Research Article

An Improved Biometric-Based User Authentication Scheme for C/S System

Li Jiping,1 Ding Yaoming,1 Xiong Zenggang,1 and Liu Shouyin2

1 School of Computer and Information Science, Hubei Engineering University, Xiaogan 432000, China
2 College of Physical and Technology, Central China Normal University, Wuhan 430079, China

Correspondence should be addressed to Ding Yaoming; xgdym@21cn.com

Received 24 August 2013; Revised 18 February 2014; Accepted 27 February 2014; Published 27 April 2014

Academic Editor: Chuan-Ming Liu

Copyright © 2014 Li Jiping et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The authors first review the recently proposed Das’s biometric-based remote user authentication scheme, and then show that Das’s scheme is still insecure against some attacks and has some problems in password change phase. In order to overcome the design flaws in Das’s scheme, an improvement of the scheme is further proposed. Cryptanalysis shows that our scheme is more efficient and secure against most of attacks; moreover, our scheme can provide strong mutual authentication by using verifying biometric, password as well as random nonces generated by the user and server.

1. Introduction

In a client/server system, the validity of remote user is necessary to assure the security of the system. Traditional remote identity-based authentication schemes [1–9] are based on passwords only. However, simple passwords are always easy to break by using simple dictionary attacks since they have low entropy. To overcome this problem, cryptographic secret keys and passwords are used in the remote user authentication schemes. But the long and random cryptographic keys are difficult to memorize and hence they must be stored somewhere [10]; it is expensive to maintain the long cryptographic keys. Furthermore, both passwords and cryptographic keys are unable to provide nonrepudiation because they can be forgotten or lost or when they are shared with other people, there is no way to know who the actual user is [11]. A biometric system operates by acquiring biometric data from an individual, extracting a feature set from the acquired data and comparing this feature set against the template set in the database [12–14]. In [3], the authors propose an online $(t,n)$ threshold secret sharing scheme based on biometric verification and threshold password authentication. In [15], a continuous user authentication scheme based on biometric verification by fusing hard and soft traits is proposed irrespective of user posture in front of the system. In [16], a BIO3G protocol based on biometric authentication for 3G mobile environments is proposed to provide real end to end strong user authentication. In [17], the author analyzes the security of Das’s biometric-based authentication scheme and shows that the scheme is still insecure against some attacks and does not provide mutual authentication between the user and server. In [18], the author proposes an enhanced scheme based on biometric verification and smart card to remove the security weakness of Das’s scheme analyzed in [17]. However, the proposed scheme in [18] cannot withstand the replay attack between the user and the remote server. In the abovementioned schemes, biometric verification allows one to confirm or establish an individual identity. Therefore, biometric keys are proposed which are based on physiological and behavioral characteristics of persons such as fingerprints, faces, irises, hand geometry, and palm prints. Some advantages of biometric keys are described as follows [19, 20].

(i) Biometric keys cannot be lost or forgotten.
(ii) Biometric keys are very difficult to copy or share.
(iii) Biometric keys are extremely hard to forge or distribute.
(iv) Biometric keys cannot be guessed easily.
(v) Someone’s biometrics is not easier to break than others.
Table 1: Notations used in the proposed scheme.

| Notation | Description |
|----------|-------------|
| $C_i$    | Client      |
| $R_i$    | Trusted registration center |
| $S_i$    | Server      |
| PW$_i$   | Password shared between $C_i$ and $S_i$ |
| ID$_i$   | Identity of the user $C_i$ |
| $B_i$    | Biometric template of the user $C_i$ |
| $d(*)$   | Symmetric parametric function |
| $\tau$  | Predetermined threshold for biometric verification |
| $h(*)$  | A secure one-way hash function |
| $X_s$    | A secret information maintained by the server |
| $R_s$    | A random number chosen by $C_i$ |
| $r_s$    | A random number chosen by $S_i$ |
| $A \parallel B$ | Data $A$ concatenates with data $B$ |
| $A \oplus B$ | XOR operation of $A$ and $B$ |

As a result, biometric-based remote user authentications are inherently more reliable and secure than usual traditional password-based remote user authentication schemes.

