Remote three-factor authentication scheme based on Fuzzy extractors

Min Zhang1,2*, Jiashu Zhang1 and Ying Zhang1

1 Sichuan Province Key Lab of Signal and Information Processing, Southwest JiaoTong University, Chengdu, Sichuan 610031, China
2 School of Foreign languages, Southwest University for Nationalities, Chengdu, Sichuan 610066, China

ABSTRACT

In order to provide high-security remote authentication, the three-factor authentication scheme combining biometric with smart card and password has been proposed. With a careful review of the recently proposed Lee–Hsu’s scheme, this paper points out some design flaws in it. In order to withstand them, a new scheme based on Fuzzy extractors is proposed. With a detail analysis, this paper demonstrates that the proposed scheme is more practical and reasonable. It also has higher security and deals with biometric more appropriately in spite of higher computation cost at client than Lee–Hsu’s scheme. Furthermore, an access control method has been introduced in it for the purpose of making different users enjoy different access privileges with regard to the data. What is more, the proposed scheme can also achieve key agreement. Copyright © 2014 John Wiley & Sons, Ltd.

KEYWORDS
biometric; remote authentication scheme; secure sketch; fuzzy extractors; access control

*Correspondence
Min Zhang, Sichuan Province Key Lab of Signal and Information Processing, Southwest JiaoTong University, Chengdu, Sichuan 610031 China, E-mail: cdc.szhangmin@gmail.com

1. INTRODUCTION

In the insecure environment, identity authentication is an essential scheme in network security because it can guarantee the conformation of two communication parties’ identity. In order to solve this problem, Lamport [1] first proposed a remote authentication scheme based on password for the insecure environment. However, Jan et al. [2] pointed out Lamport’s scheme could not resist privileged insider attack because the server must store a password list. Hwang and Li [3] suggested a remote authentication based on smart card in 2000. According to the literatures [4–8], we can know that an authentication scheme must meet the following factors as a secure and efficient remote one:

• password should be verified at the client;
• scheme should consider the Fuzzy property of biometrics;
• the server need not store users’ passwords;
• different kinds of attacks can be resisted;
• the user can change his or her password freely without communicating with the server;
• scheme should achieve mutual authentication and negotiate a session key for later communication.

Traditional remote identity-based authentication schemes are based on IDs and passwords that can be easily forgotten, lost, or attacked. What is more, these schemes become more and more inadequate in remote authentication environment. On the contrary, an opportunity for a more reliable and automated method of identity verification can be provided by biometric keys because of the following advantages [5]:

• biometric keys cannot be lost or forgotten;
• biometric keys are not easy to share or copy;
• biometric keys are difficult to fabricate or distribute; and
• biometric keys are difficult to be guessed.

In traditional biometric authentication system [9,10], the original data are extracted from the sensor to generate cancelable biometric template. Then, this template compares with the pre-stored templates that have been stored in the database at the registration phase. However, biometric template security is one of the most crucial issues in designing such an authentication scheme. Several vulnerabilities of these schemes have been proposed in the literature [11–15]. At the same time, some approaches have been proposed to overcome biometric security challenges and alleviate the biometric privacy issue [16,17], such as BioHashing [18,19] and BioPhasor [20,21].
Recently, some three-factor remote user authentication schemes have been proposed to resolve the single password and single biometric authentication problems [4–7]. In 2009, Fan and Lin [4] proposed a provably secure remote truly three-factor authentication scheme with privacy protection on biometrics. In 2010, Li and Hwang [5] proposed an efficient biometric-based remote user authentication scheme using smart card, which is more efficient compared with other schemes. Later, Li et al. [8] pointed out that Li and Hwang’s scheme does not provide proper authentication and cannot resist man-in-the-middle attacks. At the same time, Das [6] pointed out that Li and Hwang’s scheme has some flaws in login and authentication phases, password change phase, and verification of biometrics using hash function. Then, Das proposed a new scheme to resolve earlier flaws. However, Lee and Hsu [7] found that the Das’s scheme has some flaws also: privileged insider attack, offline password guessing attacks, and the identity of the user cannot become anonymous. To overcome these weaknesses, Lee and Hsu proposed a secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps. Unfortunately, we find that Lee and Hsu’s scheme has many flaws as follows:

- user’s identity be spoofed attack;
- brute-force attack to obtain password and biometric;
- three-factor may become two-factor;
- the methods of dealing biometric are not appropriate.

To overcome these weaknesses, we proposed a new three-factor scheme based on Fuzzy extractors. In recent years, the Fuzzy extractors theory has been studied widely [22–24]. Fuzzy extractors have some important advantages as follows:

- the problem of key storage can be solved easily;
- the Fuzzy property of biometric is suited to Fuzzy extractors.

From the analysis of security and performance in this paper, we can see that the proposed scheme has slightly higher computation at the client but the server has lower computation than Lee–Hsu’s scheme. What is more, the proposed scheme can not only resist common attacks but also deal with biometric more appropriately.

