i}$ and sends it to server $S_j$.

$$k_{ij} = H(T_{y_i}(\omega_j) \bmod N_i) \tag{21}$$

$$MAC_{S_j}\ ? = h_{k_{ji}}(SID_j,\ ID_i,\ \mu_i) \tag{22}$$

$$MAC_{U_i} = h_{k_{ij}}(ID_i,\ SID_j,\ \omega_j\Big) \tag{23}$$

Step 7. Upon receiving $MAC_{U_i}$, server $S_j$ checks whether $MAC_{U_i}$ is correct. If it holds, the shared session key confirmation is complete.

$$MAC_{U_i}\ ? = h_{k_{ij}}(ID_i,\ SID_j,\ \omega_j\Big) \tag{24}$$

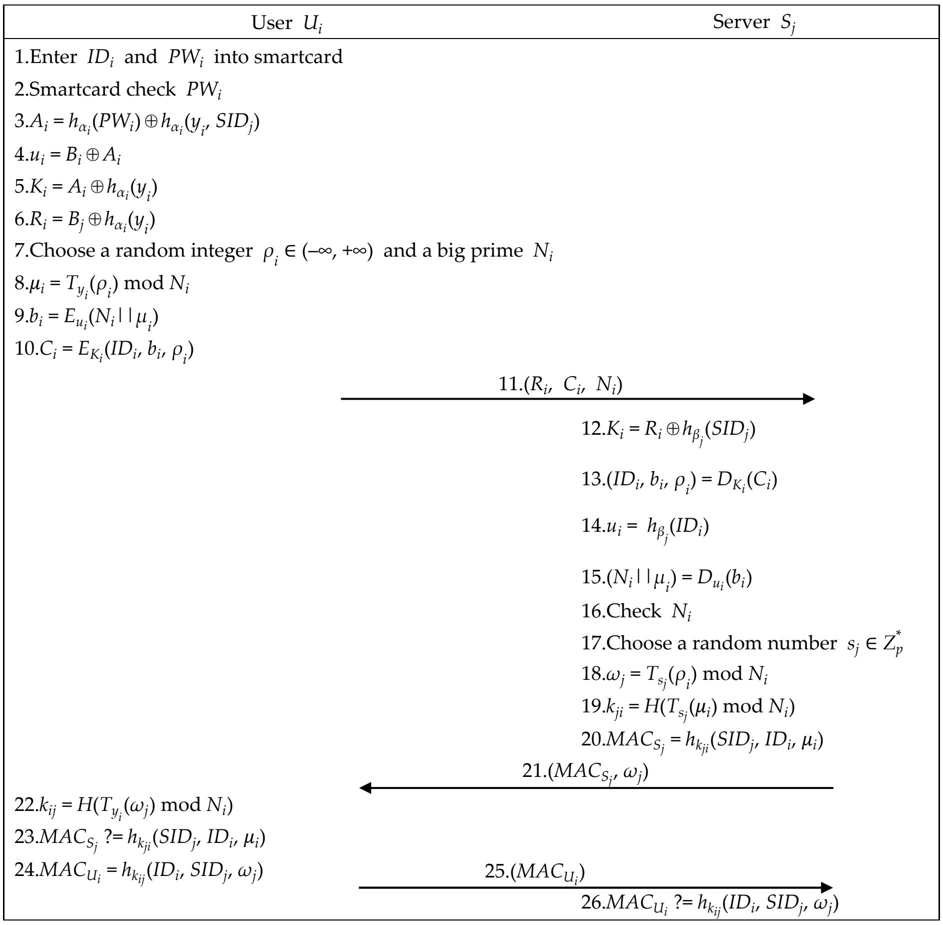

User $U_i$
Server $S_j$
1.Enter $ID_i$ and $PW_i$ into smartcard
2.Smartcard check $PW_i$
3.$A_i = h_{\alpha_i}(PW_i) \oplus h_{\alpha_i}(y_i, SID_j)$
4.$u_i = B_i \oplus A_i$
5.$K_i = A_i \oplus h_{\alpha_i}(y_i)$
6.$R_i = B_j \oplus h_{\alpha_i}(y_i)$
7.Choose a random integer $\rho_i \in (-\infty, +\infty)$ and a big prime $N_i$
8.$\mu_i = T_{y_i}(\rho_i) \bmod N_i$
9.$b_i = E_{u_i}(N_i || \mu_i)$
10.$C_i = E_{K_i}(ID_i, b_i, \rho_i)$
11.$(R_i,\ C_i,\ N_i)$
12.$K_i = R_i \oplus h_{\beta_j}(SID_j)$
13.$(ID_i, b_i, \rho_i) = D_{K_i}(C_i)$
14.$u_i = h_{\beta_j}(ID_i)$
15.$(N_i || \mu_i) = D_{u_i}(b_i)$
16.Check $N_i$
17.Choose a random number $s_j \in Z_p^*$
18.$\omega_j = T_{s_j}(\rho_i) \bmod N_i$
19.$k_{ji} = H(T_{s_j}(\mu_i) \bmod N_i)$
20.$MAC_{S_j} = h_{k_{ji}}(SID_j, ID_i, \mu_i)$
21.$(MAC_{S_j}, \omega_j)$
22.$k_{ij} = H(T_{y_i}(\omega_j) \bmod N_i)$
23.$MAC_{S_j}$ ?= $h_{k_{ji}}(SID_j, ID_i, \mu_i)$
24.$MAC_{U_i} = h_{k_{ij}}(ID_i, SID_j, \omega_j)$
25.$(MAC_{U_i})$
26.$MAC_{U_i}$ ?= $h_{k_{ij}}(ID_i, SID_j, \omega_j)$

**Figure 4.** The authenticated key exchange phase of the proposed scheme.

### *3.5. Offline Password Change Phase*

User $U_i$ and smartcard cooperatively perform the following steps to complete the password changing process, as illustrated in Figure 5.

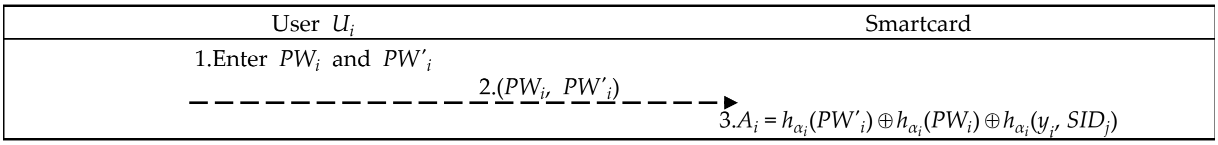

User $U_i$
Smartcard
1.Enter $PW_i$ and $PW'_i$
2.$(PW_i, PW'_i)$
3.$A_i = h_{\alpha_i}(PW'_i) \oplus h_{\alpha_i}(PW_i) \oplus h_{\alpha_i}(y_i, SID_j)$

**Figure 5.** Offline password change phase of the proposed scheme.

Step 1. User $U_i$ enters PIN to start smartcard and inputs old $PW_i$ and new $PW'_i$.
Step 2. Smartcard updates $A_i$ and stores it.

$$A_i = h_{\alpha_i}(PW'_i) \oplus h_{\alpha_i}(PW_i) \oplus h_{\alpha_i}(y_i, SID_j) \tag{25}$$

## 4. Security Analysis

We apply BAN logic [52] and AVISPA tool [53] for formal security proof. We also present informal security proof, which proves that the proposed scheme can achieve some security requirements.

### *4.1. Formal Security Proof Using BAN Logic*

This subsection describes the logical analyses of the proposed scheme by using the logical tool defined by Burrows et al. [52]. The process of proof in this section is similar with some schemes, because these schemes, including the proposed scheme, aim to prove that the principles in schemes can believe the established session keys. The notations used in the BAN logic [52] analysis are defined in Table 2.

**Table 2.** Notations of BAN logic [52] used in analyzing the proposed scheme.

| Notations | Definitions |
|---|---|
| $P, Q$ | Principles. |
| $X, Y$ | Statements. |
| $r, w$ | Readers (receivers) and writers (senders). |
| $K$ | Encryption key. |
| $P$ *believes* $X$ | $P$ believes $X$. |
| $P$ *once said* $X$ | $P$ once said $X$. |
| C(X) | $X$ is transited through communication channel $C$. |
| $r$(C)/$w$(C) | Readers/writers of $C$. |
| $P$ *sees* C(X) | $P$ sees $C(X)$. |
| $P$ *sees* $X \mid C$ | $P$ sees $X$ via $C$. |
| $(X)_K$ | $X$ is encrypted with the key $K$. |
| $P \overset{K}{\leftrightarrow} Q$ | $P$ and $Q$ establish a secure communication channel using $K$. |

#### 4.1.1. Initial Assumptions

Making initial assumptions is necessary for ensuring success of scheme and establishing the foundation of logical proof [52]. Initial assumptions of the proposed scheme are listed below.

- A1. $P \in r(C_{P,\ Q})$: $P$ can read from channel $C_{P,\ Q}$.
- A2. $P$ *believes* $w(C_{P,\ Q}) = \{P,\ Q\}$: $P$ believes that $P$ and $Q$ can write on $C_{P,\ Q}$.
- A3. $P$ *believes* $Q$ *once said* $(\Phi \rightarrow \Phi)$: $P$ believes that $Q$ only says what it believes.
- A4. $P$ *believes* $\#(N_P)$: $P$ believes that $N_P$ is fresh.
- A5. $P$ *believes* $(\xrightarrow{a}_{\text{ECMDH}(secret)} P)$: $P$ believes that $a$ is $P$'s extended chaotic maps-based Diffie-Hellman secret [24,49].

#### 4.1.2. Inference Rules

The purpose of inference rules is analyzing belief, which pays attention to beliefs of principals in authentication and key agreement schemes, in order to verify message, freshness, and trustworthiness of origin of scheme [52,54–57]. We apply the seeing rules, interpretation rules, freshness rules, and the rationality rules for logical proof.