In this paper, we propose an improvement of Da’s biometric-based remote user authentication scheme using smart cards in order to withstand his design flaws. The remainder of this paper is organized as follows. In Section 2, we briefly review the Da’s biometric-based remote user authentication scheme using smart cards [21]. In Section 3, we analyze the design flaws in Da’s scheme. In Section 4, we propose an improvement of the scheme in order to eliminate the design flaws discussed in Section 3. Security analysis of our scheme and performance comparison with other related schemes are implemented in Section 5. Finally, we conclude the paper in Section 6.

2. Review of Da’s Biometric-Based Remote User Authentication Scheme

In this section, we review in brief Da’s biometric-based remote user authentication scheme [21]. For describing the Da’s scheme [21], we use the notations shown in Table 1. Da’s scheme consists of the following four phases, namely, registration phase, login phase, authentication phase, and change password phase. Details of each phase are given in the following subsections.

2.1. Registration Phase. In order to login to the system, the remote user $C_i$ needs to perform the following stages, as shown in Algorithm 1.

Step 1. The user inputs his/her personal biometric $B_i$ on a specific device and offers his/her password PW$_i$ and the identity ID$_i$ of the user to the registration center $R_i$ in person.

Step 2. The registration center $R_i$ then computes $f_i = h(B_i)$, $r_i = h(PW_i) \oplus f_i$, and $e_i = h(ID_i \parallel X_i) \oplus r_i$. Here $X_i$ is secret information generated by the server.

Step 3. Finally, the registration center $R_i$ loads (ID$_i$, h(·), $f_i$, $e_i$, $r_i$) on the user’s smart card and sends the information to the user $C_i$ via a secure channel.

2.2. Login Phase. In this phase, if a user $C_i$ wants to login to the server $S_i$, he/she needs to perform the following steps, as shown in Algorithm 2.

Step 1. $C_i$ first inserts his/her smart card into the smart card reader of a terminal and offers his/her personal biometric template, $B_i$, on the specific device to verify his/her biometric.

Step 2. Next, the user’s personal biometric template $B_i$ is matched against the template stored in the system.

Step 3. If the above verification does not hold, then $C_i$ does not pass the biometric verification and, as a result, the remote user authentication is terminated. Otherwise, on the other hand, if the abovementioned verification holds, $C_i$ passes the biometric verification and $C_i$ then inputs his/her password PW$_i$ to perform Step 4.

Step 4. The smart card computes $r'_i = h(PW_i) \oplus f_i$. If $r'_i \neq r_i$, then password verification fails, and the client terminates the session.

Step 5. If $r'_i = r_i$, the smart card computes $M_1 = e_i \oplus r'_i$, which is equal to $h(ID_i \parallel X_i)$, $M_2 = M_1 \oplus R_s$, which is equal to $h(ID_i \parallel X_i) \parallel R_s$, and $M_3 = h(R_s)$, where $R_s$ is a random number generated by the user.

Step 6. Finally, $C_i$ sends the message (ID$_i$, $M_2$, $M_3$) to the remote server $S_i$.

2.3. Authentication Phase. After receiving the login request message (ID$_i$, $M_2$, $M_3$), $S_i$ performs the following steps, as shown in Algorithm 3, in order to authenticate whether the user $C_i$ is legal or not.

Algorithm 2: Login phase of Das’s scheme.

Algorithm 3: Authentication phase of Das’s scheme.

**Step 1.** $S_i$ first checks the format of $C_i$’s ID$_i$.

**Step 2.** If the above format is valid, $S_i$ then computes $M_4 = h$(ID$_i$ || X$_i$), $M_5 = M_4 \oplus M_3$ and then verifies whether $h(M_5) = M_3$. If it does not hold, then $S_i$ rejects $C_i$’s login request. In case the verification is successful, then $S_i$ computes $M_6 = M_4 \oplus R$, $M_7 = h(M_5 || M_3)$, and $M_8 = h(R)$. 

**Step 3.** $S_i$ then sends the message $\langle M_7, M_6, M_8 \rangle$ to $C_i$.

**Step 4.** After receiving the message in Step 3, $C_i$ verifies whether $M_7 = h(M_5 || R)$. Thus, if the verification does not pass, $C_i$ terminates the session. Otherwise, $C_i$ proceeds as follows by computing $M_9 = M_8 \oplus M_3 (= R)$ and verifying further whether $h(M_9) = M_3$. If $h(M_9) \neq M_3$, $C_i$ terminates the session. On the other hand, $C_i$ computes $M_{10} = h(M_6 || M_9)$ and sends the message $\langle M_{10} \rangle$ to the server $S_i$.