The rest of this paper is organized as follows. In Section 2, we review Lee and Hsu’s scheme, and in Section 3, we show the weaknesses of Lee and Hsu’s scheme. Then, the proposed scheme is presented in Section 4. We analyze the performance of this proposed scheme and show that the scheme can resist several attacks in Section 5. Our conclusion is given in Section 6.

2. THE REVIEW OF LEE AND HSU’S SCHEME

In this section, we briefly introduce the Chebyshev polynomial used in Lee–Hsu’s scheme and review Lee–Hsu’s scheme.

### 2.1. Chebyshev polynomials

The degree of Chebyshev polynomial of degree $n$ is defined by $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$ [25]. For any integer $n \geq 0$, $T_0(x) = 1$, $T_1(x) = x$, if $x \in [-1, 1]$, in virtue of trigonometric identities, we can verify easily the semi-group property, which is eligible for designing public-key cryptosystems [26].

Zhang proved that the semi-group property holds for the Chebyshev polynomial defined on interval $(−\infty, +\infty)$, which can enhance the property [27] as follows:

$$T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x)) \mod p$$

where $n \geq 2$, $x \in (-\infty, +\infty)$, and $p$ is a large prime number.

Evidently, $T_n(T_{n}(x)) = T_n(x) = T_n(x) \mod p$. The security of it relies on the difficulties of the following problems.

**Definition 1.** Given two elements $x$ and $y$, the discrete logarithm problem is to find an integer $r$, such as $T_r(x) = y$.

**Definition 2.** Given three elements $x$, $T_1(x)$, and $T_2(x)$, the Diffie–Hellman problem is to find the element $T_{x}(x)$.

Up to now, there is no algorithm to be able to solve any of the above problems.

### 2.2. Lee and Hsu’s scheme

In this section, we briefly review Lee and Hsu’s scheme [7], which 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 sub-sections and in Figure 1. The notations throughout Lee–Hsu’s scheme are summarized in Table 1.

In the beginning, the registration center $R_i$ selects a random number $s$, a random integer $X_i$, and computes $SPUB = T_{X_i}(s) \mod p$. The registration center $R_i$ sends $X_i, s, p$ to the server via secure channel.

#### 2.2.1. Registration phase

When the remote user $C_i$ wants to login to the system, he or she has to register in registration center. He or she needs to perform the following steps:

1. The user $C_i$ sends identity $ID_i, f_i = h(B_i)$, $h(PW_i||B_i||N)$ to the registration center $R_i$ via a secure channel, where $B_i$ is $C_i$’s biometric template, $PW_i$ is $C_i$’s password, and $N$ is a random data.
2. $R_i$ computes $P_i = h(ID_i||X_i), r_i = h(PW_i||B_i||N) \oplus f_i, e_i = P_i \oplus r_i$, then embedded $(ID_i, h(\cdot), e_i, s, SPUB, p)$ in the user’s smart card and sends the card to $C_i$.
3. After receiving the smart card, $C_i$ computes $BPW = B_i \oplus h(PW_i)$ and inserts the random number $N$ and $BPW$ into smart card.

DOI: 10.1002/sec
2.2.2. Login phase

In this phase, when a legal user $C_i$ wants to access the server $S_i$, he or she needs to perform the following steps:

1. $C_i$ inserts his or her smart card into the card reader and offers his or her biometric template $B_i$ and password $PW_i$ on a specific device.

2. The smart card computes $B'_i = BPW_i \oplus h(PW_i)$ and verifies $B'_i = B_i$ if $B'_i \neq B_i$, the smart card rejects the request.

3. The smart card generates a random integer $RC$ and computes $f'_i = h(B'_i)$, $r'_i = h(PW_i \parallel B'_i \parallel N) \oplus f'_i$, $P'_i = e_i \oplus r'_i$, $M_1 = T_{SK} (s) \mod p$, $M_2 = T_{SK} (SPUB) \mod p$, $NID_i = ID_i \oplus h(M_1 \parallel M_2 \parallel t_i)$, $\alpha = h(ID_i \parallel NID_i \parallel P'_i \parallel M_1 \parallel M_2 \parallel t_i)$, $\beta = h(ID'_i \parallel P'_i \parallel M'_i \parallel M_2 \parallel t_i)$, $M'_i = T_{SK} (M'_i) = T_{SK} (s) \mod p$, $\gamma = h(ID'_i \parallel P'_i \parallel M'_i \parallel M_2 \parallel t_i)$, $\gamma' = h(ID'_i \parallel P'_i \parallel M'_i \parallel M_2 \parallel t_i)$, and verifies $\gamma' = \gamma$.

4. $C_i$ sends $\{NID_i, M_1, \alpha, t_i\}$ to $S_i$.