The seeing rules define that if a principle sees a formula, the principle also sees its components with knowing necessary keys. We apply S1 and S2 as below.

- S1. $\frac{P\ sees\ C(X),\ P \in r(C)}{P\ believes\ (P\ sees\ X|C),\ P\ sees\ X}$: If $P$ receives and reads $X$ via $C$, then $P$ believes that $X$ has arrived on $C$ and $P$ sees $X$.
- S2. $\frac{P\ sees\ C(X,\ Y)}{P\ sees\ X,\ P\ sees\ Y}$: If $P$ sees a hybrid message $(X, Y)$, then $P$ sees $X$ and $Y$ separately.

The interpretation rules define that a principle can believe some hybrid facts by logical reasoning. We apply I1, I2, and I3, as below.

- I1. $\frac{P\ believes\ (w(C)=\{P,\ Q\})}{P\ believes\ (P\ sees\ X|C) \rightarrow Q\ once\ said\ X}$: If $P$ believes that $C$ can only be written by $P$ and $Q$, then $P$ believes that if $P$ receives $X$ via $C$, then $Q$ said $X$.
- I2. $\frac{P\ believes\ (Q\ once\ said\ (X,\ Y))}{P\ believes\ (Q\ once\ said\ X),\ P\ believes\ (Q\ once\ said\ Y)}$: If $P$ believes that $Q$ said a hybrid message $(X, Y)$, then $P$ believes that $Q$ has said $X$ and $Y$ separately.
- I3. $\frac{P\ believes\ (\overset{a}{\rightarrow}_{\text{ECMDH}(secret)}\ P),\ P\ believes\ (\overset{T_b(x)\ \text{mod}\ N}{\rightarrow}_{\text{ECMDH}(public)}\ Q)}{P\ believes\ (P \overset{T_{ab}(x)\ \text{mod}\ N}{\leftrightarrow} Q)}$: If $P$ believes that $a$ is $P$'s extended chaotic maps-based Diffie-Hellman secret and $T_b(x)$ mod $N$ is extended chaotic maps-based Diffie-Hellman component from $Q$, then $P$ believes that $T_{ab}(x)$ mod $N$ is symmetric key shared between $P$ and $Q$.

The freshness rules define that if one part of a formula is fresh, the entire formula must be fresh. We apply F1 and F2 as below.

- F1. $\frac{P\ believes\ (Q\ once\ said\ X),\ P\ believes\ \#(X)}{P\ believes\ (Q\ once\ said\ X)}$: If $P$ believes that another $Q$ said $X$ and $P$ also believes that $X$ is fresh, then $P$ believes that $Q$ recently said $X$.
- F2. $\frac{P\ believes\ \#(X)}{P\ believes\ \#(X,\ Y)}$: If $P$ believes that a part of a mixed message $X$ is fresh, then it believes that the whole message $(X, Y)$ is fresh.

The rationality rules define that a principle can only believe what it believes. We have R1 as below.

- R1. $\frac{P\ believes\ (\Phi_1 \rightarrow \Phi_2),\ P\ believes\ \Phi_1}{P\ believes\ \Phi_2}$: If $P$ believes that $\Phi_1$ implies $\Phi_2$ and $P$ believes that $\Phi_1$ is true, then $P$ believes that $\Phi_2$ is true.

#### 4.1.3. Goals

Goals are what schemes must achieve, and goals are required while designing schemes. The goals of the proposed scheme are listed below.

- Goal 1. $U_i\ believes\ (U_i \xleftrightarrow{T_{y_i s_j}(\rho_i)\ \text{mod}\ N_i} S_j)$: User $U_i$ believes that $T_{y_i s_j}(\rho_i)$ mod $N_i$ is a symmetric key shared between participants $U_i$ and $S_j$.
- Goal 2. $S_j\ believes\ (U_i \xleftrightarrow{T_{y_i s_j}(\rho_i)\ \text{mod}\ N_i} S_j)$: Server $S_j$ believes that $T_{y_i s_j}(\rho_i)$ mod $N_i$ is a symmetric key shared between participants $U_i$ and $S_j$.
- Goal 3. $U_i\ believes\ S_j\ believes\ (U_i \xleftrightarrow{T_{y_i s_j}(\rho_i)\ \text{mod}\ N_i} S_j)$: User $U_i$ believes that $S_j$ believes $T_{y_i s_j}(\rho_i)$ mod $N_i$ is a symmetric key shared between $U_i$ and $S_j$.
- Goal 4. $S_j\ believes\ U_i\ believes\ (U_i \xleftrightarrow{T_{y_i s_j}(\rho_i)\ \text{mod}\ N_i} S_j)$: Server $S_j$ believes that $U_i$ believes $T_{y_i s_j}(\rho_i)$ mod $N_i$ is a symmetric key shared between $U_i$ and $S_j$.

#### 4.1.4. Proof

The proposed scheme can be normalized as Steps 1 and 2.

Step 1. $S_j$ *sees* $(\xrightarrow{T_{y_i}(\rho_i) \bmod N_i}_{\text{ECMDH}(public)} U_i, C_{S_j,\, U_i}(ID_i||b_{\,i}||x_{\,i2})K_i,\ N_i)$

Step 2. $U_i$ *sees* $(\xrightarrow{T_{s_j}(\rho_i) \bmod N_i}_{\text{ECMDH}(public)} S_j, C_{U_i,\, S_j}(SID_j,\ ID_i,\ T_{y_i}(\rho_{\,i}) \bmod N_i)_{k_{ij}},\ T_{s_j}(\rho_{\,i})$ $\bmod N_i)$

Equation (26) means user $U_i$ believes that $y_i$ is its extended chaotic maps-based Diffie-Hellman secret. Equation (27) means user $U_i$ believes that $T_{s_j}(\rho_{\,i}) \bmod N_i$ is the extended chaotic maps-based Diffie-Hellman component from server $S_{j.}$ To accomplish Goal 1 (User $U_i$ believes that $k_{ij} = T_{y_i s_j}(\rho_{\,i}) \bmod N_i$ is a symmetric key shared between participants user $U_i$ and server $S_j$), Equations (25) and (26) must hold, because of the interpretation rule (I3) and assumption (A5).

$$U_i \textit{ believes } (\xrightarrow{y_i}_{\text{ECDHM}(secret)} U_i) \tag{26}$$

$$U_i \textit{ believes } (\xrightarrow{T_{s_j}(\rho_i) \bmod N_i}_{\text{ECDHM}(public)} S_j) \tag{27}$$

The meaning of Equation (28) is described below. The first fact is that server $S_j$ once said that $T_{s_j}(x) \bmod p$ is the extended chaotic maps-based Diffie-Hellman public component from server $S_j$, $(SID_j,\ ID_i,\ T_{y_i}(\rho_{\,i}) \bmod N_i)$ is encrypted by $k_{ij}$ and $T_{s_j}(\rho i) \bmod N_i$. The second fact is that server $S_j$ once said that $T_{s_j}(x) \bmod p$ is the extended chaotic maps-based Diffie-Hellman public component from server $S_j$. In Equation (29), user $U_i$ believes that the first fact implies the second fact. Equation (28) means that user $U_i$ believes that server $S_j$ once said that $T_{s_j}(\rho_{\,i}) \bmod N_i$ is the extended chaotic maps-based Diffie-Hellman public component from server $S_j$. Next, to accomplish Equation (27), Equations (28) and (29) must hold because of assumption (A3) and the rationality rule (R1).

$$U_i \textit{ believes } (S_j \textit{ once said } (\xrightarrow{T_{s_j}(\rho_i) \bmod N_i}_{\text{ECDHM}(public)} S_j,\ (SID_{\,j},\ ID_i,\ T_{y_i}(\rho_{\,i}) \bmod N_i)k_{ij},\ T_{s_j}(\rho_{\,i}) \bmod p) \to (\xrightarrow{T_{s_j}(\rho_i) \bmod N_i}_{\text{ECDHM}(public)} S_j)) \tag{28}$$

$$U_i \textit{ believes } (S_j \textit{ once said } (\xrightarrow{T_{s_j}(\rho_i) \bmod N_i}_{\text{ECDHM}(public)} S_j)) \tag{29}$$

To accomplish Equation (29), Equation (30) must hold, which means that user $U_i$ believes that $T_{s_j}(\rho_{\,i}) \bmod N_i$, which is that the extended chaotic maps-based Diffie-Hellman public component from server $S_j$ is fresh because of freshness rules (F1) and (F2), and assumption (A4).

$$U_i \textit{ believes } \#(\xrightarrow{T_{s_j}(\rho_i) \bmod N_i}_{\text{ECDHM}(public)} S_j) \tag{30}$$

Equation (31) means that user $U_i$ can read from the channel $C_{S_j,\, U_i}$. Equation (32) means that user $U_i$ believes that user $U_i$ and server $S_j$ can write messages on channel $C_{S_j,\, U_i}$. Equation (33) means that user $U_i$ sees and believes that $T_{s_j}(\rho_{\,i}) \bmod N_i$ is in the channel $C_{S_j,\, U_i}$, which is the extended chaotic maps-based Diffie-Hellman public component from server $S_j$. To accomplish Equation (30), we have Equations (31)–(33) that must hold because of the interpretation rules (I1), the seeing rules (S1), (S2), assumptions (A1) and (A2). By using the interpretation rules (I3), our proposed scheme realizes that Goal 1 is achieved. Similarly, we ensured that the proposed scheme realizes Goal 2 by using the same arguments of Goal 1.

$$U_i \in\ r(C_{S_j,\, U_i}) \tag{31}$$

$$U_i \textit{ believes } (w(C_{\ S_j,\, U_i}) = \{U_{\,i},\ S_j\}) \tag{32}$$

$$U_i \textit{ sees believes } C_{S_j,\, U_i}(\xrightarrow{T_{s_j}(\rho_i) \bmod N_i}_{\text{ECDHM}(public)} S_j) \tag{33}$$