**Step 5.** After receiving $C_i$’s message, $S_i$ verifies whether $M_{10} = h(M_6 || R)$. 

**Step 6.** If the abovementioned does not hold, $S_i$ rejects $C_i$’s login request.

**Step 7.** In case the verification is successful, then only $S_i$ accepts $C_i$’s login request.

2.4. Password Change. The password change phase of Das’s scheme [21] has the following steps.
Step 1. It inserts the smart card into the card reader and offers \( B_i \).

Step 2. It verifies whether the user’s personal biometric template \( B_i \) matches against the template stored in the system.

Step 3. If \( C_i \) passes the biometric verification, then only \( C_i \) enters his/her old password \( PW_i \) and new changed password \( PW_i^{new} \).

Step 4. The smart card then computes \( r_i' = h(PW_i^{old}) \oplus f_i \); if \( r_i' \neq r_i \), the password change phase is terminated. If \( r_i' = r_i \), then only smart card computes \( r_i'' = h(PW_i^{new}) \oplus f_i \), \( e_i' = e_i \oplus r_i (= h(ID_i \| X_i)) \), and \( e_i'' = e_i' \oplus r_i \).

Step 5. Finally, replace \( e_i' \) with \( e_i'' \) and \( r_i \) with \( r_i'' \) on the smart card.

### 3. Cryptanalysis of Das’s Scheme

This section demonstrates that Das’s scheme [21] has some drawbacks: denial-of-service attack, user impersonation attack, replay attack, and password change problem.

#### 3.1. Denial-of-Service Attack

One of fundamental properties of a secure one-way hash function is that its outputs are very sensitive to small perturbations in their inputs. The cryptographic hash function cannot be applied straightforwardly when the input data are with noise such as biometrics [22]. Then the predetermined threshold for biometric verification cannot be used to measure outputs of hash functions. In the registration phase of Das’s scheme, the register center \( R \) computes \( f_i = h(B_i) \) and \( r_i = h(PW_i) \oplus f_i \) and then stores \( f_i \) and \( r_i \) in the smart card. In the login phase, \( C_i \) inserts his/her smart card into the card reader and provides his/her personal biometrics \( B_i \) on a specific device to verify the users biometrics by verifying whether \( h(B_i) = f_i \) or not. In Step 4 of login phase, password verification is performed by verifying whether \( r_i' = r_i \). However, both the biometric verification and password verification procedures may result in serious flaws because \( h(B_i) = f_i \) may never succeed, since the inputted biometrics belonging to the same person may differ slightly from time to time [22], so the next login and authentication procedure will be terminated. As a result, this may cause the legal user to be unable to pass biometric verification at the login phase of Das’s scheme. Therefore, Das’s scheme is vulnerable to the denial-of-service attack.

#### 3.2. User Impersonation Attack

We see from the login and authentication phase of Das’s scheme that an attacker can impersonate a legal user to access to the server. In the login phase of Das’s scheme, since the user \( C_i \) sends the message \( \langle ID_i, M_2, M_3 \rangle \) to the remote server \( S_i \) where \( C_i \) identity is not masked, this will result in user impersonation attack as follows.

When an attack denoted as \( A_i \) wants to access the remote server, he/she can eavesdrop the message \( \langle ID_i, M_2, M_3 \rangle \) by tapping communication lines or wireless link between the legal user \( C_i \) and the remote server \( S_i \). Once \( A_i \) acquires the message \( \langle ID_i, M_2, M_3 \rangle \) he can send the eavesdropped message to the remote server \( S_i \). Since the legal user’s ID is not masked, so the check of user’s validity can easily pass. We can clearly see that when \( S_i \) computes \( M'_i = h(ID_i \| X_i) \) and \( M'_i = M_i \oplus M'_i \), the verification of \( h(M'_i) = M_i \) is successful. Then \( S_i \) computes \( M'_i = M_i \oplus R_i, M'_i = h(M_i \| R_i) \), and \( M'_i = h(R_i) \) and then sends message \( \langle M'_i, M'_i, M'_i \rangle \) to \( C_i \). The attack \( A_i \) may eavesdrop the message \( \langle M'_i, M'_i, M'_i \rangle \) and modifies the \( M'_i \), replaces it with \( M'_i \), and then sends a forged message \( \langle M'_i, M'_i, M'_i \rangle \) to \( C_i \). Obviously \( M'_i \neq h(M_2 \| R_i) \), so \( C_i \) terminates the session. However, the attacker \( A_i \) will pass the verification \( \langle M'_i, M'_i, M'_i \rangle \) and \( A_i \) computes \( M'_i = M'_i \oplus M_i \). Since the attack \( A_i \) can verify \( M'_i \) with \( M'_i \), he proceeds as follows by computing \( M'_i = h(M_2 \| R_i) \) or not. We can see obviously that the above equation holds, so the remote server \( S_i \) accepts the attacker’s login request and the user impersonation attack will occur sequentially.