---

![Figure 1. Lee and Hsu's scheme.](image-url)
Table I. The notations used in Lee and Hsu’s scheme.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_i</td>
<td>Client</td>
</tr>
<tr>
<td>R_i</td>
<td>Trusted registration center</td>
</tr>
<tr>
<td>S_i</td>
<td>Server</td>
</tr>
<tr>
<td>pw_i</td>
<td>Password shared between the server and client</td>
</tr>
<tr>
<td>ID_i</td>
<td>Identity of the user C_i</td>
</tr>
<tr>
<td>B_i</td>
<td>Biometric template of the user C_i</td>
</tr>
<tr>
<td>P</td>
<td>A large prime number</td>
</tr>
<tr>
<td>X_i</td>
<td>A random integer chosen from the registration</td>
</tr>
<tr>
<td>S</td>
<td>A random number chosen by the registration</td>
</tr>
<tr>
<td>SPUB</td>
<td>The public key of ( R_i ) where ( SPUB = TX_i(S) ) modp</td>
</tr>
<tr>
<td>R_C, R_S</td>
<td>Two random integers</td>
</tr>
<tr>
<td>t_i</td>
<td>The timestamp</td>
</tr>
<tr>
<td>h_i</td>
<td>A secure one-way hash function</td>
</tr>
<tr>
<td>A</td>
<td></td>
</tr>
<tr>
<td>A \oplus B</td>
<td>The exclusive-or (XOR) operation between A and B</td>
</tr>
<tr>
<td>A \neq B</td>
<td>Check whether the value of A is equal to B</td>
</tr>
</tbody>
</table>

2.2.3. Authentication phase

After receiving the login request messages, the server \( S_i \) performs the following steps to access mutual authentication.

1. \( S_i \) checks whether the equation \( t' - t_i > \Delta t \) holds, where \( t' \) is the current timestamp of \( S_i \) and \( \Delta t \) denotes the expected valid time interval of transmission delay. If the equation holds, \( S_i \) rejects \( C_i \).

2. \( S_i \) computes \( M'_2 = TX_i(M_1) \) modp and \( ID_i = NID_i \oplus h(M_1||M'_2) \) and checks the validity of \( ID_i \).

3. \( S_i \) computes \( P_i = h(ID_i||X_i) \), \( a = h(ID_i||NID_i||P_i||M_1||M'_2||t_i) \).

4. Then, \( S_i \) verifies \( a' = a \). If \( a' \neq a \), \( S_i \) stops the session.

5. If so, \( S_i \) computes \( M_3 = TR_i(s) \) modp and \( \beta = h(ID_i||P_i||M_2||M'_2||t_2) \). Then, \( S_i \) sends \( \{M_3, \beta, t_2\} \) to \( C_i \).

6. After receiving \( \{M_3, \beta, t_2\} \), \( C_i \) first checks whether the equation \( t' - t_2 > \Delta t \) holds. If the equation holds, \( C_i \) rejects \( S_i \).

7. \( C_i \) computes \( \beta' = h(ID_i||P_i||M_2||M'_2||t_2) \) and verifies whether \( \beta' = \beta \). If they are not equal, \( C_i \) stops the session. Otherwise, \( C_i \) computes \( M_4 = TR_i(M_3) = TR_i(R_i(s)) \) modp and \( \gamma = h(ID_i||M_1||M'_2||t_3) \). \( C_i \) then sends \( \{\gamma, t_3\} \) to \( S_i \).

8. Upon receiving \( \{\gamma, t_3\} \), \( S_i \) first checks whether the equation \( t' - t_3 > \Delta t \) holds. If the equation holds, \( S_i \) rejects \( C_i \). Otherwise, \( S_i \) computes \( M_4 = TR_i(M_4) = TR_i(R_i(s)) \) modp and \( \gamma' = h(ID_i||M_1||M'_2||t_3) \) and checks whether \( \gamma' = \gamma \).

9. If it holds, \( S_i \) accepts \( C_i \)'s login request, and the verification is successful. Then, both \( C_i \) and \( S_i \) can use the session key \( M_4 \) and \( M'_4 \) to communicate with each other by using a symmetric crypto-system.

2.2.4. Changing password phase

In order to change the password, the user \( C_i \) performs the following steps:

1. User \( C_i \) inserts the smart card and offers both the biometric template \( B_i \) and old password \( PW_i \).

2. The smart card computes \( B'_i = BPW_i \oplus h(PW_i) \) and verifies \( B'_i = B_i \). If \( B'_i \neq B_i \), the smart card rejects the request.

3. If \( C_i \) passes the biometric verification, \( C_i \) enters his or her new password \( PW^\text{new}_i \).

4. The smart card computes \( f_i = h(B_i) \). If \( h(PW_i||[B_i]||N) \oplus f_i \equiv h(PW^\text{new}_i||[B_i]||N) \oplus f_i \), \( P_i = e_i \oplus r_i \), \( e_i = P_i \oplus r'_i \).

5. Finally, \( C_i \) replaces the \( e_i \) with \( e'_i \) in the smart card.

3. WEAKNESSES OF LEE AND HSU’S SCHEME

To analyze the security of Lee and Hsu’s scheme in this section, we assume that the attacker \( A \) can control the insecure channel and steal the message from smart card completely. Besides, we can see that Lee and Hsu’s scheme is vulnerable to user spoofing attack, password and biometrics guessing attack, and lack of three-factor authentication. In addition, the server’s secret key is also at risk. Finally, we will discuss the hash function problem about Lee and Hsu’s scheme in biometric verification. We now describe the following details.