The meaning of Equation (34) is described below. The first fact is that server $S_j$ once said that $T_{y_i s_j}(\rho_{\,i}) \bmod N_i$ is the symmetric key shared between $U_i$ and $S_j$. The second fact is that server $S_j$ believes that $T_{y_i s_j}(\rho_{\,i}) \bmod N_i$ is the symmetric key shared between

$U_i$ and $S_j$. In Equation (35), user $U_i$ believes that the first fact implies the second fact. To accomplish the Goal 3, we have Equations (34) and (35), which must hold because of the rationality rule (R1) and assumption (A3).

$$U_i\ believes\ ((S_{\ j}\ once\ said\ U_i \xleftrightarrow{T_{y_i s_j}(\rho_i)\ \bmod\ N_i} S_j) S_j\ believes\ (U_{\ i} \xleftrightarrow{T_{y_i s_j}(\rho_i)\ \bmod\ N_i} S_j)) \tag{34}$$

$$U_i\ believes\ (S_{\ j}\ once\ said\ U_i \xleftrightarrow{T_{y_i s_j}(\rho_i)\ \bmod\ N_i} S_j) \tag{35}$$

Equation (36) means that user $U_i$ believes symmetric key $T_{y_i s_j}(\rho_{\ i}) \bmod N_i$ is fresh. To accomplish Equation (35), Equation (36) must hold because of the freshness rules (F1) and (F2) and assumption (A4).

$$U_i\ believes\ \#(U_{\ i} \xleftrightarrow{T_{y_i s_j}(\rho_i)\ \bmod\ N_i} S_j) \tag{36}$$

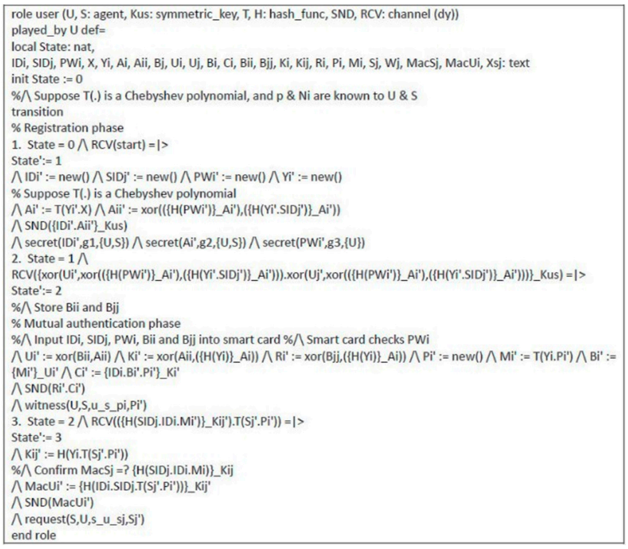
Equation (37) means that user $U_i$ sees and believes that $T_{y_i s_j}(\rho_{\ i}) \bmod N_i$ is in the channel $C_{S_j,\ U_i}$. To accomplish Equation (36), Equations (31), (32) and (37) must hold because of the interpretation rule (I1), the assumptions (A1) and (A2), and the seeing rules (S1) and (S2). Thus, the proposed scheme realizes that Goal 3 is achieved. Similarly, using the same arguments of Goal 3, the proposed scheme realizes Goal 4.

$$U_i\ sees\ believes\ C_{S_j,\ U_i}(U_{\ i} \xleftrightarrow{T_{y_i s_j}(\rho_i)\ \bmod\ N_i} S_j) \tag{37}$$

Therefore, the proposed scheme realizes Goals 1, 2, 3, and 4.

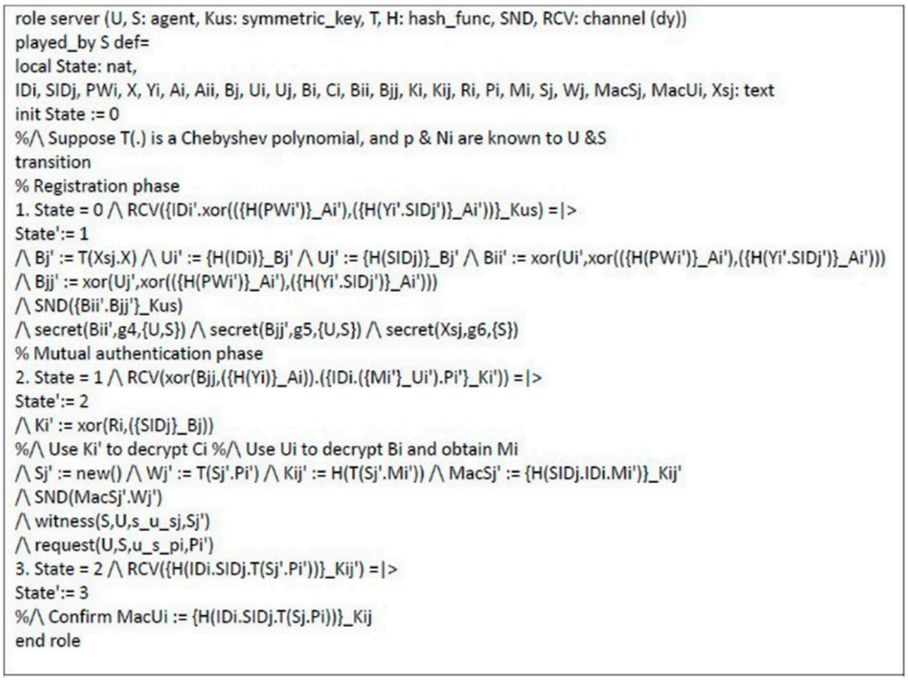
### 4.2. Formal Security Proof Using AVISPA

Automated validation of internet security protocols and applications (AVISPA) is a high-level language tool for security protocols, and it provides automatic analysis techniques through its back-ends, called on-the-fly model-checker (OFMC), constraint logic based attack searcher (CL-AtSe), SAT-based model-checker (SATMC), and tree automata based on automatic approximations for the analysis of security protocols (TA4SP) [53,58–60]. The AVISPA tool executes a simulated protocol through high-level protocol specification language (HLPSL) [61]. We used the AVISPA tool to verify the proposed scheme. The HLPSL specification of user U and server S are shown in Figures 6 and 7, respectively. The session role, environment role, and goals are also specified in HLPSL, shown in Figure 8. Figure 9 shows the results and proves that the proposed scheme is safe.



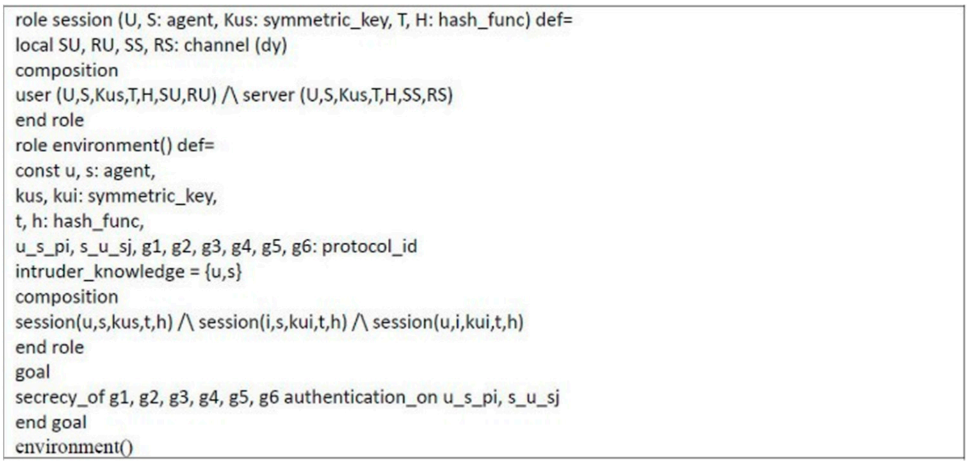
```
role user (U, S: agent, Kus: symmetric_key, T, H: hash_func, SND, RCV: channel (dy))
played_by U def=
local State: nat,
IDi, SIDj, PWi, X, Yi, Ai, Aii, Bj, Ui, Uj, Bi, Ci, Bii, Bjj, Ki, Kij, Ri, Pi, Mi, Sj, Wj, MacSj, MacUi, Xsj: text
init State := 0
%/\ Suppose T(.) is a Chebyshev polynomial, and p & Ni are known to U & S
transition
% Registration phase
1.  State = 0 /\ RCV(start) =|>
State':= 1
/\ IDi' := new() /\ SIDj' := new() /\ PWi' := new() /\ Yi' := new()
% Suppose T(.) is a Chebyshev polynomial
/\ Ai' := T(Yi'.X) /\ Aii' := xor(({H(PWi')}_Ai'),({H(Yi'.SIDj')}_Ai'))
/\ SND({IDi'.Aii'}_Kus)
/\ secret(IDi',g1,{U,S}) /\ secret(Ai',g2,{U,S}) /\ secret(PWi',g3,{U})
2.  State = 1 /\
RCV({xor(Ui',xor(({H(PWi')}_Ai'),({H(Yi'.SIDj')}_Ai'))).xor(Uj',xor(({H(PWi')}_Ai'),({H(Yi'.SIDj')}_Ai')))}_Kus) =|>
State':= 2
%/\ Store Bii and Bjj
% Mutual authentication phase
%/\ Input IDi, SIDj, PWi, Bii and Bjj into smart card %/\ Smart card checks PWi
/\ Ui' := xor(Bii,Aii) /\ Ki' := xor(Aii,({H(Yi)}_Ai)) /\ Ri' := xor(Bjj,({H(Yi)}_Ai)) /\ Pi' := new() /\ Mi' := T(Yi.Pi') /\ Bi' :=
{Mi'}_Ui' /\ Ci' := {IDi.Bi'.Pi'}_Ki'
/\ SND(Ri'.Ci')
/\ witness(U,S,u_s_pi,Pi')
3.  State = 2 /\ RCV(({H(SIDj.IDi.Mi')}_Kij').T(Sj'.Pi')) =|>
State':= 3
/\ Kij' := H(Yi.T(Sj'.Pi'))
%/\ Confirm MacSj =? {H(SIDj.IDi.Mi)}_Kij
/\ MacUi' := {H(IDi.SIDj.T(Sj'.Pi'))}_Kij'
/\ SND(MacUi')
/\ request(S,U,s_u_sj,Sj')
end role
```

