# An Advanced Anonymous and Biometrics-based Multi-server Authentication Scheme Using Smart Cards

Chin-Chen Chang[1], Wei-Yuan Hsueh[2], and Ting-Fang Cheng[1]

*(Corresponding author: Chin-Chen Chang)*

Department of Information Engineering and Computer Science, Feng Chia University[1]
No. 100, Wenhwa Rd., Seatwen, Taichung, Taiwan, 40724, R.O.C.
Department of Computer Science and Information Engineering, National Chung Cheng University[2]
No.168, Sec. 1, University Rd., Min-Hsiung Township, Chiayi, Taiwan, 62102, R.O.C.
(Email: alan3c@gmail.com)

*(Received July 8, 2014; revised and accepted Jan. 16, 2015)*

## Abstract

We find that Chuang and Chen's biometrics-based multi-server authentication scheme is unable to resist against stolen smart card and forgery attacks; in addition, their scheme has weak biometrics detection and privacy preservation problems. Thus, in this paper, we propose an advanced biometrics-based authentication scheme for a multi-server environment with higher security and efficiency. Our scheme not only resists potential attacks but satisfies various additional requirements as well. Compared with related biometrics-based schemes, our scheme not only ensures security but also has lower computation cost. In particular, our scheme overcomes the false negative problem in biometrics detection.

*Keywords: Authentication, biometrics, false negative, key agreement, multi-server*

## 1 Introduction

Due to the development of the Internet and the convenience it provides, network identity authentication has become an important security issue that can authenticate the validity of any two communication parties on the Internet. Lamport's design [11] in 1981 was the first remote user authentication scheme in an insecure environment; however, in this scheme, the remote server has to store a verification table in order to authenticate the validity of users, which may risk the leakage of users' confidential information. After that, some researchers [1, 10, 16] proposed smart-card-based authentication schemes to achieve mutual authentication between a user and server. By applying the smart card mechanism, the server no longer needs to store the verification table in its database; on the contrary, most of verification parameters, such as users' personal information and secret parameters, are stored in the smart card. Users can use their own smart card to generate and send request messages to the server, which allows the server to recognize the validity of user. Traditionally, smart-card-based remote authentication has depended upon the verification of a user's identity and password; however, a user's identity can be easily ascertained by anyone. Furthermore, a user often tends to choose a short, simple, easy-to-remember, and auto-correlated string as his/her password (e.g., telephone number, birthday, or commemoration day); thus, the user's password may be guessed by someone such as a close friend or colleague. In order to reduce the risk of the user identity or password being compromised, several researchers have begun to employ biometric information as part of the verification of user validity, because biometric information (e.g. iris, fingerprint) is unique to each user and hard for others to guess or obtain. Since 2002, more and more studies have combined individuals' biometric information to achieve user authentication using smart cards [4, 12, 13]. However, all of the schemes in [4, 12, 13] are designed for the single-server architecture.

Taking into account the diversification of services, users may want to access different services from different service providers, which may cause users to register an account for each service provider in the single-server environment. Thus, He et al. [7] proposed a smart-card-based authentication scheme for the multi-server environment which allow users to register the system only one time for access to all service providers in the system. Besides, in order to reduce the risk of the user identity or password being compromised, some biometrics-based authentication schemes for the multi-server environment are proposed [3, 6, 17, 18]; however, Yang and Yang's [17] biometric password-based multi-server authentication scheme has heavy computation cost because it applies so many modular exponentiation operations. In addition, in regard to Yoon and Yoo's scheme [18], He [6] found that it is vulnerable to privileged-insider attack, masquerade attack, and stolen smart card attack. Recently, Chuang and Chen [3] pointed out a common weakness of most biometrics-based schemes is inefficiency in addition to some security problems; thus, they proposed an improved scheme with higher efficiency and security under the assumption that all registered servers are trusted. However, both Choi [2] and Mishra [14] found that Chuang and Chen's scheme cannot resist various attacks and fails to preserve the forward secrecy.

In our work, we find that Chuang and Chen's scheme [3] also suffers from the stolen smart card attack and forgery attack as well as a privacy preservation problem. Besides, their scheme has improper biometric error detection based on hash function, which may cause a serious false negative problem such that a valid user cannot successfully log in and access servers. Therefore, in this paper, we propose a more secure and efficient version that not only resists well-known attacks but satisfies the essentials for a well-designed multi-server authentication scheme. In particular, compared with Chuang and Chen's scheme [3] and other biometrics-based schemes [4, 12, 13, 17, 18], our scheme overcomes the false negative problem in biometrics detection by adopting the functions defined in Dodis et al.'s literature [5], which feature fault tolerance in biometrics information.

The remainder of this paper is organized as follows. In Section 2, we briefly review Chuang and Chen's biometrics-based multi-server authentication scheme using smart cards and discuss its weaknesses. Section 3 introduces some requirements that our proposed scheme needs to achieve and reviews Dodis et al.'s secure sketch definitions used in our proposed scheme. Subsequently, our advanced biometrics-based multi-server authentication scheme is provided in Section 4, and security analyses of our proposed scheme are discussed in Section 5, followed by comparisons of relevant schemes in Section 6. Finally, our conclusions are shown in Section 7.

## 2 Review and Cryptanalyses of Chuang and Chen's Scheme

In 2014, Chuang and Chen [3] pointed out that the common weakness of most biometrics-based schemes is inefficiency in addition to some security problems. Hence, they proposed a hash-based scheme with higher efficiency as shown in Figure 1. Furthermore, the security of their scheme is based on the assumption that all registered servers are trusted. Though Chuang and Chen claimed that their scheme can enhance several security properties, we find that their scheme still has some security flaws. Thus, in this section, we analyze and describe its vulnerabilities as follows. Note that Chuang and Chen claimed that, in their scheme, an authorized server could be trusted under the assumption of trust computing; hence, herein, we also do not consider the possibility of a server being dishonest.