#### 3.3. Replay Attack

In Das’s scheme, the replay and man-in-the-middle attack is withstood by checking whether \( M'_i (= M_i \oplus M_i) = M_i \) where \( M_i \) is equal to \( R_i \) and is stored in the database of remote server \( S_i \). It is noted that \( M_i = M_i \oplus M_i = M_i \oplus R_i \oplus M_i = R_i(M_i = M_i) \) is disclosed to any user when one breaks the remote server \( S_i \). When the remote server \( S_i \) is compromised by an attacker, he/she can change \( \langle ID_i, M_3 \rangle \) in the database of the remote server \( S_i \). Obviously, once \( M_3 \) is changed, the replayed message \( \langle ID_i, M_3, M'_i \rangle \) will not be discarded and \( M_i \) will be replaced by \( M'_i \).

#### 3.4. Password Change

In password change procedure of Das’s scheme, if remote user \( C_i \) wants to change his/her password, he/she must pass biometric verification by verifying \( h(B_i) = f_i \). However, the inputted biometrics belonging to the same person may differ slightly from time to time [22], so the password change procedure will be terminated. In addition, for more time, since \( h(B_i) \neq f_i \), then \( r_i' = h(PW_i^{old}) \oplus f_i \) computed by smart card is not equal to \( r_i \) stored in the smart card, so the password change procedure will also be terminated. According to the above analysis, Das’s scheme cannot realize the password change freely.

### 4. Proposed Scheme

In this section, we propose an improvement of the Das’s biometric-based remote user authentication scheme [21] using smart cards in order to withstand the flaws discussed in Section 3. For convenience, we use the same notations used as in Das’s scheme shown in Table I.

#### 4.1. Registration Phase

In order to login to the system, the remote user \( C_i \) needs to perform the following steps, as shown in Algorithm 4.
Step 1. The user $C_i$ inputs his/her personal biometric $B_i$ on a specific device and offers his/her password $PW_i$ and the identity $ID_i$ to the registration center $R_i$ in person.

Step 2. The registration center $R_i$, then computes $f_i = h(B_i)$, $g_i = h(ID_i)$, $r_i = h(PW_i) \oplus f_i$, and $e_i = h(g_i \parallel X_i) \oplus r_i$. Here $X_i$ is secret information generated by the server. We note that $X_i$ and passwords of the corresponding users are not disclosed to any others for all secure future communications.

Step 3. Finally, the registration center $R_i$ loads $(h(\cdot), f_i, g_i, e_i, r_i, \tau, d(\cdot))$ on the user’s smart card and sends this information to the user $C_i$ via a secure channel.

### 4.2. Login Phase

In order to login to the system, the remote user $C_i$ needs to perform the following stages, as shown in Algorithm 5.

Step 1. $C_i$ first inserts his/her smart card into the card reader of a terminal and offers his/her personal biometric template, $B'_i$, on the specific device. If $d(B_i, B'_i) > \tau$, the remote user authentication is terminated. Otherwise, $C_i$ passes the biometric verification and then inputs his/her password $PW_i$ to perform Step 2.

Step 2. The smart card computes $r'_i = h(PW_i) \oplus f_i$. If $d(r'_i, r_i) > \tau$, then password verification fails, and the system terminates the session; otherwise, the smart card computes $M_1 = e_i \oplus r'_i$, which is equal to $h(g_i \parallel X_i)$, $M_2 = h(R_c \parallel T)$, where $R_c$ is a random number generated by the user $C_i$ and $T$ is the current timestamp of $C_i$’s system, and $M_3 = M_1 \oplus M_2$.

Step 3. Finally, the user $C_i$ sends the message $(g_i, M_2, M_3, T)$ to the remote server $S_i$.