**Figure 6.** HLPSL specification of user.



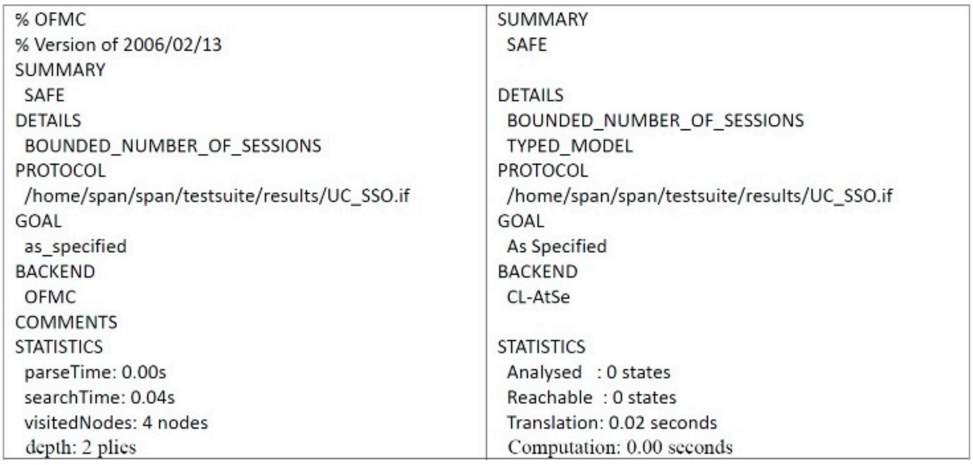
```
role server (U, S: agent, Kus: symmetric_key, T, H: hash_func, SND, RCV: channel (dy))
played_by S def=
local State: nat,
IDi, SIDj, PWi, X, Yi, Ai, Aii, Bj, Ui, Uj, Bi, Ci, Bii, Bjj, Ki, Kij, Ri, Pi, Mi, Sj, Wj, MacSj, MacUi, Xsj: text
init State := 0
%/\ Suppose T(.) is a Chebyshev polynomial, and p & Ni are known to U &S
transition
% Registration phase
1. State = 0 /\ RCV({IDi'.xor(({H(PWi')}_Ai'),({H(Yi'.SIDj')}_Ai'))}_Kus) =|>
State':= 1
/\ Bj' := T(Xsj.X) /\ Ui' := {H(IDi)}_Bj' /\ Uj' := {H(SIDj)}_Bj' /\ Bii' := xor(Ui',xor(({H(PWi')}_Ai'),({H(Yi'.SIDj')}_Ai')))
/\ Bjj' := xor(Uj',xor(({H(PWi')}_Ai'),({H(Yi'.SIDj')}_Ai')))
/\ SND({Bii'.Bjj'}_Kus)
/\ secret(Bii',g4,{U,S}) /\ secret(Bjj',g5,{U,S}) /\ secret(Xsj,g6,{S})
% Mutual authentication phase
2. State = 1 /\ RCV(xor(Bjj,({H(Yi)}_Ai)).({IDi.({Mi'}_Ui').Pi'}_Ki')) =|>
State':= 2
/\ Ki' := xor(Ri,({SIDj}_Bj))
%/\ Use Ki' to decrypt Ci %/\ Use Ui to decrypt Bi and obtain Mi
/\ Sj' := new() /\ Wj' := T(Sj'.Pi') /\ Kij' := H(T(Sj'.Mi')) /\ MacSj' := {H(SIDj.IDi.Mi')}_Kij'
/\ SND(MacSj'.Wj')
/\ witness(S,U,s_u_sj,Sj')
/\ request(U,S,u_s_pi,Pi')
3. State = 2 /\ RCV({H(IDi.SIDj.T(Sj'.Pi'))}_Kij') =|>
State':= 3
%/\ Confirm MacUi := {H(IDi.SIDj.T(Sj.Pi))}_Kij
end role
```

**Figure 7.** HLPSL specification of server.



```
role session (U, S: agent, Kus: symmetric_key, T, H: hash_func) def=
local SU, RU, SS, RS: channel (dy)
composition
user (U,S,Kus,T,H,SU,RU) /\ server (U,S,Kus,T,H,SS,RS)
end role
role environment() def=
const u, s: agent,
kus, kui: symmetric_key,
t, h: hash_func,
u_s_pi, s_u_sj, g1, g2, g3, g4, g5, g6: protocol_id
intruder_knowledge = {u,s}
composition
session(u,s,kus,t,h) /\ session(i,s,kui,t,h) /\ session(u,i,kui,t,h)
end role
goal
secrecy_of g1, g2, g3, g4, g5, g6 authentication_on u_s_pi, s_u_sj
end goal
environment()
```

**Figure 8.** HLPSL specification of session role, environment role, and goals.



```
% OFMC                                            SUMMARY
% Version of 2006/02/13                            SAFE
SUMMARY
 SAFE                                             DETAILS
DETAILS                                            BOUNDED_NUMBER_OF_SESSIONS
 BOUNDED_NUMBER_OF_SESSIONS                        TYPED_MODEL
PROTOCOL                                          PROTOCOL
 /home/span/span/testsuite/results/UC_SSO.if       /home/span/span/testsuite/results/UC_SSO.if
GOAL                                              GOAL
 as_specified                                      As Specified
BACKEND                                           BACKEND
 OFMC                                              CL-AtSe
COMMENTS
STATISTICS                                        STATISTICS
 parseTime: 0.00s                                  Analysed   : 0 states
 searchTime: 0.04s                                 Reachable  : 0 states
 visitedNodes: 4 nodes                             Translation: 0.02 seconds
 depth: 2 plies                                    Computation: 0.00 seconds
```

**Figure 9.** Results of AVISPA.

### 4.3. Informal Security Proof

We present theoretical analyses that proved that proposed scheme could achieve security requirements.

#### 4.3.1. Preventing MITM Attack

In order to prevent MITM attack, user $U_i$ and server $S_j$ can confirm whether the message is resent, modified, and replaced, by checking information through message authentication codes $MAC_{S_j}$ and $MAC_{U_i}$. User $U_i$ verifies $MAC_{S_j}= h_{k_{ji}}(SID_j, ID_i, \mu_i)$ at Step 6, and server $S_j$ verifies $MAC_{U_i}= h_{k_{ij}}\left(ID_i, SID_j, \omega_j\right)$ at Step 7 in the authenticated key exchange phase of the proposed scheme. In this way, the adversary cannot modify message authentication codes $MAC_{S_j}$ and $MAC_{U_i}$ without session key $k_{ij}$.Thus, the proposed scheme can prevent MITM attack.

#### 4.3.2. Key Confirmation

User $U_i$ can check session key $k_{ij}$ by $MAC_{S_j}\ ?= h_{k_{ji}}(SID_j, ID_i, \mu_i)$, and server $S_j$ can also check session key $k_{ji}$ through $MAC_{U_i}\ ?= h_{k_{ij}}\left(ID_i, SID_j, \omega_j\right)$ in the proposed scheme. As a result, the proposed scheme achieves key confirmation.

#### 4.3.3. Preventing Key-Compromise Impersonation and Server Spoofing Attacks

User $U_i$'s random number $y_i$ is stored in a smartcard, which is hard to obtain information. The adversary must have user $U_i$'s smartcard and correct password if they want to impersonate a legitimate user. The number of attempts that a password can be entered is limited; if the number of attempts to enter a password exceeds the allowable number of attempts, the smartcard will get locked. On the other hand, the adversary cannot obtain $K_i$ due to not knowing $x_{S_j}$, and afterwards the process cannot be completed by adversary. As a result, the proposed scheme can prevent key-compromise impersonation and server spoofing attacks.

#### 4.3.4. Mutual Authentication

In the authenticated key exchange phase of the proposed scheme, server $S_j$ encrypts $(SID_j, ID_i, \mu_i)$ from user $U_i$ to message authentication code $MAC_{S_j}$ with session key $k_{ji}= H(T_{s_j}(\mu_i) \bmod Ni)$ and sends $(MAC_{S_j}, \omega_j)$ to user $U_i$. In Step 6, user $U_i$ uses $\omega_j$ from server $S_j$ to obtain session key $k_{ij}$ and verify $MAC_{S_j}= h_{k_{ji}}(SID_j, ID_i, \omega_j)$. Server $S_j$ verifies message authentication code $MAC_{U_i}= h_{k_{ij}}\left(ID_i, SID_j, \omega_j\right)$ sent by user $U_i$ in Step 7. $MAC_{S_j}$ and $MAC_{U_i}$ are included in session keys that only two parties of communication have, so only user $U_i$ and server $S_j$ can verify each other.

#### 4.3.5. User Anonymity

User $U_i$'s identity $ID_i$ is protected by being encrypted in $C_i= E_{K_i}(ID_i, b_i, \rho_i)$ with $K_i$, before being sent. Server $S_j$ must obtain $K_i$ by computing $K_i= R_i \oplus h_{\beta_j}(SID_j)$. The adversary cannot obtain $ID_i$ even with $R_i$ and $C_i$ because only server $S_j$ has knowledge of secret $x_{S_j}$. The adversary cannot obtain $K_i$ without $x_{S_j}$ and decrypting $C_i$; thus, the adversary cannot obtain $ID_i$. As a result, the proposed scheme provides user anonymity during communication.