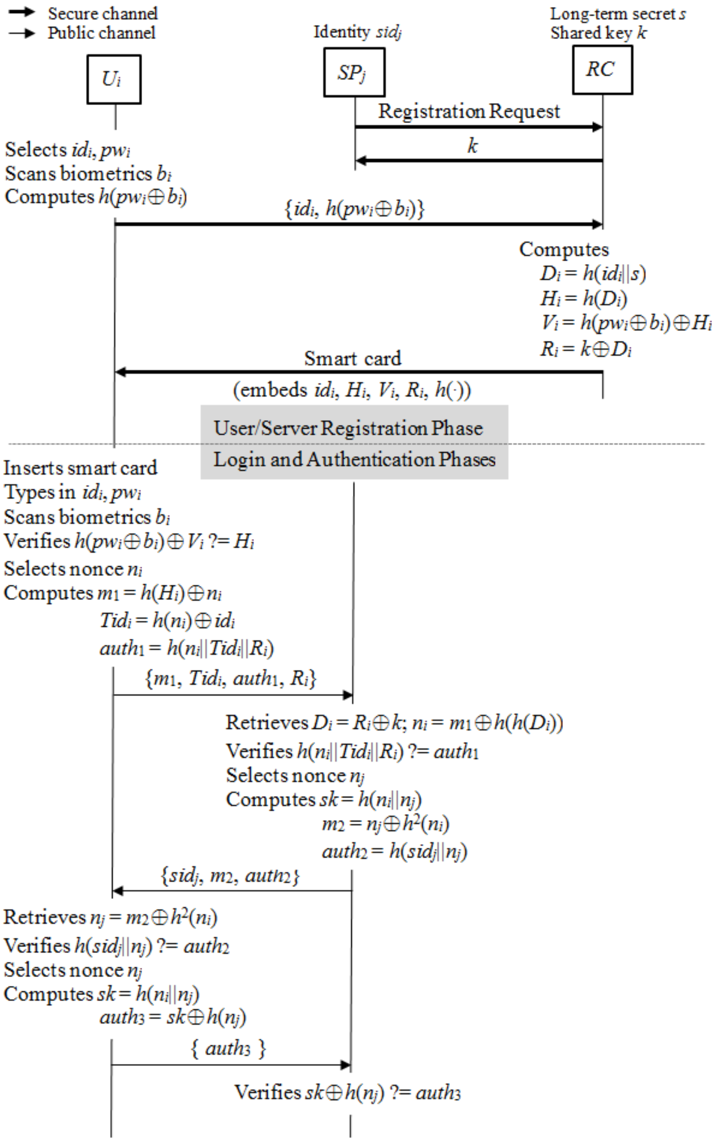
Secure channel
Public channel
Identity $sid_j$
Long-term secret $s$
Shared key $k$
$U_i$
$SP_j$
$RC$
Registration Request
$k$
Selects $id_i, pw_i$
Scans biometrics $b_i$
Computes $h(pw_i \oplus b_i)$
$\{id_i, h(pw_i \oplus b_i)\}$
Computes
$D_i = h(id_i \| s)$
$H_i = h(D_i)$
$V_i = h(pw_i \oplus b_i) \oplus H_i$
$R_i = k \oplus D_i$
Smart card
(embeds $id_i$, $H_i$, $V_i$, $R_i$, $h(\cdot)$)
User/Server Registration Phase
Login and Authentication Phases
Inserts smart card
Types in $id_i, pw_i$
Scans biometrics $b_i$
Verifies $h(pw_i \oplus b_i) \oplus V_i$ ?= $H_i$
Selects nonce $n_i$
Computes $m_1 = h(H_i) \oplus n_i$
$Tid_i = h(n_i) \oplus id_i$
$auth_1 = h(n_i \| Tid_i \| R_i)$
$\{m_1, Tid_i, auth_1, R_i\}$
Retrieves $D_i = R_i \oplus k$; $n_i = m_1 \oplus h(h(D_i))$
Verifies $h(n_i \| Tid_i \| R_i)$ ?= $auth_1$
Selects nonce $n_j$
Computes $sk = h(n_i \| n_j)$
$m_2 = n_j \oplus h^2(n_i)$
$auth_2 = h(sid_j \| n_j)$
$\{sid_j, m_2, auth_2\}$
Retrieves $n_j = m_2 \oplus h^2(n_i)$
Verifies $h(sid_j \| n_j)$ ?= $auth_2$
Selects nonce $n_j$
Computes $sk = h(n_i \| n_j)$
$auth_3 = sk \oplus h(n_j)$
$\{ auth_3 \}$
Verifies $sk \oplus h(n_j)$ ?= $auth_3$

Figure 1: Review of Chuang and Chen's scheme

### 2.1 Stolen Smart Card Attack

In [3], Chuang and Chen claimed that if someone knows a valid user's parameters stored in the user's smart card, he/she cannot forge a valid message to pass authentication; however, if an attacker, $U_A$, steals the card from a $U_i$ in some way and retrieves the information stored in it, he/she can easily forge a valid message and impersonate the user to log in to the system by the following procedures.

**Step 1.** $U_A$ extracts the parameters $\{id_i, H_i, V_i, R_i, h(\cdot)\}$ from the smart card.

**Step 2.** $U_A$ chooses a random nonce $n_A$ and computes $m_{1A} = h(H_i) \oplus n_A$, $Tid_i = h(n_A) \oplus id_i$, and $auth_{1A} =$

$h(n_A||Tid_i||R_i)$, then $U_A$ sends the fake login request message $\{m_{1A}, Tid_i, auth_{1A}, R_i\}$ to $SP_j$.

**Step 3.** After receiving the message from $U_A$, $SP_j$ retrieves $D_i = R_i \oplus k$ and $n_A = m_{1A} \oplus h(h(D_i))$ by using its secret $k$ shared with $RC$. Then $SP_j$ authenticates the validity of $U_A$ by checking $h(n_A||Tid_i||R_i)? = auth_{1A}$. It is obvious that the condition holds; thus $SP_j$ is convinced that $U_A$ is a valid user and accepts the login request.

**Step 4.** $SP_j$ randomly chooses a nonce $n_j$ and computes $m_2 = n_j \oplus h^2(n_A)$ and $auth_2 = h(sid_j||n_j)$. Then, $SP_j$ sends a reply message $\{sid_j, m_2, auth_2\}$ to $U_A$.

**Step 5.** Upon receiving the reply message from $SP_j$, $U_A$ can easily retrieve $n_j = m_2 \oplus h^2(n_A)$ and verify whether $h(sid_j||n_j)$ is equal to the received $auth_2$. Obviously, the condition is satisfied; thus, $U_A$ computes $auth_{3A} = h(n_A||n_j) \oplus h(n_j)$ and sends $auth_{3A}$ back to $SP_j$.

**Step 6.** Obviously, the reply message $auth_{3A}$ can pass the verification by $SP_j$ after it has received this message. Until now, the attacker $U_A$ has successfully logged in to $SP_j$ and negotiated a session key $sk = h(n_A||n_j)$ with $SP_j$ for future communication.

As aforementioned in regard to attack procedures, in [3], anyone who obtains a lost smart card can easily forge a valid message to pass the authentication, share a common session key, and access the services in the system. Moreover, as discussed in [14], an attacker can use a stolen smart card and intercepted messages to mount a server spoofing attack.

### 2.2 Forgery Attack

If an attacker, $U_A$, is a valid but malicious user, he/she can use his/her smart card and the parameters $\{id_A, H_A, V_A, R_A, h(\cdot)\}$ stored in it to mount a forgery attack without using his/her real identity $id_A$ via the following procedures.

**Step 1.** $U_A$ generates a fake identity $id_f$ with the same length as the output of the hash function $h(\cdot)$. Then, $U_A$ chooses a random nonce $n_A$ and computes $m_{1A} = h(H_A) \oplus n_A$, $Tid_f = h(n_A) \oplus id_f$, and $auth_{1A} = h(n_A||Tid_A||R_A)$. Finally, $U_A$ sends a fake login message $\{m_{1A}, Tid_f, auth_{1A}, R_A\}$ to $SP_j$.

**Step 2.** Upon receiving the message from $U_A$, $SP_j$ retrieves $D_A = R_A \oplus k$ and $n_A = m_{1A} \oplus h(h(D_A))$ by using its secret $k$. Then $SP_j$ computes and verifies whether the equation $h(n_A||Tid_f||R_A) = auth_{1A}$ holds or not.

Obviously, the verification would be successful. Because $SP_j$ does not verify the validity of $Tid_f$, it cannot detect a forgery attack that uses a forged identity. Afterwards, by performing Steps 3 to 6 in Subsection 2.1, $U_A$ and $SP_j$ can complete mutual authentication and share a common session key for future communication. Furthermore, as mentioned in [14], an attacker also can use a lost smart card to forge a fake message to log in to the server.