3.1. User spoofing attack

In Lee and Hsu’s scheme, we see that stealing identity \( ID_i \) of another legal user \( C_i \) will cause a bad result. If attacker \( A \) has \( C_i \)'s \( ID_i \), attacker \( A \) can re-register to \( S_i \) by sending \( \{ID_i, f_i = h(B_i), h(PW_i||[B_i]||N)\} \). Then, \( R_i \) will send \( \{ID_i, h(\cdot), e_i, s, SPUB, p\} \) back to \( A \). After obtaining \( e_i, A \) can obtain \( P_i = h(ID_i||X_i) \) in hand and impersonate \( C_i \) owning \( ID_i \). Then, \( A \) will compute \( M_1 = TR_i(s) \) modp, \( M_2 = TR_i(SPUB) \) modp, \( NID_i = ID_i \oplus h(M_1||M_2) \), \( a = h(ID_i||NID_i||P_i||M_1||M'_2||t_1) \), where \( R_i \) is a random value chosen by \( A \). So \( A \) will send \( \{NID_i, M_1, a, t_1\} \) to \( S_i \) to masquerade \( C_i \).

3.2. Password and biometrics guessing attack

This attack is an extension of “user spoofing attack.” Attacker \( A \) can obtain the value \( P_i = h(ID_i||X_i) \) by re-registering to \( R_i \) with identity \( ID_i \) of \( C_i \). We consider the situation when this attacker \( A \) obtains the smart card corresponding to the identity \( ID_i \), and \( A \) somehow extracts the information \( (ID_i, h(\cdot), e_i, s, SPUB, p, PW_i, N) \) stored on smart card. Then, \( A \) can guess password \( PW_i \) and obtain the biometric \( B_i \) of \( C_i \) explained as follows:

Step 1: Computes \( h(ID_i||X_i) \oplus r_i \), which needs to be \( r_i = h(PW_i||[B_i]||N) \oplus f_i \).

Step 2: Guesses a password \( PW^\text{gus}_i \), computes \( B'_i = BPW_i \oplus PW^\text{gus}_i \), and obtains \( f'_i = h(B'_i) \).
Step 3: Check if \( h(PW_i || |B_i| || N) \oplus f_i = r_i \). If so, then \( A \) now possesses \( PW_i \) as well as the biometric template \( B_i \) of \( C_i \).

Attacker \( A \) may use these values of \( C_i \) to access other servers on behalf of \( C_i \) as it is convenient to keep same password for different servers. Moreover, because the biometric template is compromised, \( C_i \) cannot revoke and reissue his or her biometric template. Besides, one biometric template of the same user may be stored and shared in various databases with more and more biometric systems, which imperils biometric security and users’ privacy.

3.3. Scheme cannot achieve real three-factor

In Lee and Hsu’s scheme, a user \( C_i \) must provide right smart card, password, and biometric template for authentication. An attacker can obtain a user’s password by brute-force attack, dictionary attack, or Trojan horse. If attacker \( A \) could obtain the user’s smart card and password \( PW_i \), \( A \) can compute \( B_i = BPW \oplus h(PW) \). \( P_i = e_i \oplus h(PW || |B_i| || N) \oplus h(B_i) \) to pass mutual authentication without the user’s biometric template. Moreover, if an attacker \( B \) obtains the user’s smart card and biometric template, \( B \) can compute \( h(PW) = BPW \oplus B_i \). He or she can compute the authentication information \( P_i = e_i \oplus h(PW || |B_i| || N) \oplus h(B_i) \) to pass mutual authentication without the user’s password. In other words, an attack with two authentication factors can pass the authentication also.

3.4. Flaws in verification of biometrics using hash function

Hash function has been used in many existing biometric-based remote user authentication schemes to ensure security. One of the fundamental properties of the hash function is that the outputs are very sensitive to small perturbations in their inputs. Generally speaking, those cryptographic hash functions cannot be applied straightforwardly when the input data with noisy such as biometrics. In Lee and Hsu’s scheme, the authentication phase is partly dependant upon the verification of \( P_i = h(D_i || X_i) \). \( P_i = e_i \oplus h(PW || |B_i| || N) \oplus h(B_i) \). However, there may be a few differences between the input biometrics each time. This case will cause the legal user unable to pass the authentication phase.

4. THE PROPOSED SCHEME

In this section, we briefly introduce the Fuzzy extractors and secure sketch used in our proposed scheme and the details of our proposed scheme.

4.1. Secure sketch scheme [28]

In a secure sketch scheme as is shown in Figure 2, there are two components. The first is the sketch generation algorithm. In this algorithm, we take the original biometric template \( w \) as the input and a sketch \( S \) as output. The second algorithm is the biometric template reconstruction algorithm. In the second algorithm, we can take another biometric template \( w' \) and the sketch \( S \) as input then \( S^* \) can be obtained. If \( w \) and \( w' \) are sufficiently similar such as both of them come from the same person, we will have \( S = S^* \). From the theory of secure sketch, we can obtain the enrollment biometric even in the login phase. There are two components in the secure sketch scheme. From the literature [22], we give a simple example to achieve \( SS \) and \( Rec \).