#### 4.3.6. Resistant to Bergamo et al.'s Attack

Bergamo et al.'s attack is based on [62]. (i) The adversary is able to obtain related elements $(x, \rho_i, \mu_i, \omega_j)$; and (ii) several Chebyshev polynomials pass through the same point due to periodicity of the cosine function. In the proposed scheme, the adversary is unable to obtain any related elements $(x, \rho_i, \mu_i, \omega_j)$ as these are encrypted in transmitted messages where only user $U_i$ and server $S_j$ can retrieve decryption key. Moreover, the proposed protocol utilizes the extended Chebyshev polynomials, in which the periodicity of the cosine function is avoided by extending the interval of $x$ to $(-\infty, +\infty)$ [51]. As a result, the proposed scheme can resist the attack proposed by Bergamo et al. [62].

## 5. Performance Analysis

We present relevant security requirements and computational complexity comparison.

### *5.1. Comparisons of Security Requirements*

Table 3 shows comparisons of security requirements that were presented in the schemes designed by Wang and Zhao [63], Yoon and Jeon [46], Lin [48], Lin and Zhu [64], Lee et al. [49], Madhusudhan et al. [65], Sureshkumar et al. [38], and us. Wang and Zhao's [63], Lin's [48], and Lin and Zhu's schemes [64] are not secure against key-compromise impersonation attack, since the transmitted messages can be replayed by an adversary. Wang and Zhao's [63], Madhusudhan et al.'s [65], and Sureshkumar et al.'s [38] scheme cannot prevent server spoofing attack. Our scheme is secure against both key-compromise impersonation attack and server spoofing attack. Furthermore, our scheme provides user anonymity, which Wang and Zhao's [63], Yoon and Jeon's [46], and Madhusudhan et al.'s [65] scheme do not. Our scheme can also prevent MITM attack, which Wang and Zhao's [63], Yoon and Jeon's [46], and Madhusudhan et al.'s [65] scheme cannot. Furthermore, our scheme ensures that users and servers use the same shared key in a session via key confirmation, which is not present in Wang and Zhao's [63], Yoon and Jeon's [46], Lin's [48], Lin and Zhu's [64], Lee et al.'s [49], Madhusudhan et al.'s [65], and Sureshkumar et al.'s [38] scheme. Moreover, our scheme can prevent DoS attacks, which Wang and Zhao's [63], Yoon and Jeon's [46], Lin's [48], and Madhusudhan et al.'s [65] scheme cannot.

**Table 3.** Comparisons of Security Requirements.

| Properties | [63] | [46] | [48] | [64] | [49] | [65] | [38] | Ours |
|---|---|---|---|---|---|---|---|---|
| Preventing key-compromise impersonation attack | X | O | X | X | O | O | O | O |
| Preventing server spoofing attack | X | O | O | O | O | X | X | O |
| Multi-server environments | X | X | X | X | X | O | O | O |
| Preventing MITM attack | X | X | O | O | O | X | O | O |
| Stolen-verification table attack | X | O | O | O | X | X | O | O |
| Key confirmation | X | X | X | X | X | X | X | O |
| Preventing clock synchronization problem | O | X | X | O | X | O | X | O |
| User anonymity | X | X | O | O | O | X | O | O |
| Preventing denial-of-service (DoS) attack | X | X | X | O | O | X | O | O |

### *5.2. Comparisons of Computational Complexity*

We present the computational complexity comparison with Lee et al.'s [49], Madhusudhan et al.'s [65], and Sureshkumar et al.'s [38] scheme, as shown in Table 4. We ignore the time taken for computing XOR operations because the value is too low to influence result. Although our scheme needs more one-way hash function operations than Lee et al.'s [49] scheme and more symmetry encryption operations than Lee et al.'s [49], Madhusudhan et al.'s [65], and Sureshkumar et al.'s [38] scheme, our scheme allows key confirmation. Even so, our scheme has the less overall computational cost than Lee et al.'s [49], Madhusudhan et al.'s [65], and Sureshkumar et al.'s [38] scheme. Users in our scheme can enjoy telemedicine services witha lower computational cost. As a result, our scheme is more efficient than Lee et al.'s [49], Madhusudhan et al.'s [65], and Sureshkumar et al.'s [38]. Figure 10 illustrates the computational complexity of the server with varying number of users, and Figure 11 illustrates the computational complexity of user with varying number of servers. The computational complexity of user in Lee et al.'s scheme [49] is related to the number of servers. Computational complexity of user in Madhusudhan et al.'s [65], Sureshkumar et al.'s [38], and the proposed scheme is not related to number of servers, and the proposed scheme shows the least computational complexity among the compared schemes.

**Table 4.** Comparisons of Computational Complexity.

| Roles | Lee et al. [49] | Madhusudhan et al. [65] | Sureshkumar et al. [38] | Ours |
|---|---|---|---|---|
| User | $3T_h+3T_{ch}+T_{sym} \approx 3T_h+525T_h+2.5T_h = 530.5T_h$ | $8T_h+5T_{ch} \approx 8T_h+875T_h = 883T_h$ | $7T_h+5T_{ch} \approx 7T_h+875T_h = 882T_h$ | $5T_h+2T_{ch}+2T_{sym} \approx 5T_h+350T_h+5T_h = 360T_h$ |
| Server | $T_h+3T_{ch}+2T_{sym} \approx T_h+525T_h+5T_h = 531.5T_h$ | $5T_h+4T_{ch} \approx 5T_h+700T_h = 705T_h$ | $3T_h+2T_{ch} \approx 3T_h+350T_h = 353T_h$ | $4T_h+2T_{ch}+2T_{sym} \approx 4T_h+350T_h+5T_h = 359T_h$ |
| Both | $1061T_h$ | $1588T_h$ | $1235\ T_h$ | $719\ T_h$ |

$T_{ch}$: Time for performing a Chebyshev chaotic maps operation; $T_{sym}$: Time for performing a symmetry encryption operation; $T_h$: Time for performing a one-way hash function operation; $T_{ch} \approx 175T_h$; $T_{sym} \approx 2.5T_h$.

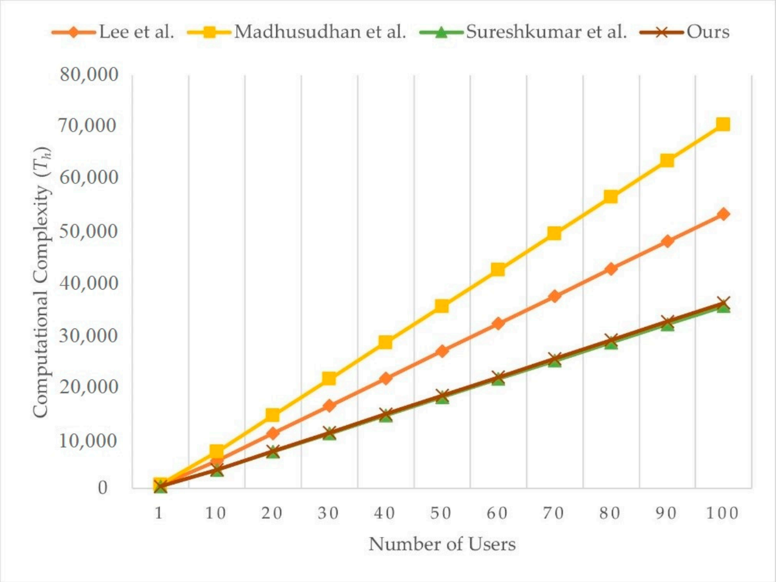


**Figure 10.** Computational complexity of server with varying number of users.

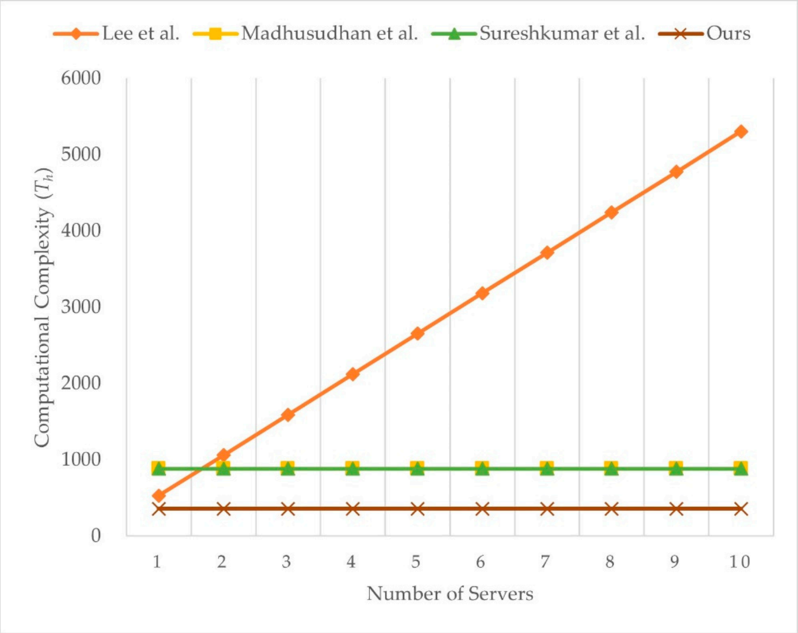


**Figure 11.** Computational complexity of user with varying number of servers.

## 6. Implementation

We developed SC-UCSSO system using the proposed scheme, in a multi-function smart token, as shown in Figure 12, which supports the public key infrastructure and the X.509 certificate. A user can insert a smart token to a computer or a laptop and insert the smartcard shown in Figure 13 into a smart token, in order to use the system. Figures 14 and 15 show the registration and login interfaces. Figure 16 shows that the user can login to multiple services, which implies that the proposed system can be used in multi-server environments. The proposed system also provides account checking (Figure 17) to manage the user's accounts. The user can login to the online telemedicine website using a computer, laptop, smartphone, or any wireless devices that has a webcam with a smart token and a smartcard in synchronous telemedicine scenario. The channel of online video consult between the patient and medical professional is protected by the session key generated by the proposed scheme. In asynchronous telemedicine, the measured biodata is transmitted to a smartphone using Bluetooth, and the user can decide when to send data to the designed server of telemedicine systems. The user logins with smart token and smartcard, before sending data. Transmitted measured data between smartphone and servers would be protected by the session key generated through the proposed scheme. The user has data ownership because the user can control data's destination and the time of being transmitted. Once data are sent by user, the privacy of user would be protected because the transmission channel is secure with the session key.