### 2.3 Hash Function Problem in Terms of Biometrics

In Chuang and Chen's scheme, a user's smart card stores the parameter $V_i = h(pw_i \oplus b_i) \oplus H_i$, which includes the user's biometric information $b_i$ scanned at the time when the user registered with $RC$. In the login phase of their scheme, the smart card verifies the validity of the card holder by checking $h(pw_i \oplus b_i^*) \oplus V_i \overset{?}{=} H_i$, where $b_i^*$ is the biometric information scanned at this time. As advocated by Chuang and Chen [3], Figure 1, a valid card holder would pass the verification; however, in fact, a hash function is sensitive and free from collision, and the biometric information scanned by the same user each time may be slightly different. That is, the mapping from input to output of a hash operation is one-to-one, so a subtle change of input must impact the output of the hash operation. As a result, it would be unsuitable to use the hash function to detect the biometric information, as it may prevent a valid user from passing the authentication in Chuang and Chen's scheme.

### 2.4 Non-provision of User Privacy

In [3], Chuang and Chen indicated that the information stored in a smart card is extractable. Based on this assumption, anyone can obtain a user's real identity directly from the user's smart card because the identity is stored inside it.

On the other hand, if an attacker, $U_A$, wants to trace the locations or information related to a specific user, he/she may collect all transmitted messages from different sessions. In login and authentication phases of Chuang and Chen's scheme, as shown in Figure 1, we find that a user always transmits the same parameter $R_i$ in each session. Hence, an attacker can monitor the transmitted $R_i$ of each session to trace a specific user even if he/she does not know the user's actual identity from the message $\{m_1, Tid_i, auth_1, R_i\}$ publicly transmitted in Login Phase.

As a result, in Chuang and Chen's method, users are not guaranteed their privacy.

## 3 Preliminaries

Here, we introduce the essentials that must be achieved by a well-designed multi-server authentication scheme using smart cards, and the definition used in our proposed scheme using biometrics.

### 3.1 Requirements

In order to design a secure and efficient smart-card-based multi-server authentication scheme, the following six considerations must be satisfied.

1) No verification table: A registration center should not store any verification table in its database for the security consideration.

2) Single registration: This is the major property that distinguishes a multi-server system from a single-server system. For convenience, users only need to register with the registration center one time; then they can access the services from any service provider in a multi-server system.

3) Freely choose password: Users can freely choose and change their passwords without requiring the involvement of a registration center, in order to decrease the system load.

4) Mutual authentication and session key agreement: Users and service providers need to authenticate each other in order to prevent security problems and negotiate a common session key and thereby keep their communications secret.

5) Security: The designed authentication scheme should not only withstand various attacks but also avoid the synchronization problem. In addition, it also should preserve user privacy.

6) Efficiency: Since a smart card cannot support heavy computation in general, the computation load of the smart card must be made as low as possible.

### 3.2 Secure Sketch

As discussed in Subsection 2.3, a hash function is sensitive and free from collision, and the biometric information scanned by the same user may be slightly different each time, so a hash function is not capable of detecting the validity of biometric information. In order to overcome the problem in [3], in our proposed scheme, we adopt the functions defined by Dodis et al. [5] to deal with related operations of biometric information.

In 2004, Dodis et al. [5] defined that an $(M, m, m', t)$-secure sketch is a randomized map $SS : M \rightarrow \{0,1\}^*$, where $m$ is min-entropy and $m'$ is the lower bound of average min-entropy. One of its properties is as follows:

For any given vector $b' \in M$ satisfying $dis(b, b') \leq t$, there is a deterministic recovery function $Rec$ such that $Rec(b', SS(b)) = b$, where $dis$ is a distance function. Because a sketch does not reveal the information about $b$ and it needs to give another value $b'$ close to $b$, the design is secure.

Based on this definition, we can set SS as an $(M, m, m+k-n, t)$-secure sketch and $SS(B) = SS(B; X) = B \oplus E(X)$ for any given $[n, k, 2t+1]$ error-correcting code $E$, where $B$ is uniform, $X$ is random variable, $n$ is the length of strings, $k$ is the dimension of the code, and $t$ is the number of tolerated errors. Also, there is a decoder $D$ of the code $E$, which can correct up to $t$ errors, such that $D(B' \oplus SS(B; X)) = X$ if $dis(B, B') \leq t$. As a result, the recovery function $Rec$ can be set as $Rec(B', SS(B; X)) = SS(B; X) \oplus E(D(B' \oplus SS(B; X))) = B$.

## 4 Proposed Scheme

In this section, in order to solve the weaknesses of Chuang and Chen's scheme discussed in Section 2, and to achieve greater security, we present an advanced anonymous and biometrics-based multi-server authentication scheme using smart cards. The notations used in our proposed scheme are listed in Table 1.

Table 1: Notations

| Item | Description |
|---|---|
| $U_i$ | A user $i$ |
| $SP_j$ | A service provider $j$ |
| $RC$ | A trusted registration center |
| $id_i$ | The identity of $U_i$ |
| $sid_j$ | The identity of $SP_j$ |
| $pw_i$ | The password of $U_i$ |
| $b_i$ | The biometric information of $U_i$ |
| $x$ | The secret key of $RC$ |
| $y$ | The secret number of $RC$ |
| $E(\cdot)$ | The encoding function based on Dodis et al.'s definition [5] (i.e., the error-correcting code in Subsection 3.2) |
| $D(\cdot)$ | The decoding function based on [5] (i.e., the decoder in Subsection 3.2) |
| $h(\cdot)$ | A secure one-way hash function |

As in Chuang and Chen's scheme, there are three kinds of participants: users ($U_i$'s), service providers ($SP_j$'s), and a trusted registration center ($RC$). One of responsibilities of $RC$ is to manage all service providers; the other is to assign a smart card for each legitimate user $U_i$ who has registered with $RC$ successfully. Once a user obtains the smart card from $RC$, he/she can use it and his/her personal information, such as identity, password, and biometrics, to log in to the system and access services provided by service providers $SP_j$'s in this system. Accordingly, our proposed scheme consists of five phases: server registration, user registration, login, authentication, and password change. Additionally, in the system initialization, $RC$ generates its secret key $x$ and a secret number $y$. The first four phases are illustrated in Figure 2.

### 4.1 Server Registration Phase

If a server $SP_j$ wants to become an authorized server, it needs to send a registration request to $RC$. Once $RC$ accepts the application provided by this server, it uses its secret key $x$ and the secret number $y$ to compute $k_1 =$

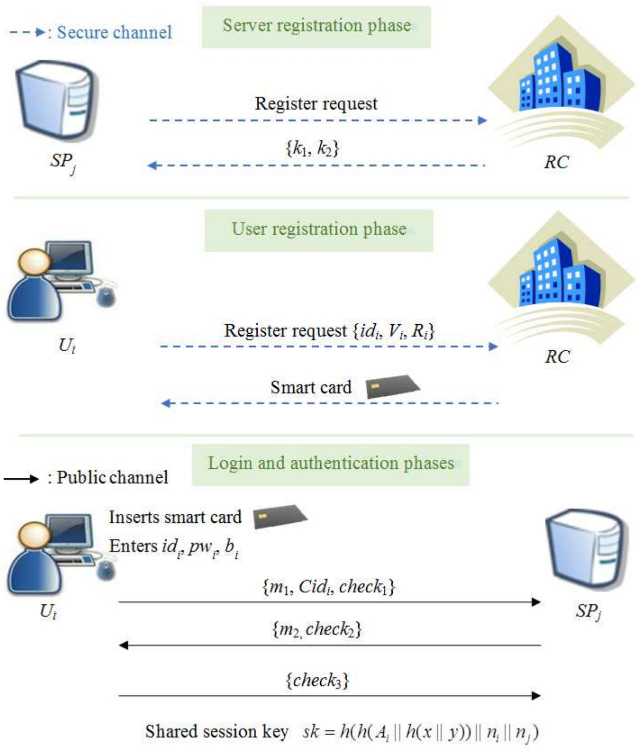
- - ▸: Secure channel
Server registration phase
Register request
{$k_1$, $k_2$}
$SP_j$
RC
User registration phase
$U_i$
Register request {$id_i$, $V_i$, $R_i$}
RC
Smart card
→ : Public channel
Login and authentication phases
Inserts smart card
Enters $id_i$, $pw_i$, $b_i$
$U_i$
$SP_j$
{$m_1$, $Cid_i$, $check_1$}
{$m_2$, $check_2$}
{$check_3$}
Shared session key $sk = h(h(A_i \,||\, h(x \,||\, y)) \,||\, n_i \,||\, n_j)$