Step 1: Computing \( SS(w; x) = w \oplus C(x) \), \( w \) denotes the binary biometric information of user; \( C \) denotes the error-correcting coding such as BCH coding; \( x \) denotes random binary string.

Step 2: Computing \( Rec(w', v) = v \oplus D(w' \oplus v) \). \( D \) denotes the decoding procedure of the code \( C \).

Thus, we can recover \( w \) by computing \( Rec(w', v) = v \oplus D(w' \oplus v) \).

4.2. Fuzzy extractors scheme [29]

As is shown in Figure 3, Fuzzy extractors can convert noisy nonuniform biometrics into reliably reproducible, uniformly random strings. At first, we can extract a uniformly random string \( R \) from its input \( w \) in a noise-tolerant way. Noise tolerance means that if the input changes to
some $w'$ but remains close to $w$, we can obtain the string $R$ also. To assist in reproducing $R$ from $w'$, the Fuzzy extractors outputs a non-secret string $P$. It is important to note $R$ remains uniformly random even the attacker obtains the string $P$. From the theory of Fuzzy extractors, random string $R$ can be used as a key. We can construct Fuzzy extractors from strong extractors $Ext$ and secure sketch [30].

From Figure 4, we can know that there are two steps to construct Fuzzy extractors.

Step 1: $Gen(w; r, x)$: set $P = (SS(w; r, x), R = Ext(w; x)$, and output $R$ and help data $P$.

Step 2: $Rep(w', (s, x))$: recover $w = Rec(w', s)$ and output $R = Ext(w; x)$.

The strong extractors $Ext$ can be generated by universal hash functions.

4.3. The proposed scheme

In this section, we shall present our remote user authentication scheme based on Fuzzy extractors and secure sketch. The notations in Table II are used in the proposed scheme. There are four phases in our scheme: (i) initialization, (ii) registration, (iii) login and authentication, and (iv) password change phase.

Sometimes, somebody may possess legal smart card but have no right to visit the servers because of resigning or other reasons. At the same time, different users have different access privileges with regard to the data. We can use access control table to solve the above problems. The following is an example of access control in Table III:

This is just a simple example of an access control table. From the earlier table, we can control the user more exquisitely. Even if somebody has a right smart card, the server can also deny the requirement.

4.4. Initialization phase

The registration center generates and keeps the master key $skey$ secretly. The length of $skey$ is at least 1024 bits.

4.5. Registration phase of proposed scheme

When the remote user wants to register and become a new legal user in the system, as shown in Figure 5, he or she needs to perform the following steps:

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$skey$</td>
<td>The main key of the server. The key is absolute security in our scheme.</td>
</tr>
<tr>
<td>$skey_i$</td>
<td>The subordination key of the server for each client $C_i$.</td>
</tr>
<tr>
<td>data$_{reg}$</td>
<td>The data were generated in the registration phase.</td>
</tr>
<tr>
<td>data$_{login}$</td>
<td>The data were generated in the login or authentication phase.</td>
</tr>
<tr>
<td>$pw_i$</td>
<td>User inputs his or her password for authentication.</td>
</tr>
<tr>
<td>$PW_i$</td>
<td>The template of password $pw_i$.</td>
</tr>
<tr>
<td>$B_i$</td>
<td>The original biometric of user. Generally speaking, original biometric must binary in our scheme.</td>
</tr>
<tr>
<td>$BB_i$</td>
<td>Cancelable biometric template for authentication.</td>
</tr>
<tr>
<td>$ID_i$</td>
<td>The identity of user.</td>
</tr>
<tr>
<td>$SID$</td>
<td>The identity of the server.</td>
</tr>
<tr>
<td>$H(.)$</td>
<td>Hash function.</td>
</tr>
<tr>
<td>$P$</td>
<td>According the help data, we can obtain key $k$ by Fuzzy extractors.</td>
</tr>
<tr>
<td>$k$</td>
<td>We can acquire uniformly random string $k$ from biometric by Fuzzy extractors.</td>
</tr>
<tr>
<td>$R_u$</td>
<td>A random string is generated by the client.</td>
</tr>
<tr>
<td>$R_s$</td>
<td>A random string is generated by the server.</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>The biometric matching algorithm.</td>
</tr>
<tr>
<td>$A \oplus B$</td>
<td>The exclusive-or (XOR) operation.</td>
</tr>
<tr>
<td>$A | B$</td>
<td>The concatenation operation.</td>
</tr>
<tr>
<td>$N$</td>
<td>A random number is generated by the client in order to protect the user’s password.</td>
</tr>
<tr>
<td>$A \neq B$</td>
<td>Check whether the value of $A$ is equal to $B$.</td>
</tr>
</tbody>
</table>
Remote three-factor authentication scheme based on Fuzzy extractors

<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Identity</th>
<th>Authority</th>
</tr>
</thead>
<tbody>
<tr>
<td>00001</td>
<td>Active/invalid</td>
<td>Manager</td>
<td>Read/write/modify/copy/and so on</td>
</tr>
<tr>
<td>00002</td>
<td>Active/invalid</td>
<td>Administrator</td>
<td>All rights</td>
</tr>
<tr>
<td>00003</td>
<td>Active/invalid</td>
<td>Common user</td>
<td>Read</td>
</tr>
</tbody>
</table>

4.6. Login and authentication phase of proposed scheme

In this phase, when the remote user wants to access the server, he or she needs to perform login and authentication as the following steps in Figure 6.