### 4.3. Authentication Phase

When the remote server $S_i$ receives the login request $(g_i, M_2, M_3, T)$ at time $T^*$, it will perform the following steps as shown in Algorithm 6 to authenticate whether the user $C_i$ is legal or not.

Step 1. Verify $T$. If $(T^* - T) > \Delta T$, the authentication phase aborts, where $\Delta T$ is the expected time interval for the transmission delay of the system. On the contrary, if $(T^* - T) \leq \Delta T$, the next step will be performed.

Step 2. $S_i$ checks the format of $C_i$’s $ID_i$. It computes $M_4 = h(g_i \parallel X_i)$ using the secret value $X_i$ maintained by the server $S_i$ and then computes $M_5 = M_4 \oplus M_3$ and verifies whether $M_5 = M_2$. If it does not hold, then $S_i$ rejects $C_i$’s login request. In case the verification is successful, the next step will be performed.

Step 3. $S_i$ computes $M_6 = h(R_c \parallel T_s)$ and $M_7 = M_4 \oplus M_6$, where $T_s$ is the current timestamp of the server $S_i$, and then $S_i$ sends message $(M_4, M_6, M_7, T_s)$ to the user $C_i$.

Step 4. After receiving the message $(M_4, M_6, M_7, T_s)$ at time $T^{**}$, $C_i$ first checks the freshness of $T_s$ by verifying $(T^{**} - T_s) > \Delta T$. If it holds, the following session is terminated; otherwise $C_i$ computes $M_8 = M_4 \oplus M_7$ and then verifies whether $M_8 = M_6$. If it does not hold, $C_i$ terminates the session. Otherwise, it goes to the next step.

Step 5. $C_i$ computes $M_9 = M_4 \oplus M_8$ and then verifies whether $M_9 = M_7$. If it does not hold, $S_i$ is rejected by $C_i$; otherwise, if it holds, $C_i$ computes $M_{10} = h(R_c \parallel T^*)$, where $T^*$ is the current timestamp of the user $C_i$, and then computes $M_{11} = M_7 \oplus M_{10}$ and sends the message $(M_{11}, R_c, T^*)$ to the remote server $S_i$.

Step 6. When $S_i$ receives the message $(M_{11}, R_c, T^*)$ at time $T^{***}$, it verifies $(T^{***} - T^*) > \Delta T$. If it holds, the authentication phase is terminated. Otherwise, if it does not hold, $S_i$ computes $M_{12} = h(R_c \parallel T^*)$ and then computes $M_{13} = M_4 \oplus M_6 \oplus M_{12}$. After computing $M_{13}$, then $S_i$ verifies whether $M_{13} = M_{11}$. If it holds, $S_i$ accepts $C_i$’s login request; otherwise, $S_i$ rejects the login request.

### 4.4. Password Change

In our scheme, user $C_i$ can freely change the password $PW_i^{old}$ to a new one $PW_i^{new}$. The password change procedure is performed as follows.

Step 1. $C_i$ inserts the smart card into the card reader and offers his/her personal biometrics $B'_i$, then the smart card computes $f'_i = h(B'_i)$ and verifies it by checking $d(f'_i, f_i) \leq \tau$. Where $f_i = h(B_i)$ is the information stored in the smart card.

Step 2. If it holds, $C_i$ inserts old password $PW_i^{old}$ and new password $PW_i^{new}$; otherwise the password change procedure is terminated.

Step 3. Smart card performs $r'_i = h(PW_i^{old}) \oplus f'_i$ and checks $d(r'_i, r_i) \leq \tau$ where $r_i$ is the information stored in the smart card.

Step 4. If it holds, the smart card computes $r''_i = h(PW_i^{new}) \oplus f_i$, $e'_i = e_i \oplus r_i (= h(ID_i \parallel X_i))$, and $e''_i = e'_i \oplus r_i$.

Step 5. Finally, replace $e_i$ with $e''_i$ and $r_i$ with $r''_i$ on the smart card.
5. Security Analysis and Performance of the Proposed Scheme

5.1. Security Analysis. If a legal user lost his/her smart card, it is extremely hard for an adversary to derive the user's sensitive information such as user’s identity, password, and biometrics because the extraction of parameters from the smart card is quite difficult. Furthermore, the adversary cannot change the password because he/she cannot pass the biometric verification.