Figure 2: The flowchart of our proposed scheme

$h(sid_j||h(y))$ and $k_2 = h(x||y)$, then sends them back to the $SP_j$ via a secure channel, where $sid_j$ is the identity of $SP_j$. Afterwards, the server $SP_j$ and $RC$ share the secrets $k_1$ and $k_2$. Note that, in this system, each authorized server holds a unique secret $k_1$, which cannot be known by the others even though they have the same secret $k_2$. This is because that the secrets $x$ and $y$ are only known by $RC$.

### 4.2 User Registration Phase

In the course of the system's operation, a user $U_i$, who wants to access the resources of service providers in this system, must first register an account with $RC$. Then $RC$ assigns a smart card embedded with some essential secret parameters to $U_i$. The detailed steps of user registration are described as follows.

**Step R1.** $U_i$ generates his/her identity $id_i$ and a password $pw_i$, and scans personal biometric information $b_i$ (e.g., fingerprint) into the specific device. Then, $U_i$ chooses a random number $r_i$ to compute $\alpha_i = b_i \oplus E(r_i)$, $V_i = h(pw_i) \oplus \alpha_i$, and $R_i = h(pw_i \oplus r_i)$, and submits a registration request message $\{id_i, V_i, R_i\}$ to $RC$ via a secure channel.

**Step R2.** After receiving the registration request from $U_i$, $RC$ computes five parameters for $U_i$: $A_i = h(id_i||x)$, $B_i = h(id_i||R_i)$, $C_i = h^2(R_i) \oplus h(y)$, $D_i = h(R_i) \oplus A_i \oplus h(x||y)$, and $E_i = h(A_i||h(x||y)) \oplus h(R_i)$.

**Step R3.** $RC$ stores the secret parameters $\{V_i, B_i, C_i, D_i, E_i, h(\cdot)\}$ into a smart card and sends it to the user $U_i$ via a secure channel.

### 4.3 Login Phase

Once a user has been assigned a smart card from $RC$, he/she can use it to access any service at any time from the system by logging in to the corresponding server. Assume that a user $U_i$ wants to log in to a server $SP_j$. First, he/she has to insert his/her smart card into a card reader, type in his/her $id_i^*$ and $pw_i^*$, and scan personal biometrics information $b_i^*$ into a specific device. Then, the smart card performs the following operations.

**Step L1.** The smart card computes $R_i^* = h(pw_i^* \oplus D(V_i \oplus h(pw_i^*) \oplus b_i^*))$ and verifies $h(id_i^*||R_i^*) \stackrel{?}{=} B_i$. If the verification is satisfied, as mentioned in Subsection 3.2, it indicates that the inputted $b_i^*$ is close to the registered $b_i$ in the user registration phase, and both inputted $id_i$ and $pw_i$ are correct. More precisely, the smart card is convinced that $U_i$ is really the card holder and proceeds to the next step.

**Step L2.** The smart card randomly generates a nonce $n_i$ and computes $h(y) = C_i \oplus h^2(R_i^*)$, $m_1 = h(sid_j||h(y)) \oplus n_i$, $Cid_i = D_i \oplus h(R_i^*) \oplus h(n_i)$, $G_i = E_i \oplus h(R_i^*)$, and $check_1 = h(h(sid_j||h(y))||n_i||G_i)$.

**Step L3.** The smart card sends $\{m_1, Cid_i, check_1\}$ as a login request message to $SP_j$.

### 4.4 Authentication Phase

Upon the login request message, $SP_j$ and $U_i$ execute the following steps to complete the mutual authentication and session key agreement.

**Step A1.** $SP_j$ retrieves the nonce $n_i = m_1 \oplus k_1$ using its secret $k_1$ shared with $RC$, and checks the freshness of $n_i$. If the nonce $n_i$ is fresh, $SP_j$ subsequently uses the retrieved $n_i$, the received $Cid_i$ from $U_i$, and its secret $k_2$ shared with $RC$ to compute $A_i = Cid_i \oplus h(n_i) \oplus k_2$. Afterwards, $SP_j$ computes and verifies whether $h(k_1||n_i||h(A_i||k_2))$ is equal to the received $check_1$. If the verification is failed, $SP_j$ rejects the login request; otherwise, it confirms that $U_i$ is valid and proceeds to the next step.

**Step A2.** $SP_j$ randomly generates a nonce $n_j$ and computes $m_2 = n_j \oplus n_i \oplus k_1$, $sk = h(h(A_i||k_2)||n_i||n_j)$, and $check_2 = h(sk)$. At last, $SP_j$ sends a reply message $\{m_2, check_2\}$ to the user $U_i$.

**Step A3.** After receiving the reply message from $SP_j$, the smart card first retrieves the nonce $n_j$ by computing $n_j = m_2 \oplus h(sid_j||h(y)) \oplus n_i$ and checks the freshness of $n_j$. If the nonce $n_j$ is fresh, the smart card then uses it to compute the session key $sk = h(G_i||n_i||n_j)$ and verifies $h(sk) \stackrel{?}{=} check_2$. If the above authentication is satisfied, the smart card ensures the validity of $SP_j$, which has received the correct $n_i$. Finally, the smart card computes and sends $check_3 = h(sk||n_j)$ to $SP_j$.

**Step A4.** Upon receiving $check_3$, $SP_j$ computes and verifies whether $h(sk||n_j) = check_3$. If the equation holds, it indicates that $U_i$ not only is a legal user but also has received the correct $n_j$ generated by it; otherwise, the session is aborted.

Now, $U_i$ and $SP_j$ have shared a common session key $sk$ such that they can use it to protect their future communication before the user logs out.

### 4.5 Password Change Phase

At any moment, if a user $U_i$ wants to change his/her password, he/she needs to insert his/her smart card into a card reader, submit $id_i$ and $pw_i$, and scan personal biometrics information $b_i$ into a specific device for changing his/her old password $pw_i$ to a new one $pw_i^{new}$. Then, the smart card executes the following steps.

**Step P1.** The smart card computes $\alpha_i = V_i \oplus h(pw_i)$, $r_i = D(b_i \oplus \alpha_i)$, and $R_i = h(pw_i \oplus r_i)$, and verifies whether $h(id_i||R_i)$ equals to the $B_i$ stored in it. If they are equal, the smart card asks $U_i$ to type in a new password; otherwise, the password change request is refused.