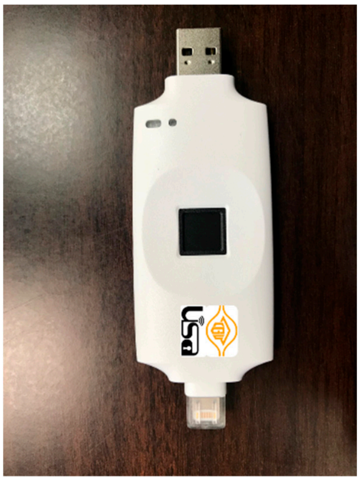


**Figure 12.** Multi-function smart token.

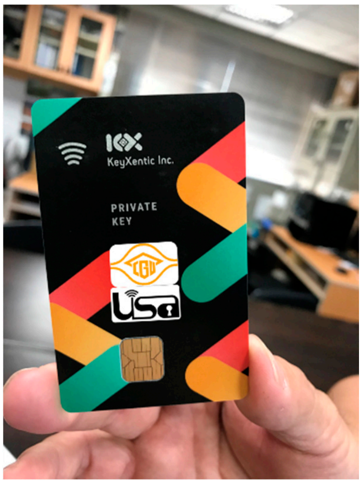
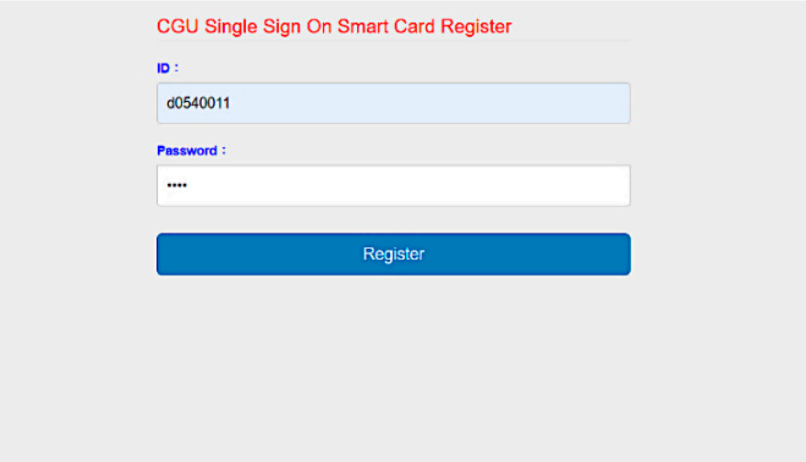


**Figure 13.** Smartcard.

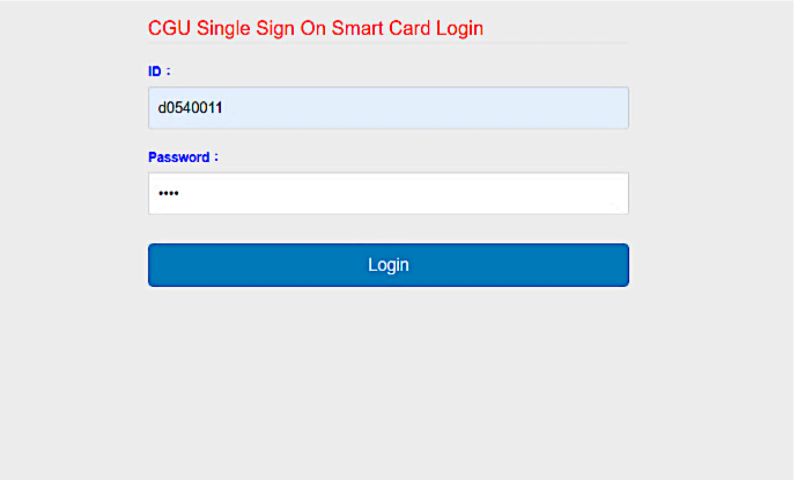


CGU Single Sign On Smart Card Register

ID : d0540011

Password : ••••

**Figure 14.** Interface of registration.

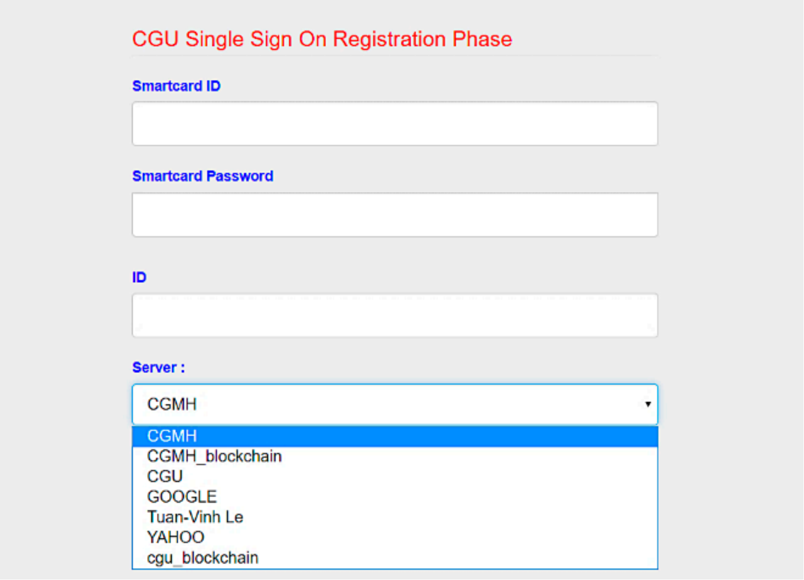


CGU Single Sign On Smart Card Login

ID : d0540011

Password : ••••

**Figure 15.** Interface of login.



CGU Single Sign On Registration Phase

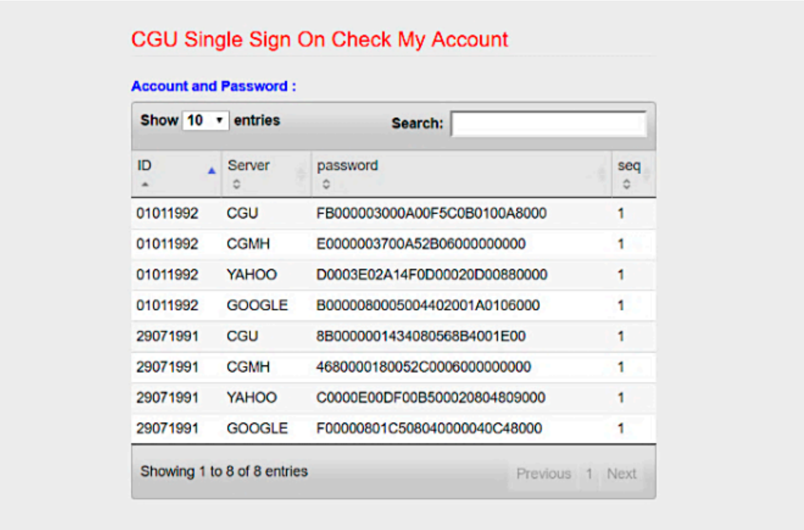
Smartcard ID

Smartcard Password

ID

Server : CGMH

**Figure 16.** Interface of choosing services.



CGU Single Sign On Check My Account

Account and Password :

Show 10 entries Search:

| ID | Server | password | seq |
|---|---|---|---|
| 01011992 | CGU | FB000003000A00F5C0B0100A8000 | 1 |
| 01011992 | CGMH | E0000003700A52B06000000000 | 1 |
| 01011992 | YAHOO | D0003E02A14F0D00020D00880000 | 1 |
| 01011992 | GOOGLE | B0000080005004402001A0106000 | 1 |
| 29071991 | CGU | 8B0000001434080568B4001E00 | 1 |
| 29071991 | CGMH | 4680000180052C0006000000000 | 1 |
| 29071991 | YAHOO | C0000E00DF00B500020804809000 | 1 |
| 29071991 | GOOGLE | F00000801C508040000040C48000 | 1 |

Showing 1 to 8 of 8 entries

**Figure 17.** Interface of account checking.

## 7. Discussion

We give a discussion for brief review, real-life scenario, and limitations of this research.

Telemedicine systems work in public networks, where privacy preservation issue of users and sensitive and private transmitted information is important [1]. Security issues related to data transmission are discussed, such as eavesdropping, MITM attack, data tempering attack, message modification attack, data interception attack, etc. [12] Although regulations, such as HIPAA, GDPR, Safe Harbor Laws, etc., were developed, technical support is still not enough [12–14]. We proposed an SC-UCSSO for the 5G-IoT telemedicine systems, which can be applied in the 5G-IoT telemedicine multi-server environments. Security of the proposed scheme was proved by BAN logic, AVISPA tool, and theoretical analyses. The proposed scheme achieved general security requirements, such as preventing MITM attack, preventing key-compromise impersonation, and server spoofing attacks, and user anonymity, key confirmation, and mutual authentication. Moreover, the proposed scheme overcomes the drawbacks of the compared previous schemes, such as stolen-verification table attack, clock synchronization problem, and DoS attack, as shown in Table 3 in the previous section. The proposed scheme applies the extended Chebychev chaotic maps that can resist Bergamo et al.'s attack [62]. Performance of the proposed scheme is also compared with Lee et al.'s [49], Madhusudhan et al.'s [65], and Sureshkumar et al.'s [38] scheme by analyzing the computational complexity of each scheme, and the results showed that the proposed scheme was less expensive ($719T_h$) in total than Lee et al.'s [49] ($1061T_h$), Madhusudhan et al.'s [65] ($1588T_h$), and Sureshkumar et al.'s [38] scheme ($1235T_h$), as shown in Table 4.

We give four possible real-life scenarios of telemedicine systems in 5G-IoT environments that can apply the proposed scheme.