5.1.1. Denial-of-Service Attack. In our proposed protocol, we take into account hash function’s sensitivity to small perturbations in its inputs. In the login phase, user’s biometric verification is performed by checking $d(B_i, B'_i) > \tau$ instead of checking $h(B'_i) = f_i$. Moreover, the password verification is performed by checking $d(r'_i, r_i) > \tau$ instead of $r'_i = r_i$. So denial-of-service attack caused by hash function’s fundamental properties can be withstood.

5.1.2. Stolen-Verifier Attack. Our scheme can resist stolen-verifier attack because the scheme is free from the verifier/password table. In our protocol, the remote server $S_i$ does not keep password tables. Therefore, an attacker cannot steal user’s password from $S_i$. Moreover, the password is masked by hash function in the procedure of message transfer between the user $C_i$ and remote server $S_i$.

5.1.3. Many Logged-In Users Attack. Most systems, which maintain the password table to verify user login, are vulnerable to this kind of threat. Our scheme can resist the threat since our scheme requires on-card computation for login to the remote server $S_i$ and once the smart card is removed, the login process will be aborted.
5.1.4. Guessing Attack. Our protocol can resist guessing attack, which is a critical concern in password-based systems, since the password in our protocol is transmitted as a digest of some other secret information. The attacker cannot guess the user’s password from the digest because of the one-way characteristic of the hash function, even if the attacker may get the digest which contains the password.

5.1.5. Replay Attack. Replaying an intercepted message can be prevented in our proposed protocol. If an attacker intercepts \((ID_3, M_2, M_3, T)\) and tries to login to the remote server \(S_1\) via replaying the same message, he/she cannot pass the verification of the login request due to \((T^* - T) > \Delta T\), where \(T^*\) is the system time when the remote server \(S_1\) receives the replayed message. Moreover, if an attacker intercepts \((M_4, M_6, M_7, T')\) and tries to replay the message to the user \(C_1\), this kind of attack also can be prevented due to \((T'^* - T') > \Delta T\).

5.1.6. User Impersonation Attack. In the login phase of our scheme, the message sent to remote server \(S_1\) is \((g_i, M_2, M_3, T)\) instead of \((ID_3, M_2, M_3, T)\), where the user’s identity \(ID_3\) is masked by hash function. Even though an attacker eavesdrops the message \((g_i, M_2, M_3, T)\), he cannot derive the user’s identity, \(ID_3\), due to the one-way characteristic of hash function. In the authentication phase, when the remote server \(S_1\) receives the login request message \((g_i, M_2, M_3, T)\), it will check the validity of user’s identity. Since the attacker cannot derive legal user’s identity, the check of user’s identity cannot pass which will result in the termination of authentication phase. Through the above analysis, we can see that user impersonation attack can be avoided in our scheme.

5.1.7. Server Masquerading Attack. If an attack \(A_1\) attempts to masquerade as the legitimate server \(S_1\), he/she must make the forged replay message to the user when receiving the user’s login request message \((g_i, M_2, M_3, T)\). However, the forged replay message is more difficult to fake since the time-stamped message \((M_4, M_6, M_7, T')\) is sent to the user \(C_1\) when the remote server \(S_1\) is receiving \(C_1\)’s login request message \((g_i, M_2, M_3, T)\). Moreover, the attacker \(A_1\) cannot masquerade as the server by forging the replay message \((M_4, M_6, M_7, T)\), because \(A_1\) cannot compute \((M_4, M_7)\) sending to the user \(C_1\) without knowing the secret value \(X_s\) kept by the server \(S_1\). Hence, the attacker \(A_1\) cannot masquerade as the legal server to the user by launching the server masquerading attack.

5.1.8. Insider Attack. In the registration phase, if the user’s password \(PW_i\) and the biometrics information \(B_i\) are revealed to the server \(S_1\), the insider of the server may directly obtain \(PW_i\) and \(B_i\), and the insider impersonates as the user \(C_1\) to access the user’s other accounts in the server. But in the login phase of our scheme, if the insider wants to access \(C_1\)’s other accounts, he/she must input his/her smart card to the card reader and provide his biometric information \(B_i’\) in order to pass the verification \(d(B_i, B_i') < \tau\). Since the insider cannot provide the user \(C_1\)’s smart card, the biometric verification will be aborted. So the insider attack can be prevented.