**Step P2.** After $U_i$ types in his/her new password $pw_i^{new}$, the smart card uses it to compute

$$\begin{aligned}
V_i^{new} &= V_i \oplus h(pw_i) \oplus h(pw_i^{new}),\\
R_i^{new} &= h(pw_i^{new} \oplus r_i),\\
B_i^{new} &= h(id_i||R_i^{new}),\\
C_i^{new} &= C_i \oplus h^2(R_i) \oplus h^2(R_i^{new}),\\
D_i^{new} &= D_i \oplus h(R_i) \oplus h(R_i^{new}),\\
E_i^{new} &= E_i \oplus h(R_i) \oplus h(R_i^{new}).
\end{aligned}$$

**Step P3.** Lastly, the smart card replaces $V_i$, $B_i$, $C_i$, $D_i$, and $E_i$ with $V_i^{new}$, $B_i^{new}$, $C_i^{new}$, $D_i^{new}$, and $E_i^{new}$ stored inside it. Now, the user's password has been successfully changed without the help of $RC$.

## 5 Security Analyzes

In this section, we analyze the resistance to various attacks and the achievement of security requirements in our proposed scheme. Assume that an attacker, $U_A$, exists in the system who can not only control the whole public communication channel between users and service providers but also intercept, eavesdrop, or tamper with any transmitted message. We consider various different scenarios to provide detailed analyzes of our scheme below.

### 5.1 Resistance to Off-line Password Guessing Attack

Here, we illustrate two possible attackers' behaviors that would lead to an off-line password guessing attack as follows:

1) Stolen smart card of a user

   If the attacker, $U_A$, has stolen a legal user's ($U_i$'s) smart card, he/she may extract the parameters $\{V_i, B_i, C_i, D_i, E_i, h(\cdot)\}$ from it and try to mount an off-line password guessing attack; however, knowing the parameters in the card does not impact the security of $U_i$'s password at all.

   For $V_i$, it is calculated by the equation $V_i = h(pw_i) \oplus \alpha_i = h(pw_i) \oplus b_i \oplus E(r_i)$, where $b_i$ is $U_i$'s unique biometric information, and $r_i$ is a random number secretly chosen by $U_i$. Actually, $V_i$ belongs to a two-factor protection on $r_i$. In the regular login process of our scheme (i.e., Step L1), only inputting the correct $pw_i$ and $b_i$ can retrieve the random number $r_i$ by computing $D(V_i \oplus h(pw_i) \oplus b_i)$. Note that there is no way to obtain or forge the user's unique biometrics information $b_i$. In other words, we also can treat $V_i$ as a two-factor protection on $pw_i$. It is hard for $U_A$ to detect whether he/she has guessed the correct password without the knowledge of $b_i$ and $r_i$.

   Similarly, for other stored parameters $B_i$, $C_i$, $D_i$, and $E_i$, their expressions all contain more than one element that $U_A$ does not know in addition to the user's password $pw_i$, such as the user's identity $id_i$, long-term secrets $x$ and $y$ of $RC$, and the random number $r_i$. Hence, we can treat these parameters as multi-factor protections. It is hard for $U_A$ to detect whether he/she has guessed the correct password without the knowledge of these elements. That is, $id_i$ and $r_i$ in $B_i = h(id_i||R_i)$; $r_i$ and $y$ in $C_i = h^2(R_i) \oplus h(y)$; and $r_i$, $id_i$, $x$, and $y$ in $D_i = h(R_i) \oplus A_i \oplus h(x||y)$ and $E_i = h(A_i||h(x||y)) \oplus h(R_i)$. Consequently, our proposed scheme prevents the attacker from being about to guess a valid user's password in polynomial time from a user's stolen smart card.

2) Intercepting transmitted messages between users and service providers

   If the attacker, $U_A$, intends to guess a specific user's password by intercepting messages transmitted between the user and an $SP_j$ over the Internet, he/she will fail. This is because that, in our scheme, none of the transmitted messages are related to a user's password. As a result, it is impossible for an attacker to mount an off-line password guessing attack by collecting data transmitted over the Internet.

As discussed above, our proposed scheme can resist an off-line password guessing attack.

## 5.2 Resistance to Forgery Attack

Herein, to explain the resistance to forgery attack, we consider two cases: the general view and the privileged-insider view.

1) The general view
If the attacker, $U_A$, attempts to forge a valid message in order to log in to $SP_j$, he/she needs to forge the login messages as $m_1 = h(sid_j||h(y)) \oplus n_A$, $Cid_A = A_A \oplus h(n_A) \oplus k_2$, and $check_1 = h(h(sid_j||h(y))||n_A||h(A_A||k_2))$, where $n_A$ and $A_A$ are two random numbers chosen by him/her; however, it is difficult for an attacker to forge a valid login message without the long-term secrets $k_1$ and $k_2$ of the server. Even if the attacker knows the identity $sid_j$ of the server, he/she still cannot know the secret $k_1$ without the knowledge of $h(y)$.

   On the other hand, the attacker may intercept a legal user's ($U_i$'s) login message $\{m_1, Cid_i, auth_1\}$, where $m_1 = h(sid_j||h(y)) \oplus n_i$, $Cid_i = A_i \oplus h(n_i) \oplus k_2$, and $check_1 = h(h(sid_j||h(y))||n_i||h(A_i||k_2))$, and try to forge another valid login message from it in order to impersonate that user. That is, $U_A$ has to retrieve the user's information $A_i$ from the intercepted message, to generate a nonce $n_A$, and to compute a fake login message $m_1^* = h(sid_j||h(y)) \oplus n_A$, $Cid_i^* = A_i \oplus h(n_A) \oplus k_2$, and $check_1^* = h(h(sid_j||h(y))||n_A||h(A_i||k_2))$ to $SP_j$. Obviously, it is computationally infeasible for $U_A$ to retrieve the user's information $A_i$ without the knowledge of the long-term secrets $k_1$ and $k_2$ of the server. Consequently, the forged message will be refused by $SP_j$ in Step A1. Hence, $U_A$ cannot forge a valid login message to impersonate the user $U_i$.

   In addition, as mentioned in Subsection 5.1, most parameters stored in a smart card have the characteristic of multi-factor protection. Even if $U_A$ has stolen a legal user's smart card and extracted the secret parameters $\{V_i, B_i, C_i, D_i, E_i, h(\cdot)\}$ from it, he/she still cannot forge a valid message to log in to $SP_j$ without knowing $h(y)$, $pw_i$, $r_i$, and $U_i$'s biometric information $b_i$.

2) The privileged-insider view
If the attacker, $U_A$, is a valid but untrusted user, he/she may use his/her own parameters $\{V_A, B_A, C_A, D_A, E_A, h(\cdot)\}$ stored in his/her smart card to conduct the following forgery attack. $U_A$ first computes $R_A = h(pw_A \oplus D(V_A \oplus h(pw_A) \oplus b_A))$ by using his/her $pw_A$ and $b_A$, then $U_A$ uses $R_A$ and stored $C_A$ to retrieve $h(y)$ by computing $h(y) = h^2(R_A) \oplus C_A$. Once $U_A$ obtains the secret $h(y)$, he/she may try to mount a forgery attack, as mentioned in the third paragraph of Case 1, by taking a stolen smart card of a valid user $U_i$ and retrieving the secrets $\{V_i, B_i, C_i, D_i, E_i, h(\cdot)\}$ from it. Fortunately, as shown in the previous case (i.e., case 1), it is also impossible for $U_A$ to forge a valid login message $\{m_1^*, Cid_i^*, auth_1^*\}$ to masquerade as another legal user without the knowledge of their $pw_i$ and $b_i$, so the fake login message can still be detected by $SP_j$ in Step A1.