Scenario 1: Patient inserts smartcard (e.g., health insurance card or smartcard, as in Figure 13) into measurement devices that include a smartcard reader, such as sphygmomanometer or blood-glucose meter, before measuring biodata. Once a patient inserts smartcard, the authenticated key agreement phase of the proposed scheme is activated, and measured biodata can be transmitted securely to server as it is encrypted by the session key.

Scenario 2: Patient's wearable healthcare device (e.g., sensors, smart watch, etc.) transmits the measured biodata to the related mobile application (APP) on a smartphone, through data synchronization via Bluetooth, NFC, RFID, etc. If the patient wants to transmit the measured biodata to server, the patient can use a smartphone with a smartcard adopter, such as the smart token in Figure 13. Once a patient inserts the smartcard, the authenticated key agreement phase of proposed scheme is activated, and the measured biodata can be securely transmitted to server as it is encrypted by the session key.

Scenario 3: Patient's measured biodata are recorded or stored in storage at home. If the patient wants to transmit the measured biodata to server, the patient can use the smartcard with a reader. Once a patient inserts the smartcard, the authenticated key agreement phase of proposed scheme is activated, and the measured biodata can be transmitted securely to server as it is encrypted by the session key.

Scenario 4: If a medical professional would like to access the measured biodata on server, the medical professional has to use the smartcard (e.g., healthcare certification IC card [66]) with a reader. Once a medical professional inserts smartcard, the authenticated key agreement phase of proposed scheme is activated, and the measured biodata can be securely transmitted as it is encrypted by the session key.

Scenario 1 to 3 allow the patient to decide the data's destination and time of transmission.

This research has limitations. We only give a software security analysis, but hardware security and availability are other aspects of security in telemedicine systems, such as electromagnetic interference (EMI), which might affect the functions on wearable devices. Although there are already measurement devices with a smartcard reader on the market, we did not evaluate the hardware's effects with the proposed scheme. We assumed that the users (patient/medical professional) have a smartcard (health insurance card/ healthcare certification IC card) and proposed a smartcard-based scheme, but authentication could be achieved in many ways, such as three-factor authentication, two-step verification, fast identity online (FIDO), etc., which can be related to works in the future.

## 8. Conclusions

Telemedicine systems is a multi-functional remote medical service that can help patients in bed in long-distance communication environments [1–4]. As telemedicine systems work in public networks, privacy preservation issue of sensitive and private transmitted information is important. [1]. We proposed a SC-UCSSO for 5G-IoT telemedicine systems, which could achieve some general security requirements, such as preventing MITM attack, preventing key-compromise impersonation and server spoofing attacks, provide user anonymity, and overcomes the drawbacks of the previous schemes compared herein. The proposed scheme establishes a secure communication channel using the authenticated session keys between patients and services of telemedicine systems, without threats of eavesdrop, impersonation, etc., and allow patient access to multiple telemedicine services, with a pair of identity and password. Formal security analysis using BAN logic [52] and the AVISPA tool [67] was given. We also gave a performance analysis and proved that the proposed scheme is more efficient than previous compared schemes, and computational complexity of the user in proposed scheme was not related to the number of servers. Moreover, the proposed scheme is suitable for asynchronous and synchronous telemedicine, and patients have data ownership because the user can control and decide data's destination and time of transmission.

**Author Contributions:** Conceptualization, T.-W.L. and C.-L.H.; methodology, T.-W.L.; formal analysis, T.-W.L. and T.-V.L.; informal analysis, T.-W.L.; software implementation, B.-Y.H.; writing—original draft preparation, T.-W.L.; writing—review and editing, C.-L.H. and C.-F.L.; supervision, C.-L.H.; funding acquisition, C.-L.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Ministry of Science and Technology, Taiwan, grant number MOST107-2221-E-182-006 and MOST 108-2221-E-182-011, Healthy Aging Research Center, Chang Gung University, Taiwan, grant number EMRPD1K0461 and EMRPD1I0481, and Chang Gung University, Taiwan, grant number PARPD3K0011 and SCRPD3H0011.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

## References

1. Garai, Á.; Péntek, I.; Adamkó, A. Revolutionizing Healthcare with IoT and Cognitive, Cloud-Based Telemedicine. *Acta Polytech. Hung.* **2019**, *16*, 163–181. [CrossRef]
2. Fong, B.; Fong, A.C.M.; Li, C.K. *Telemedicine Technologies: Information Technologies in Medicine and Telehealth*; John Wiley and Sons: New York, NY, USA, 2010. [CrossRef]
3. Ryu, S. History of Telemedicine: Evolution, Context, and Transformation. *Healthc. Inform. Res.* **2010**, *16*, 65–66. [CrossRef]
4. Abderrahim, A.; Ibtissam, F.; Habiba, C.; Hicham, E.A.; Nabil, H. AES-PRESENT: A New Secure Iot-Based Scheme for Telemedicine and E-Health Systems. *ARPN J. Eng. Appl. Sci.* **2018**, *13*, 9554–9559.
5. Fan, Y.J.; Yin, Y.H.; Xu, L.D.; Zeng, Y.; Wu, F. IoT-Based Smart Rehabilitation System. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1568–1577. [CrossRef]
6. Pasluosta, C.F.; Gassner, H.; Winkler, J.; Klucken, J.; Eskofier, B.M. An Emerging Era in the Management of Parkinson's Disease: Wearable Technologies and the Internet of Things. *IEEE J. Biomed. Health Inform.* **2015**, *19*, 1873–1881. [CrossRef] [PubMed]
7. Chang, S.; Chiang, R.; Wu, S.; Chang, W. A Context-Aware, Interactive M-Health System for Diabetics. *IT Prof.* **2016**, *18*, 14–22. [CrossRef]
8. Baker, S.B.; Xiang, W.; Atkinson, I. Internet of Things for Smart Healthcare: Technologies, Challenges, and Opportunities. *IEEE Access* **2017**, *5*, 26521–26544. [CrossRef]
9. Sarkar, S.; Misra, S. From Micro to Nano: The Evolution of Wireless Sensor-Based Health Care. *IEEE Pulse* **2016**, *7*, 21–25. [CrossRef]
10. Yin, Y.; Zeng, Y.; Chen, X.; Fan, Y. The Internet of Things in Healthcare: An Overview. *J. Ind. Infor. Integr.* **2016**, *1*, 3–13. [CrossRef]
11. Anwar, S.; Prasad, R. Framework for Future Telemedicine Planning and Infrastructure Using 5G Technology. *Wirel. Pers. Commun.* **2018**, *100*, 193–208. [CrossRef]
12. Zriqat, I.a.A.; Altamimi, A.M. Security and Privacy Issues in Ehealthcare Systems: Towards Trusted Services. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*, 229–236.
13. The 104th United States Congress, Health Insurance Portability and Accountability Act of 1996. Public Law 104-191. *United States Statut. Large.* **1996**, *110*, 1936–2103. Available online: https://pubmed.ncbi.nlm.nih.gov/16477734/ (accessed on 19 April 2021).
14. Note on Proposal for a Regulation of the European Parliament and of the Council on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of Such Data (General Data Protection Regulation): 2012/0011 (Cod). Council of the European Union. 2013. Available online: https://lobbyplag.eu/governments/assets/pdf/CD-16140_14-C1.pdf (accessed on 20 April 2021).
15. Lee, T.F. Provably Secure Anonymous Single-Sign-on Authentication Mechanisms Using Extended Chebyshev Chaotic Maps for Distributed Computer Networks. *IEEE Syst. J.* **2018**, *12*, 1499–1505. [CrossRef]
16. Liu, X.; Liu, J.; Wang, W.; Zhu, S. Android Single Sign-on Security: Issues, Taxonomy and Directions. *Future Gener. Comput. Syst.* **2018**, *89*, 402–420. [CrossRef]
17. Zakaria, N.H.; Zainul, M.F.; Katuk, N.; Tahir, H.M.; Omar, M.N. An Evaluation of Page Token in Openid Single Sign on (SSO) to Thwart Phishing Attack. *J. Telecommun. Elect. Comput. Eng.* **2018**, *10*, 19–23.
18. Marciniak, R. Role of New It Solutions in the Future of Shared Service Model. *Pollack Period.* **2013**, *8*, 187–194. [CrossRef]
19. National Health Service. The NHS Long Term Plan. National Health Service. Available online: https://www.longtermplan.nhs.uk/wp-content/uploads/2019/08/nhs-long-term-plan-version-1.2.pdf (accessed on 8 April 2021).
20. Philip, N.Y.; Rodrigues, J.J.P.C.; Wang, H.; Fong, S.J.; Chen, J. Internet of Things for in-Home Health Monitoring Systems: Current Advances, Challenges and Future Directions. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 300–310. [CrossRef]
21. Pramanik, P.K.D.; Pareek, G.; Nayyar, A. Security and Privacy in Remote Healthcare: Issues, Solutions, and Standards. In *Telemedicine Technologies: Big Data, Deep Learning, Robotics, Mobile and Remote Applications for Global Healthcare*; Elsevier: Amsterdam, The Netherlands, 2019; pp. 201–225. [CrossRef]
22. Devaraj, S.J. Emerging Paradigms in Transform-Based Medical Image Compression for Telemedicine Environment. In *Telemedicine Technologies: Big Data, Deep Learning, Robotics, Mobile and Remote Applications for Global Healthcare*; Academic Press: Cambridge, MA, USA, 2019; pp. 15–29. [CrossRef]
23. Rao, K. The Path to 5G for Health Care. Available online: https://futurenetworks.ieee.org/images/files/pdf/applications/5G--Health-Care030518.pdf (accessed on 8 April 2021).
24. Lin, T.W.; Hsu, C.L. FAIDM for Medical Privacy Protection in 5G Telemedicine Systems. *Appl. Sci.* **2021**, *11*, 1155. [CrossRef]
25. Fan, K.; Jiang, W.; Li, H.; Yang, Y. Lightweight RFID Protocol for Medical Privacy Protection in IoT. *IEEE Trans. Ind. Inform.* **2018**, *14*, 1656–1665. [CrossRef]
26. Fortino, G.; Fotia, L.; Messina, F.; Rosaci, D.; Sarné, G.M.L. Trust and Reputation in the Internet of Things: State-of-the-Art and Research Challenges. *IEEE Access* **2020**, *8*, 60117–60125. [CrossRef]