5.1.9. Mutual Authentication. As described above, our scheme can withstand the user impersonation attack and server masquerading attack; consequently our scheme can provide mutual authentication between the user \(C_1\) and remote server \(S_j\).

5.1.10. Man-in-the-Middle Attack. Man-in-the-middle attack means that an active attacker intercepts the communication line between a legal user and the server and uses some means to successfully masquerade as both the server to the user and the user to the server. Then, the user will believe that he is talking to the intended server and vice versa. In our scheme, when a user \(C_1\) wants to login to the remote server \(S_j\), mutual authentication between the user \(C_1\) and remote server \(S_j\) is performed, so man-in-the-middle attack can be prevented.

5.2. Performance of the Proposed Scheme. In this subsection, we compare the performances of our improved scheme with those for Li-Hwang’s scheme [11] and Das’s scheme [21]. It is worth recalling that the protocol of Li-Hwang’s scheme [11] has security weaknesses against denial-of-service attack, replay attack, user impersonation attack, and man-in-the-middle attack. It is noted that Das’s scheme [21] has security weaknesses against denial-of-service attack, user impersonation attack, replay attack, server masquerading attack, and insider attack. The security comparisons between our scheme and the schemes proposed by Li and Hwang [11] and Das [21] are summarized in Table 2. For the convenience of evaluating the efficiency of related scheme, we define the notation \(T_n\), the time of executing a one-way hash function. The efficiency comparison with related schemes is shown in Table 3. From the table, we can see that our scheme is more efficient than Das’s scheme [21]. Though our scheme is less efficient than Li-Hwang’s scheme [11], it can provide better security against most attacks.

6. Conclusion
This paper presents a biometric-based user authentication scheme for client/server system. The method employs biometric keys and resists the threats of stolen-verifier, of which many are logged-in users with the same login identity, denial-of-service attack, guessing attack, insider attack, replay attack, user impersonation attack, server masquerading attack, and man-in-the-middle attack. Moreover, the improved scheme can realize mutual authentication between the user and remote server. The proposed scheme uses only hash function and XOR operation, which is efficient compared with that of related protocols. In addition, the user’s password can be changed freely using the proposed scheme. Our proposed scheme provides strong authentication with the help of verifying biometrics, passwords, and random nonces generated by the user and server as compared to that of related schemes.
Table 2: Security comparisons among related protocols.

| Item                                           | Our scheme | Li-Hwang’s scheme [11] | Das’s scheme [21] |
|------------------------------------------------|------------|------------------------|-------------------|
| Avoiding denial-of-service attack              | Yes        | No                     | No                |
| Avoiding stolen-verifier attack                | Yes        | Yes                    | Yes               |
| Avoiding many logged-in users attack           | Yes        | Yes                    | Yes               |
| Avoiding guessing attack                       | Yes        | Yes                    | No                |
| Avoiding replay attack                         | Yes        | No                     | No                |
| Avoiding user impersonation attack             | Yes        | No                     | No                |
| Avoiding server masquerading attack            | Yes        | No                     | Yes               |
| Avoiding man-in-the-middle attack              | Yes        | No                     | Yes               |
| Avoiding insider attack                        | Yes        | No                     | No                |
| Mutual authentication                          | Yes        | No                     | No                |
| Having flaws in password change                | No         | Yes                    | Yes               |

Table 3: Efficiency comparison with related schemes.

| Different phase | Li-Hwang’s Scheme [11] | Das’s scheme [21] | Our scheme |
|----------------|------------------------|-------------------|------------|
| Registration   |                        |                   |            |
| User computation cost | $2T_h$   | $4T_h$            | $4T_h$     |
| Server computation cost |          | $3T_h$            | $3T_h$     |
| Login          |                        |                   |            |
| User computation cost | $3T_h$   | $3T_h$            | $3T_h$     |
| Server computation cost |          | $3T_h$            | $3T_h$     |
| Authentication  |                        |                   |            |
| User computation cost | $2T_h$   | $T_h$             | $T_h$      |
| Server computation cost |          | $5T_h$            | $3T_h$     |

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the valuable comments and suggestions of the reviewers. This work is supported in part by National Natural Science Foundation of China (no. 61370223) and by Science Research Project of Hubei Provincial Department of Education (XD2012374 and B2013024).

References

[1] M. S. Hwang and C. Y. Liu, “Authenticated encryption schemes: current status and key issues,” International Journal of Network Security, vol. 1, no. 2, pp. 61–73, 2005.