   The same holds, if $U_A$ is a valid but dishonest service provider who has stolen a smart card of $U_i$ and extracted the stored information. Though $U_A$ additionally knows the secrets $k_1 = h(sid_j||h(y))$ and $k_2 = h(x||y)$, he/she still has no way to calculate $R_i = h(pw_i \oplus D(V_i \oplus h(pw_i) \oplus b_i))$ without the knowledge of $pw_i$ and $b_i$. Hence, $U_A$ still cannot impersonate a user by stealing a smart card to send a legal request in Step L3.

As mentioned above, our proposed scheme does not suffer from a forgery attack.

## 5.3 Resistance to Server Spoofing Attack

In regard to a server spoofing attack, if the attacker, $U_A$, tries to cheat a user, $U_i$, who sends a login request, he/she needs to forge and reply a valid message to $U_i$; however, $U_A$ cannot impersonate a legal server, $SP_j$, to send a valid message since he/she does not have the secrets $k_1$ and $k_2$ of $SP_j$ to extract the correct $n_i$ and $A_i$ from the intercepted login message $\{m_1, Cid_i, auth_1\}$ of $U_i$ in Step A1. Hence, $U_A$ cannot succeed in this attempt.

On the other hand, if the attacker is a valid service provider, $SP_A$, he/she still cannot masquerade as another service provider, $SP_j$, to fool a user, $U_i$, by forging an acknowledgement message in Step A2. The reason is that even though each valid server holds the same $k_2 = h(k||y)$, it is not the same as holding $k_1 = h(sidj||h(y))$, a shared key only known to $SP_j$ and $RC$, is not equal. Thus, $SP_A$ is unable to reply a valid message to $U_i$ without knowing $SP_j$'s $k_1$. As a result, there is no way for $SP_A$ to impersonate another server to communicate with users.

In addition, we assume that the malicious $SP_A$ has got a user's ($U_b$'s) smart card and tries to know the $k_1$ of another $SP_j$ by using the retrieved parameters $\{V_b, B_b, C_b, D_b, E_b, h(\cdot)\}$ from the stolen card. The main purpose is to obtain the partial secret $h(y)$ due to $k_1 = h(sid_j||h(y))$, where $sid_j$ is the public identity of $SP_j$ and $y$ is a long-term secret of $RC$; however, it is computationally hard for $SP_A$ to retrieve $h(y)$ from the stolen card, which can only be acquired by the card holder who has the correct password and personal biometric information. Thus, $SP_A$ is unable to obtain other servers' keys $k_1$'s without $h(y)$ to fool users.

No matter how the attacker masquerades as another server, his/her attempts will fail. As a result, our proposed scheme can resist a server spoofing attack.

### 5.4 Resistance to Stolen Smart Card Attack

As mentioned in Subsections 6.2 and 6.3, no matter whether the attacker is an outsider or not, he/she cannot use the stolen smart card to impersonate any other person or server. Furthermore, the attacker cannot know the actual owner of the stolen card in addition to the password guessing attack in Subsection 5.1. Hence, stealing a user's smart card does not enable an attacker to acquire the user's private information and perform illegal behavior. That is to say, our proposed scheme can fully prevent a stolen smart card attack.

### 5.5 Resistance to Stolen-verifier Attack

With regard to this attack, the attacker may try to steal the verification table stored in the database of $RC$ in order to engage in illegal behavior, such as a user impersonation attack in order to access the services of service providers; however, our scheme needs not to be worry about the stolen-verifier attack by adopting smart-card-based authentication. In our scheme, $RC$ does not store any verification table regarding users' accounts; thus, the attacker is unable to gain the information that would be used to impersonate another user successfully. In other words, the attacker cannot succeed in such an attempt by using the process discussed in Subsection 5.2. As a result, a stolen-verifier attack is infeasible in our proposed scheme.

### 5.6 Resistance to Privileged-insider Attack

It is possible that there an inside attacker, $U_A$, may have the right to obtain the message sent from users in the user registration phase over the secure channel or information from the database of the registration center. Though the system claims that it can be trusted, a privileged-insider may be able to use his/her privileges to obtain a user's identity, password, and biometric information; however, in our scheme, the user $U_i$ sends $\{id_i, V_i, R_i\}$ as a registration request as shown in Step R1 of the registration phase, where $V_i = h(pw_i) \oplus b_i \oplus E(r_i)$, $R_i = h(pw_i \oplus r_i)$, and $r_i$ is a random number. The password and biometric information are protected by a secure one-way hash function, which means that it is computationally infeasible for an insider to know a user's password and biometric information directly. Furthermore, as discussed in Subsection 5.1, a user's password cannot be guessed by others except for the user himself/herself. In addition, each user's biometric information is unique and held by himself/herself; thus, the insider has difficulty obtaining users' biometric information. Therefore, even if there is a privileged member in the system, he/she is unable to obtain a user's private information and engage in wrongdoing.

### 5.7 Resistance to Replay Attack

Our scheme is free from the replay attack, because we adopt random nonces instead of timestamps that could result in a time synchronization problem. During the protocol run, the freshness of transmitted messages $\{m_1, Cid_i, check_1\}$ from a user and $\{m_2, check_2\}$ from a service provider must be verified by checking $n_i$ and $n_j$ in Steps A1 and A3, respectively. If the attacker replays the message intercepted in the previous session, our scheme will quickly detect that the involved random nonce is invalid, because the random nonce must be different in each session. Obviously, the replayed message would not pass authentication in our scheme. As a result, our proposed scheme can withstand the replay attack.

### 5.8 No Hash Function Problem in Terms of Biometrics

As discussed in Subsection 2.3, Chuang and Chen adopted the hash function to detect a user's biometric information, but this would lead to a serious false negative problem as a valid user may not pass verification by using his/her own smart card. In order to overcome this problem, in our scheme, we do not use the hash function but rather encoding and decoding functions defined by Dodis et al. [5] to detect a user's biometric information. These encoding and decoding functions have fault tolerance in biometric information, as mentioned in Subsection 3.2, which allows a user to pass authentication even if the biometric information that he/she scanned each time is slightly different from the original one scanned during the registration phase.

Hence, our improved scheme can avoid such a problem in detecting users' biometric information.

### 5.9 Preservation of Known-key Security

In our proposed scheme, the session key is computed as $sk = h(h(A_i||k_2)||n_i||n_j)$, which involves $A_i = h(id_i||x)$, $k_2 = h(x||y)$, and the random nonces $n_i$ and $n_j$ chosen by $U_i$ and $SP_j$ in each session, respectively. The nonces are delivered in the form $m_1 = h(sid_j||h(y)) \oplus n_i = k_1 \oplus n_i$ and $m_2 = n_j \oplus n_i \oplus k_1 = n_j \oplus n_i \oplus h(sid_j||h(y))$, which implies that only the valid $SP_j$ and $U_i$ have the secret $k_1$ and $h(y)$ can retrieve the correct nonces $n_i$ and $n_j$. Furthermore, we assume that the long-term secrets $x$ and $y$ of $RC$ are only known by $RC$ itself and are not stored in the verification table or in users' smart cards based on Shannon's and Kerckhoffs' Theorems [8, 9, 15]. Consequently, the attacker cannot obtain the long-term secrets of $RC$ from its database or a user's smart card. In other words, if the $k$th session key is compromised accidentally by an attacker, it will not reveal the confidential content of messages nor the session keys negotiated in previous and following sessions, because the session key is changed by the nonces $n_i$ and $n_j$.