1. At first, the user inserts his or her smart and input password \( p_{\text{login}} \), captures his or her biometric from sensor to obtain \( B_i^\text{reg} \).

2. The client creates a cancelable biometric template \( BB_i^\text{reg} = k^\text{reg} \oplus B_i \).

3. The client generates uniqueness ID \( i \) for user and stores ID \( i \) in database. The server computes \( M_1 = ID_i \oplus H(\text{skey}) \), \( skey_i = H(\text{skey}||ID_i) \), \( M_2 = BB_i^\text{reg} \oplus \text{skey}_i \), \( U_1 = M_1 || M_2 \) and writes the server identity SID, \( H(.), U_1 \) into smart card. At last, the server sends smart card to the client by secure channel.

4. After receiving card from the server, the user inputs his or her password \( pw_i \) and generates password template \( PW_i^\text{reg} \) by computing \( PW_i^\text{reg} = pw_i @ h(k^\text{reg}||N) \). Then, the client writes password template \( PW_i^\text{reg} \) help string \( P \), secure sketch \( S \), random number \( N \) into smart card to finish the registration.

5. The server generates ID \( i \) for user and stores ID \( i \) in database. The server computes \( M_1 = ID_i \oplus H(\text{skey}) \), \( skey_i = H(\text{skey}||ID_i) \), \( M_2 = BB_i^\text{reg} \oplus \text{skey}_i \), \( U_1 = M_1 || M_2 \) and writes the server identity SID, \( H(.), U_1 \) into smart card. At last, the server sends smart card to the client by secure channel.

6. After the server receives \( C_0 \) the server computes \( H(\text{skey}) @ M_1 \) to obtain \( ID_i \) and searches in database in order to check whether this user has registered or valid according the information of Table III. If the result holds, we can perform the next step.

7. The client obtains \( k^\text{login} \) according \( B_i^\text{login} \) and \( P \) based on the theory of Fuzzy extractors. If the user is the same person between registration and login, \( k^\text{login} \) is equivalent to \( k^\text{reg} \). Then, the cancelable template \( BB_i^\text{login} \) can be computed \( BB_i^\text{login} = k^\text{login} \oplus B_i^\text{login} \).

8. In order to recover \( \text{skey}_i \) from \( M_2 = BB_i^\text{reg} \oplus \text{skey}_i \), the value of \( BB_i^\text{reg} \) must be recovered at first. This problem can be resolved by secure sketch theory. So we can obtain \( B_i^\text{reg} \) from \( B_i^\text{login} \) and \( S \), then \( BB_i^\text{reg} \) can be obtained with the computation of \( BB_i^\text{reg} = k^\text{login} \oplus B_i^\text{login} \). We can obtain \( \text{skey}_i \) by computing \( BB_i^\text{reg} \oplus M_2 = BB_i^\text{reg} \oplus BB_i^\text{reg} \oplus \text{skey}_i = \text{skey}_i \).

9. The client obtains \( N \) from smart card, computes \( PW_i^\text{login} = pw_i @ h(k^\text{login}||N) \), and checks \( PW_i^\text{login} = PW_i^\text{reg} \). If the result holds, the client generates random string \( R_a \) to compute \( M_3 = BB_i^\text{login} @ R_a \) and \( M_4 = \text{skey}_i @ R_a \). At last, the client sends \( C_0 = M_3 || M_4 || H(R_a || R_a) || R_a \) to the server.

10. After the server receives \( C_0 \) the server computes \( H(\text{skey}||ID_i) \) to obtain \( ID_i \) and searches in database in order to check whether this user has registered or valid according the information of Table III. If the result holds, we can perform the next step.
interval of transmission delay. If the result holds, we can perform the next step.

(7) The server obtains $BB_{\text{login}}^i$ from $M_3$ then computes \( \nabla BB_{\text{login}}^i \); if the result holds, we can perform the next step;

(8) The server generates random string $R_s$, then computes and sends $C_1 = M_5 \| M_6 \| H(t_2) \| t_2$ to the client.

(9) At first, the client checks the validity of $t_2$ by computing $t' \leq t_2 < \Delta t$. If the result holds, the client can obtain $SID$ from smart card, then computes $M_5 = H(SID) \| R_s$.

(10) The client computes $C_2 = H(R_s \| SID \| t_2) \| t_2$ then sends $C_2$ to the server.