27. Gambetta, D. Can We Trust Trust? In *Trust: Making and Breaking Cooperative Relations*; Gambetta, D., Ed.; Blackwell: Oxford, UK, 1988; pp. 213–237.
28. Yan, Z.; Zhang, P.; Vasilakos, A.V. A Survey on Trust Management for Internet of Things. *J. Netw. Comput. Appl.* **2014**, *42*, 120–134. [CrossRef]
29. Sharma, A.; Pilli, E.S.; Mazumdar, A.P.; Gera, P. Towards Trustworthy Internet of Things: A Survey on Trust Management Applications and Schemes. *Comput. Commun.* **2020**, *160*, 475–493. [CrossRef]
30. Ud Din, I.; Guizani, M.; Kim, B.S.; Hassan, S.; Khan, M.K. Trust Management Techniques for the Internet of Things: A Survey. *IEEE Access* **2019**, *7*, 29763–29787. [CrossRef]
31. Guo, J.; Chen, I.R.; Tsai, J.J.P. A Survey of Trust Computation Models for Service Management in Internet of Things Systems. *Comput. Commun.* **2017**, *97*, 1–14. [CrossRef]
32. Abdelghani, W.; Zayani, C.A.; Amous, I.; Sèdes, F. Trust Management in Social Internet of Things: A Survey. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin, Germany, 2016; Volume 9844, pp. 430–441.
33. Mishra, D.; Mukhopadhyay, S.; Kumari, S.; Khan, M.K.; Chaturvedi, A. Security Enhancement of a Biometric Based Authentication Scheme for Telecare Medicine Information Systems with Nonce. *J. Med. Syst.* **2014**, *38*, 41. [CrossRef]
34. Renuka, K.; Kumari, S.; Li, X. Design of a Secure Three-Factor Authentication Scheme for Smart Healthcare. *J. Med. Syst.* **2019**, *43*, 133. [CrossRef]
35. Dharminder, D.; Mishra, D.; Li, X. Construction of RSA-Based Authentication Scheme in Authorized Access to Healthcare Services: Authorized Access to Healthcare Services. *J. Med. Syst.* **2020**, *44*, 6. [CrossRef]
36. Zhang, L.; Luo, H.; Zhao, L.; Zhang, Y. Privacy Protection for Point-of-Care Using Chaotic Maps-Based Authentication and Key Agreement. *J. Med. Syst.* **2018**, *42*, 250. [CrossRef] [PubMed]
37. Zhang, Q.; Zhang, Q.; Gan, Y.; Wang, R.; Tan, Y.A. A Dynamic and Cross-Domain Authentication Asymmetric Group Key Agreement in Telemedicine Application. *IEEE Access* **2018**, *6*, 24064–24074. [CrossRef]
38. Sureshkumar, V.; Amin, R.; Obaidat, M.S.; Karthikeyan, I. An Enhanced Mutual Authentication and Key Establishment Protocol for TMIS Using Chaotic Map. *J. Inf. Secur. Appl.* **2020**, *53*, 102539. [CrossRef]
39. Guo, R.; Shi, H.; Zheng, D.; Jing, C.; Zhuang, C.; Wang, Z. Flexible and Efficient Blockchain-Based ABE Scheme with Multi-Authority for Medical on Demand in Telemedicine System. *IEEE Access* **2019**, *7*, 88012–88025. [CrossRef]
40. Kocarev, L. Chaos-Based Cryptography: A Brief Overview. *IEEE Circuits Syst. Mag.* **2001**, *1*, 6–21. [CrossRef]
41. Kocarev, L.; Lian, S. *Chaos-Based Cryptography—Theory, Algorithms and Applications*; Springer: Berlin, Germany, 2011; Volume 354.
42. Solev, D.; Janjic, P.; Kocarev, L. Introduction to Chaos. In *Studies in Computational Intelligence*; Springer: Berlin, Germany, 2011; Volume 354, pp. 1–25.
43. Dachselt, F.; Schwarz, W. Chaos and Cryptography. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2001**, *48*, 1498–1509. [CrossRef]
44. Kocarev, L.; Tasev, Z. Public-Key Encryption Based on Chebyshev Maps. In Proceedings of the 2003 International Symposium on Circuits and Systems, Bangkok, Thailand, 25–28 May 2003; pp. III28–III31.
45. Mishkovski, I.; Kocarev, L. Chaos-Based Public-Key Cryptography. In *Studies in Computational Intelligence*; Springer: Berlin, Germany, 2011; Volume 354, pp. 27–65.
46. Yoon, E.J.; Jeon, I.S. An Efficient and Secure Diffie–Hellman Key Agreement Protocol Based on Chebyshev Chaotic Map. *Commun. Nonlinear Sci. Numer. Simul.* **2011**, *16*, 2383–2389. [CrossRef]
47. Yoon, E.J.; Yoo, K.Y. Cryptanalysis of Group Key Agreement Protocol Based on Chaotic Hash Function. *IEICE Trans. Inf. Syst.* **2011**, *E94-D*, 2167–2170. [CrossRef]
48. Lin, H.Y. Improved Chaotic Maps-Based Password-Authenticated Key Agreement Using Smart Cards. *Commun. Nonlinear Sci. Numer. Simul.* **2015**, *20*, 482–488. [CrossRef]
49. Lee, T.F.; Hsiao, C.H.; Hwang, S.H.; Lin, T.H. Enhanced Smartcard-Based Password-Authenticated Key Agreement Using Extended Chaotic Maps. *PLoS ONE* **2017**, *12*, e0181744. [CrossRef]
50. Lin, T.W.; Hsu, C.L. Anonymous Group Key Agreement Protocol for Multi-Server and Mobile Environments Based on Chebyshev Chaotic Maps. *J. Supercomput.* **2018**, *74*, 4521–4541. [CrossRef]
51. Zhang, L. Cryptanalysis of the Public Key Encryption Based on Multiple Chaotic Systems. *Chaos Solitons Fractals* **2008**, *37*, 669–674. [CrossRef]
52. Burrows, M.; Abadi, M.; Needham, R. A Logic of Authentication. *ACM Trans. Comput. Syst. (TOCS)* **1990**, *8*, 18–36. [CrossRef]
53. Mishkovski, I.; Kocarev, L. Chaos-Based Public-Key Cryptography. In *Chaos-Based Cryptography: Theory, Algorithms and Applications*; Kocarev, L., Lian, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 27–65. [CrossRef]
54. Amin, R.; Islam, S.H.; Biswas, G.P.; Khan, M.K.; Kumar, N. A Robust and Anonymous Patient Monitoring System Using Wireless Medical Sensor Networks. *Future Gener. Comput. Syst.* **2018**, *80*, 483–495. [CrossRef]
55. Han, L.; Tan, X.; Wang, S.; Liang, X. An Efficient and Secure Three-Factor Based Authenticated Key Exchange Scheme Using Elliptic Curve Cryptosystems. *Peer-to-Peer Netw. Appl.* **2018**, *11*, 63–73. [CrossRef]
56. Liu, B.; Yang, B.; Su, X. An Improved Two-Way Security Authentication Protocol for RFID System. *Information* **2018**, *9*, 86. [CrossRef]

57. Tan, Z. Secure Delegation-Based Authentication for Telecare Medicine Information Systems. *IEEE Access* **2018**, *6*, 26091–26110. [CrossRef]
58. Armando, A.; Basin, D.; Boichut, Y.; Chevalier, Y.; Compagna, L.; Cuellar, J.; Drielsma, P.H.; Heám, P.C.; Kouchnarenko, O.; Mantovani, J.; et al. The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. In Proceedings of the Lecture Notes in Computer Science, Edinburgh, UK, 6–10 July 2005; pp. 281–285.
59. Sutrala, A.K.; Das, A.K.; Kumar, N.; Reddy, A.G.; Vasilakos, A.V.; Rodrigues, J.J.P.C. On the Design of Secure User Authenticated Key Management Scheme for Multigateway-Based Wireless Sensor Networks Using ECC. *Int. J. Commun. Syst.* **2018**, *31*, e3514. [CrossRef]
60. Alshahrani, M.; Traore, I. Secure Mutual Authentication and Automated Access Control for IoT Smart Home Using Cumulative Keyed-Hash Chain. *J. Inf. Secur. Appl.* **2019**, *45*, 156–175. [CrossRef]
61. Oheimb, D.v. The High-Level Protocol Specification Language HLPSL Developed in the EU Project AVISPA. In Proceedings of the APPSEM 2005 Workshop, Frauenchiemsee, Germany, 12–15 September 2005.
62. Bergamo, P.; D'Arco, P.; De Santis, A.; Kocarev, L. Security of Public-Key Cryptosystems Based on Chebyshev Polynomials. *IEEE Trans. Circuits Syst. I Regul. Papers* **2005**, *52*, 1382–1393. [CrossRef]
63. Wang, X.; Zhao, J. An Improved Key Agreement Protocol Based on Chaos. *Commun. Nonlinear Sci. Numer. Simul.* **2010**, *15*, 4052–4057. [CrossRef]
64. Lin, N.; Zhu, H.F. Enhancing the Security of Chaotic Maps-Based Password-Authenticated Key Agreement Using Smart Card. *J. Inf. Hiding Multimed. Signal Process.* **2017**, *8*, 1273–1282.
65. Madhusudhan, R.; Nayak, C.S. A Robust Authentication Scheme for Telecare Medical Information Systems. *Multimed. Tools Appl.* **2019**, *78*, 15255–15273. [CrossRef]
66. Healthcare Certification Authority. Available online: https://hca.nat.gov.tw/Default.aspx (accessed on 8 September 2020).
67. AVISPA: Automated Validation of Internet Security Protocols and Applications. Available online: http://www.avispa-project.org/ (accessed on 30 May 2020).