Table 2: Security comparisons of our proposed scheme with relevant schemes

| Property | Ours | [3] | [18] | [17] | [4] | [13] | [12] |
|---|---|---|---|---|---|---|---|
| Off-line password guessing attack | Yes | Yes | Yes | Yes | No | Yes | Yes |
| Forgery attack | Yes | No | No | Yes | No | Yes | Yes |
| Server spoofing attack | Yes | No | Yes | No | No | Yes | Yes |
| Stolen smart card attack | Yes | No | No | Yes | No | Yes | Yes |
| Stolen-verifier attack | Yes | Yes | Yes | Yes | Yes | No | Yes |
| Privileged-insider attack | Yes | Yes | No | No | No | No | No |
| Replay attack | Yes | Yes | Yes | Yes | Yes | No | Yes |
| No hash function problem in terms of biometrics | Yes | No | Yes | Yes | Yes | No | No |

Yes: The scheme actually satisfies the property or resists the attack;
No: The scheme does not satisfy the property.

As a result, our proposed scheme can achieve known-key security in the session key establishment.

### 5.10 User Privacy Preservation

Here, we discuss various aspects of user privacy preservation:

1) Privacy of user's identity and location
If the attacker, $U_A$, wants to know a user's ($U_i$'s) identity, he/she may capture the transmitted message $\{m_1, Cid_i, check_1\}$ sent by $U_i$, and try to retrieve $U_i$'s identity from $Cid_i = D_i \oplus h(R_i) \oplus h(n_i) = h(id_i||x) \oplus h(x||y) \oplus h(n_i)$ or $check_1 = h(h(sid_j||h(y))||n_i||G_i) = h(h(sid_j||h(y))||n_i||h(h(id_i||x)||h(x||y)))$. In our scheme, however, $U_i$ communicates with $SP_j$ anonymously such that even $SP_j$ cannot know the user's actual identity. Besides, the identity involved in $Cid_i$ and $check_1$ is protected by a secure one-way hash function as well as the secrets $x$ and $y$ of $RC$. It is computationally infeasible for the attacker to retrieve $U_i$'s real identity, $id_i$, from $Cid_i$ and $check_1$. Similarly, even if $U_A$ has obtained a user's smart card and extracted the stored secrets $\{V_i, B_i, C_i, D_i, E_i, h(\cdot)\}$ from it, he/she still cannot know the user's actual identity, $id_i$.

In addition, even if $U_A$ focuses on tracking $U_i$'s location without knowing $U_i$'s identity, he/she is unable to do so. This is because the message transmitted from a user is generated dynamically in each session by adopting a random nonce. Hence, it is difficult for an attacker to track a specific user's real location.

As a result, our scheme can preserve the anonymity and untraceability of users.

2) Privacy of user's biometric information
If $U_A$ wants to know a user's biometric information, he/she needs to obtain the user's smart card and try to retrieve the user's biometric information from $V_i$ stored in the smart card; however, the user's biometric information, $b_i$, is encrypted by the encoding function as shown in Step R1; thus, $U_A$ cannot obtain $b_i$ directly. Besides, it is hard for $U_A$ to guess the correct $b_i$ in polynomial time from the stored parameters of the stolen smart card, since each user holds unique biometric information. Consequently, the privacy of a user's biometric information is guaranteed.

## 6 Comparisons

In this section, we provide security, functionality, and performance comparisons of our scheme with other biometrics-based authentication schemes [3, 4, 12, 13, 17, 18].

First, we summarize security comparisons between our scheme and other related schemes in Table 2. As discussed in Section 5, our scheme can prevent the listed attacks and avoid the hash function problem in terms of biometrics. In regard to an off-line password guessing attack, all of the related schemes can resist it except for [4]. In [4], if an attacker has obtained a user's smart card and extracted the stored secrets in some way, then he/she can easily guess the user's password in polynomial time. In [3, 4, 18], an attacker can forge a valid message to cheat servers by using the stolen smart card of a valid user as well as intercepted messages, so these schemes are unable to resist forgery and stolen smart card attacks. Similarly, in regard to a server spoofing attack, in [3, 4], an attacker can use a stolen smart card and intercepted messages to impersonate a valid server and send a legal message to a user. In [17], however, since each server holds the same secret shared with the $RC$, a valid but dishonest server can impersonate another server to communicate with users. In Li et al.'s scheme [13], because this scheme stores the verification table in database, it cannot prevent a stolen-verifier attack. Besides, in their scheme, users and the server never check the freshness of transmitted messages during their verification procedures, so their scheme cannot resist a replay attack.

Table 3: Functionality comparisons of our proposed scheme with relevant biometrics-based authentication schemes

| Functionality | Ours | [3] | [18] | [17] | [4] | [13] | [12] |
|---|---|---|---|---|---|---|---|
| No verification table | Yes | Yes | Yes | Yes | No | No | Yes |
| Single registration | Yes | Yes | Yes | Yes | N/A | N/A | N/A |
| Mutual authentication | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Preservation of known-key security | Yes | Yes | Yes | Yes | N/A | Yes | N/A |
| Privacy of user's identity and location | Yes | No | No | No | No | No | No |
| Privacy of user's biometric information | Yes | Yes | No | Yes | No | No | Yes |
| No time synchronization | Yes | Yes | Yes | No | Yes | Yes | Yes |
| Quick error detection | Yes | Yes | No | Yes | Yes | No | No |
| Freely choose and change password | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Session key agreement | Yes | Yes | Yes | Yes | No | Yes | No |

Yes: The scheme actually satisfies the functionality;
No: The scheme does not satisfy the functionality;
N/A: The scheme does not consider or be applicable to the functionality.

It is noteworthy that the schemes in [4, 12, 13, 17, 18] cannot prevent a privileged-insider attack. In these schemes, a privileged-insider can know a user's identity, password, and biometric information directly, because the user sends his/her personal information as the registration request in plaintext form to the $RC$. Once the privileged-insider obtains a user's identity, password, and biometric information, he/she can impersonate the user to do anything. Furthermore, the schemes in [3, 12, 13] have a hash function problem in terms of biometrics, because these schemes use the hash function to detect whether a user's biometric information is correct or not, which will lead to a serious false negative problem where a valid user may be denied verification as discussed in Subsection 2.3.

Second, Table 3 summarizes functional comparisons between our scheme and other relevant schemes. All of schemes provide mutual authentication. As analyzed in Section 5, our proposed scheme achieves all functionality requirements. In [4, 13], because each user's biometric template is stored in the system, both schemes cannot achieve the no verification table requirement. Consequently, these schemes cannot preserve the privacy of users' biometric information. In [4, 12, 13], schemes were designed for single-server architecture, so the property of single registration is inapplicable to them.

Particularly, in regard to the privacy preservation of a user's identity and location, all of the related schemes [3, 4, 12, 13, 17, 18] are unable to preserve this property because, in these schemes, a user's identity can be obtained from a stolen smart card or transmitted messages. In Yoon and Yoo's scheme [18], a user's biometric template is stored in the user's smart card directly. Once an attacker gets the smart card, he/she can easily obtain the user's biometric information. Hence, the scheme in [18] cannot preserve the privacy of users' biometric information.