(11) After receiving $C_2$, the server can check the validity of $t_2$ by computing $t' \leq t_2 < \Delta t$. If the result holds, the server then computes $C_2' = H(R_s \| SID \| t_2) \| t_2$. Then, the server checks $C_2' = C_2$. If they are equal, the client could pass authentication.

(12) The server provides corresponding resource for the user according to the information of Table III.

(13) After authentication, we can use $\text{key} = H(M_4 \| M_6) = H(R_s \| R_s)$ as session key.

4.7. Password change phase of proposed scheme

(1) The user inputs old password $pw_{\text{old}}^i$ and captures his or her biometric by sensor to obtain original data $Bpwd^i$

(2) The client obtains $P$ from smart card. According to the idea of Fuzzy extractors, the value of $kpwd$ could be obtained according to $Bpwd^i$ and $P$. If the user is the same person between change password phase and registration phase, $kpwd$ is equivalent to $k_{\text{reg}}$.

(3) The client can obtain random data $N$ and password template $PW_{\text{reg}}^i$ from smart card. Then, the client computes $PW_{\text{pwd}}^i = pw_{\text{pwd}}^i \oplus H(kpwd \| N)$ and checks $PW_{\text{pwd}}^i \oplus H(kpwd \| N)$. If they are equal, the user can input new password $pw_{\text{new}}^i$ and compute $PW_{\text{pwdnew}}^i = pw_{\text{pwdnew}}^i \oplus H(kpwd \| N)$. At last, we can replace $PW_{\text{reg}}^i$ by $PW_{\text{pwdnew}}^i$ to finish the password change phase.

5. ANALYSIS OF THE PROPOSED SCHEME

In this section, we will analyze the security and performance of our proposed scheme and show it could overcome the security weaknesses of Lee and Hsu’s.
5.1. Security analysis

Here, we describe several security analyses about our proposed scheme. The first four analyses show how the proposed scheme overcomes the drawbacks of Lee–Hsu’s.

User spoofing attack: In our scheme, ID is unanimously generated by the server. At the same time, ID is protected by the server’s master key. So the attacker cannot change or fabricate other user’s ID. From the analysis, we can see that the proposed scheme can resist user spoofing attack.

Password and biometrics guessing attack: In Lee–Hsu’s scheme, the password and biometric may be revealed. In our scheme, the password template is stored in the form of $PW_i = pw_i \oplus H(k|l|N_i)$. Even if an attacker acquires the password $pw_i$, he or she cannot obtain $PW_i$ also because the value of $k$ cannot be acquired without right biometric. At the same time, the original biometric information $B_i$ is protected in the form of $BB_i = k \oplus B_i$, so the attacker cannot obtain $B_i$ also. We can draw the conclusion that our scheme can resist password and biometric guessing attack.

Scheme cannot achieve real three-factor: In Lee–Hsu’s scheme, if an attacker obtains the user’s password, he or she can obtain biometric information also. So Lee–Hsu’s three-factor scheme may become two-factor. In our scheme, even if an attacker obtains a password, he or she cannot obtain the password template because he or she cannot provide proper biometric information to obtain $k$. Even if an attacker obtains the password template $PW_i$ and password $pw_i$, he or she cannot obtain $k$ from the value of $H(k|l|N_i)$. So our scheme can really achieve three-factor authentication.

Flaws in verification of biometrics using a hash function: In Lee–Hsu’s scheme, the biometric information has been input hash function directly. But one of the fundamental properties of the hash function is that the output is very sensitive to small perturbations in their input. So the biometric information cannot be the input of the hash function. However, our scheme can avoid this drawback by secure sketch and Fuzzy extractors.

Privilege inside attack: Privilege attack means that the server can obtain original biometric or password information from the client’s data. In our scheme, the password need not be sent to the server, so the server cannot obtain the user’s password. At the same time, the server cannot compute original biometric information $B_i$ from $BB_i^{reg}$ because $BB_i^{reg} = k^{reg} \oplus B_i$, and $k^{reg}$ cannot be acquired without the client’s biometric information. So our scheme can resist privilege inside attack.

User anonymity: An attacker may eavesdrop on the communication between the user and server and try to track the user’s real identity to find some information about the user. In our scheme, the real identity $ID_i$ is generated by the server and is protected by $M_1 = ID_i \oplus H(sk)$ (sk). An attacker cannot obtain $ID_i$ because an attacker cannot obtain $sk$. Therefore, our scheme can provide user anonymity.

Lost smart card attack: Assume that an attacker can extract all the information from the smart card by the side channel attack [31]. An attacker may try to derive the password from the information, but the password is protected by $k$ and $N$ in the form of $PW_i = pw_i \oplus H(k|l|N_i)$. An attacker cannot obtain $k$ without correct biometric information. So our scheme is secure against the smart card loss problem.