[2] N.-Y. Lee and Y.-C. Chiu, “Improved remote authentication scheme with smart card,” Computer Standards and Interfaces, vol. 27, no. 2, pp. 177–180, 2005.

[3] C. T. Li, “An enhanced remote user authentication scheme providing mutual authentication and key agreement with Smart Cards,” in Proceedings of the 5th International IEEE Computer Society Conference on Information Assurance and Security, pp. 517–520, Xi’an, China, 2009.

[4] M. Kim and C. K. Koc, “A simple attack on a recently introduced hash-based strong-password authentication scheme,” International Journal of Network Security, vol. 1, no. 2, pp. 77–80, 2005.

[5] K. H. M. Wong, Z. Yuan, C. Iannone, and W. Shengwei, “A dynamic user authentication scheme for wireless sensor networks,” in Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC’06), pp. 244–251, Taichung, Taiwan, June 2006.

[6] H.-R. Tseng, R.-H. Jan, and W. Yang, “An improved dynamic user authentication scheme for wireless sensor networks,” in Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM ’07), pp. 986–990, Washington, DC, USA, November 2007.

[7] T. H. Lee, “Simple dynamic user authentication protocols for wireless sensor networks,” in Proceedings of the 2nd International Conference on Sensor Technologies and Application (SENSOR COMM ’08), pp. 657–660, Cap Esterel, France, August 2008.

[8] L.-C. Ko, “A novel dynamic user authentication scheme for wireless sensor networks,” in Proceedings of the IEEE International Symposium on Wireless Communication Systems (ISWCS ’08), pp. 608–612, Reykjavik, Iceland, October 2008.

[9] B. Vaidya, J. J. Rodrigues, and J. H. Park, “User authentication schemes with pseudonymity for ubiquitous sensor network in NGN,” International Journal of Communication Systems, vol. 23, no. 9-10, pp. 1201–1222, 2010.

[10] J. Daemen and R. V. Rijndael, “The advanced encryption standard,” Dr. Dobbs Journal, vol. 26, no. 3, pp. 137–139, 2001.

[11] C.-T. Li and M.-S. Hwang, “An efficient biometric-based remote user authentication scheme using smart cards,” Journal of Network and Computer Applications, vol. 33, no. 1, pp. 1–5, 2010.

[12] A. K. Jain, A. Ross, and S. Prabhakar, “An introduction to biometric recognition,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 1, pp. 4–20, 2004.

[13] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition, Springer, New York, NY, USA, 2009.

[14] S. Prabhakar, S. Pankanti, and A. K. Jain, “Biometric recognition: security and privacy concerns,” IEEE Security and Privacy, vol. 1, no. 2, pp. 33–42, 2003.

[15] A. Prakash, “A biometric approach for continuous user authentication by fusing hard and soft traits,” International Journal of Network Security, vol. 16, no. 1, pp. 65–70, 2014.
[16] C. K. Dimitriadis and S. A. Shaikh, “A biometric authentication protocol for 3G mobile systems: modelled and validated using CSP and rank functions,” International Journal of Network Security, vol. 5, no. 1, pp. 99–111, 2007.

[17] A. Yang, “Security weaknesses and improvements of a fingerprint-based remote user authentication scheme using smart cards,” International Journal of Advancements in Computing Technology, vol. 4, no. 3, pp. 21–28, 2012.

[18] A. N. Younghwa, “Security analysis and enhancements of an effective biometric-based remote user authentication scheme using smart cards,” Journal of Biomedicine and Biotechnology, vol. 2012, Article ID 519723, 6 pages, 2012.

[19] C.-H. Lin and Y.-Y. Lai, “A flexible biometrics remote user authentication scheme,” Computer Standards and Interfaces, vol. 27, no. 1, pp. 19–23, 2004.

[20] C.-T. Li and M.-S. Hwang, “An efficient biometrics-based remote user authentication scheme using smart cards,” Journal of Network and Computer Applications, vol. 33, no. 1, pp. 1–5, 2010.

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

[22] J. P. Linnartz and P. Tuyls, “New shielding functions to enhance privacy and prevent misuse of biometric templates,” in Proceedings of the Audio and Video-Based Biometric Person Authentication, vol. 2688 of Lecture Notes in Computer Science, pp. 393–402, 2003.