Furthermore, all of the schemes adopt the nonce mechanism rather than timestamps to resist the replay attack except for Yang and Yang's scheme [17]. Thus, only Yang and Yang's scheme suffers from the time synchronization problem. In regard to quick error detection in [12, 13, 18], a smart card can verify biometrics quickly but cannot detect the password in time, because it has to wait for the server to authenticate the messages in order to know the correctness of the password. Thus, these schemes do not really achieve this property.

Lastly, performance comparisons of the login and authentication phases of our proposed scheme and other relevant schemes are shown in Table 4. Considering that the computation cost of smart cards is limited, attention should be paid to the performance analysis of the proposed scheme. First, let us define the notations used in Table 4. $T_h$ is the computation time for performing a one-way hash function once; $T_D$ refers to the computation time of one decoding operation based on Dodis et al.'s definition [5]; $T_{ecc}$ refers to the computation time of one elliptic curve operation; $T_e$ indicates the computation time of one modular exponentiation operation; and $T_f$ indicates the computation time for executing fuzzy extractor once. In addition, we ignore the cost of the exclusive-OR operation, because its time complexity is much lower than the above operations. On the other hand, as cost implementation in [3], the order of time complexity is $T_e >> T_{ecc} >> T_h$. We assume that the costs of $T_D$ and $T_f$ are low.

Table 4 shows that, in our proposed scheme, the total computation cost of login and authentication phases is $17T_h+1T_D$. For the multi-server environment, our scheme obviously is more efficient than Yoon and Yoo's [18] and Yang and Yang's [17] schemes; however, our scheme costs a little more than Chuang and Chen's [3]. It is still reasonable, though, because our scheme can use the encoding and decoding function defined in [5] to greatly reduce the false negative problem related to biometrics error de-

Table 4: Performance comparisons of the login and authentication phases of our proposed scheme and other relevant schemes

| Participant | Ours | [3] | [18] | [17] | [4] | [13] | [12] |
|---|---|---|---|---|---|---|---|
| Registration center | X | X | $7T_h$ | X | X | X | X |
| Service provider | $6T_h$ | $8T_h$ | $2T_{ecc}+$ $5T_h$ | $3T_e+$ $3T_h$ | $5T_h$ | $6T_h$ | $3T_h$ |
| User | $11T_h+$ $1T_D$ | $9T_h$ | $2T_{ecc}+$ $5T_h$ | $2T_e+$ $5T_h+$ $T_f$ | $5T_h$ | $7T_h$ | $4T_h$ |
| Total | $17T_h+$ $1T_D$ | $17T_h$ | $4T_{ecc}+$ $17T_h$ | $5T_e+$ $8T_h+$ $T_f$ | $10T_h$ | $13T_h$ | $7T_h$ |

X: There is no computation cost for this entity in the login and authentication phases.

tection. Moreover, although our scheme has higher cost than the schemes in [4, 12, 13], our scheme supports the multi-server environment, resists most potential attacks, and has more functionalities, as shown in Tables 2 and 3. Therefore, it is worth increasing the cost in order to provide more functionalities and higher security.

As a result, our scheme not only ensures security but also maintains functionality and efficiency better than other biometrics-based schemes.

## 7 Conclusions

In this work, we find that Chuang and Chen's scheme cannot resist stolen smart card and forgery attacks and cannot guarantee user privacy. In particular, their scheme has an improper design in regard to biometrics error detection. Thus, we propose an improved biometrics-based multi-server authentication scheme using smart cards. As shown in our security analyses and comparisons, the proposed scheme not only remedies the flaws of Chuang and Chen's scheme but also prevents vulnerability to various attacks and achieves the necessary requirements. Furthermore, our proposed scheme has lower computational cost used for authentication, as shown in Table 4. Consequently, our proposed scheme is not only suitable for applying biometrics detection but also is efficient and robust against most security attacks.

## References

[1] H. Y. Chien, J. K. Jan, and Y. M. Tseng, "An efficient and practical solution to remote authentication: smart card," *Computers and Security*, vol. 21, no. 4, pp. 372–375, 2002.

[2] Y. Choi, "Security enhanced anonymous multi-server authenticated key agreement scheme using smart card and biometrics," *IACR Cryptology ePrint Archive*, pp. 1–11, 2014.

[3] M. C. Chuang and M. C. Chen, "An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1411–1418, 2014.

[4] A. K. Das, "Analysis and improvement on an efcient biometric-based remote user authentication scheme using smart cards," *IET Information Security*, vol. 5, no. 3, pp. 145–151, 2011.

[5] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'04)*, LNCS 3027, pp. 523–540, Springer, 2004.

[6] D. He, "Security flaws in a biometrics-based multi-server authentication with key agreement scheme," *IACR Cryptology ePrint Archive*, pp. 1–9, 2011.

[7] D. He, W. Zhao, and S. Wu, "Security analysis of a dynamic ID-based authentication scheme for multi-server environment using smart cards," *International Journal of Network Security*, vol. 15, no. 5, pp. 350–356, 2013.

[8] A. Kerckhoffs, "La cryptographie militaire," *Journal des Sciences Militaires*, vol. January, no. IX, pp. 5–38, 1883.

[9] A. Kerckhoffs, "La cryptographie militaire," *Journal des Sciences Militaires*, vol. February, no. IX, pp. 161–191, 1883.

[10] W. C. Ku and S. M. Chen, "Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 204–207, 2004.

[11] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[12] C. T. Li and M. S. Hwang, "An efcient biometrics-based remote user authentication scheme using smart

cards," *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 1–5, 2010.

[13] X. Li, J. W. Niu, J. Ma, W. D. Wang, and C. L. Liu, "Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 73–79, 2011.

[14] D. Mishra, "Cryptanalysis of multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics," arXiv e-print service, Cornell University, pp. 1–8, 2014.

[15] C. E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, pp. 656–715, 1949.

[16] W. H. Yang and S. P. Shieh, "Password authentication schemes with smart cards," *Computers and Security*, vol. 18, no. 8, pp. 727–733, 1999.

[17] D. Yang and B. Yang, "A biometric password-based multi-server authentication scheme with smart card," in *Proceedings of IEEE 2010 International Conference on Computer Design and Applications (ICCDA'10)*, pp. 25–27, Qinhuangdao, China, 2010.

[18] E. J. Yoon and K. Y. Yoo, "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem," *Journal of Supercomputing*, vol. 63, no. 1, pp. 235–255, 2010.

**Chin-Chen Chang** received his Ph.D. degree in computer engineering from National Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in computer and decision sciences. Both were awarded in National Tsing Hua University. Dr. Chang served in National Chung Cheng University from 1989 to 2005. His title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. He is a Fellow of IEEE and a Fellow of IEE, UK. His research interests include database design, computer cryptography, image compression and data structures.

**Wei-Yuan Hsueh** received the B.S. degree in computer science and information engineering from National United University, Miaoli, Taiwan in 2012. She received the M.S. degree in computer science and information engineering from National Chung Cheng University, Chia-Yi, Taiwan. Her current research interests include electronic commerce, information security, cryptography, and wireless communications.

**Ting-Fang Cheng** received her Ph.D. degree in computer science from National Tsing Hua University, Hsinchu, Taiwan in 2013. She received the B.S. and M.S. degrees in information engineering and computer science from Feng Chia University, Taichung, Taiwan in 2005 and 2007, respectively. Now she is as a Postdoctoral Fellow in information engineering and computer science, Feng Chia University, Taichung, Taiwan. Her current research interests include electronic commerce, information security, cryptography, mobile communications, cloud computing, and information hiding.