Mutual authentication: Our scheme can achieve mutual authentication between the user and server. During the authentication phase of our scheme, the server has to verify the validity $ID_i$ and $\mathbb{V}(BB_i^{reg}, BB_i^{login})$. The value of $BB_i^{reg}$ is protected by the server’s master key $skey$, so nobody can modify $BB_i^{reg}$ after the registration phase. The user sends a random string $R_u$ in the form of $skey \oplus R_u$. Only the server can generate $skey$ with his or her master key. Hence, both the user and the server can authenticate with each other, and mutual authentication between them is achieved.

Offline password guessing attack: In our scheme, the password is protected in the form of password template by computing $PW_i = pw_i \oplus H(k|l|N_i)$. The attacker cannot obtain $k$ without accurate biometric. So our scheme can resist offline password guessing attack.

Access control: Each authentication scheme seldomly considers the problem that the users may be canceled authorization. For example, someone has registered toward the server. However, he or she cannot pass the authentication any more for some reasons. In our scheme, we introduce an access control table (e.g., Table III) to solve this problem because we can check and manage the status of the user before authentication.

Biometric security: The client computes a cancelable biometric template as $BB_i = k \oplus B_i$. So the security of $B_i$ depends on parameter $k$. But an attacker cannot obtain $k$ without correct biometric. From the analysis, we can see that our scheme can provide biometric securely.

5.2. Performance analysis

Here, we discuss the performance of our proposed scheme as follows:

- $T_X$: time of performing an XOR operation;
- $T_W$: time of performing a one-way hash function;
- $T_C$: time of performing a symmetric encryption operation;
- $T_D$: time for performing a symmetric decryption operation;
- $T_{PC}$: time for performing a Chebyshev chaotic map operation;
- $T_{PE}$: time for performing an asymmetric encryption operation;
- $T_{PD}$: time for performing an asymmetric decryption operation;
- $T_{SS}$: time for obtaining sketch $S$ from the client’s registration biometric information $w$;
- $T_{Rec}$: time for recovering $w$ from secure sketch $s$ and the user’s login biometric information $w’$;

DOI: 10.1002/sec
TGen: time for obtaining help data \( P \) and secret key \( R \) from \( w \); and

\* TRep: time for recovering \( R \) from \( w' \) and help string \( P \); 

Here, we will analyze the performance of our proposed scheme at client.

**Step 1:** Because the value of \( T_X \) is very insignificant, we can neglect it.

**Step 2:** The value of \( T_H \) and \( T_E \) is the same [32], so we believe the value of \( T_P \) is approximate equal to \( T_E \) at the same platform.

**Step 3:** In Section 4.1, error correction of coding and decoding consume the main time of secure sketch. At the same time, Fuzzy extractor is constructed by secure sketch and universal hash function. So we can obtain

\[
T_{SS} + T_{Rec} + T_{Gen} + T_{Rep} = 2T_{Coding} + 2T_{Decoding} + 2T_H
\]

where \( T_{Coding} \) denotes the time of coding, \( T_{Decoding} \) denotes the time of decoding.

**Step 4:** On an Intel Core i5-3470 platform, we measured the consumption time BCH(172,71) and MD5, \( T_{Coding} = 0.78 \) s, \( T_{Decoding} = 0.18 \) s, \( T_H = 0.00097 \) s (s denotes second). The total value is that

\[
T_{SS} + T_{Rec} + T_{Gen} + T_{Rep} = 2T_{Coding} + 2T_{Decoding} + 2T_H = 2.54097 \text{ s}
\]

**Step 5:** The total executing time of \( 10T_H + 3T_C \) is 6.012 s.

From the analysis, we can draw the conclusion that our scheme is more time consuming at the client and slight less time consuming at the server than Lee and Hsu’s. But our scheme not only can overcome some problems of Lee and Hsu’s scheme but also can deal with the biometric more appropriately. So the proposed scheme is more practical and more reasonable.

We also compare the performance of our scheme with other schemes in Table IV. From Table V, we can see that the proposed scheme is more secure. The cost of the proposed scheme is higher at the client and lower at the server than Lee and Hsu’s scheme. However, Lee and Hsu’s scheme face many problems such as user spoofing attack, password and biometric guessing attack, scheme lacks three-factor authentication, and biometric information is dealt inappropriately. What is more, our scheme can overcome the weaknesses of Lee and Hsu’s scheme. The proposed scheme is more secure and more reasonable than Lee and Hsu’s scheme also.

**6. CONCLUSION**

In this paper, we have presented a cryptanalysis of Lee and Hsu’s scheme and find some problems in it. In order to solve these problems, secure biometric-based remote authentication with Fuzzy extractors and secure sketch has been proposed. From the analysis, we can see that the proposed scheme has higher security and deals with biometric more appropriately in spite of higher computation cost at the client than Lee–Hsu’s scheme. At the same time, the proposed scheme is more secure and more reasonable than Lee and Hsu’s scheme also.
time, the access control method has been suggested to manage authentication more carefully. At last, our scheme can achieve session key agreement also.

ACKNOWLEDGEMENTS

This work was supported by Sichuan Basic Science & Technology Foundation under Grant 2013JY0036, and by Southwest University for Nationalities Foundation under Grant 2014NZYQN47.

REFERENCES


