Enhanced Biometrics-based Remote User Authentication Scheme Using Smart Cards

Jian-Zhu Lu*, Shaoyuan Zhang, and Shijie Qie

Department of Computer Science, Jinan University,
Guangzhou, Guangdong, China 510632
tljz@jnu.edu.cn,Zhangsy@yahoo.com.cn,Qiesj@yahoo.com.cn

Abstract. In 2010, Li and Hwang proposed an efficient biometrics-based remote user authentication scheme using smart card. Recently, for improving its security and supporting session key agreement, Li et al. proposed an improvement. In this article, we show that two schemes are unsafe for a user C_i to reveal an obsolete value of R_C to an attacker, who can succeed in either impersonating the user or obtaining her/his current session key. In addition, these schemes suffer from replay attacks and DoS attacks, and their biometrics authentication cannot be used safely once the template f_i is leaked. We remedy this situation by designing an enhanced version of biometrics-based remote user authentication scheme. We discuss its functionality, security and efficiency. We also provide a comparison of the related schemes in the same category. Compared to Li and Hwang's and Li et al.'s, not only does the proposed scheme enhance the security, but furthermore, our design is more efficient than theirs.

Keywords: Biometrics, user authentication, smart cards, security

1 Introduction

The biometrics authentication system offers several advantages over other security methods. Passwords might be divulged or forgotten, and smart cards might be shared, lost, or stolen. In contrast, personal biometrics, such as fingerprints or iris scans, have no such drawbacks. It is ideally suited for both high security and remote authentication applications due to the nonreturnable nature and user convenience [13].

Remote authentication is a form of e-authentication in which user credentials, as proof of identities, are submitted over a network connection. Remote authentication poses unique security challenges given its open, uncontrolled and unsupervised nature. There are two problems in applying personal biometrics

* This work was supported in part by the National Natural Science Foundation of China under Grants 60773083, by the Provincial Natural Science Foundation of Guangdong under Grants 2008B090500201, 2009B010800023 and 2010B090400164, and by the Projects in the Scientific Innovation of Jinan University under Grants 11611510.

to remote authentication. One of the most important is obtaining easily some biometric characteristics, so that the results can never be changed. Another is the difficulty of checking whether the device is capable of verifying that a person is alive since the biometric capture devices are remotely located [12]. Because of such problems, the best approach is to integrate biometrics with passwords and smart cards to construct a secure three-factor authentication scheme. Several three-factor authentication schemes have been proposed in the literature [3, 6, 7, 12, 11, 4].

In 2010, based on the one-way hash function, biometrics verification and smart card, [9] proposed an efficient biometric-based remote user authentication scheme, in which the computation cost is relatively low compared with other related schemes. Recently, [10] showed that Li and Hwang's scheme neither provides proper authentication nor resists the man-in-the-middle attacks. They then presented an improved scheme to fix the problem.

In above schemes [9, 10], the user chose a random number R_C, and computed $M_2 = h(ID_i||X_S) \oplus R_C$ for the output of user login phase. In this article, we show that $h(ID_i||X_s)$ can easily be obtained by an attacker obtaining an obsolete value of R_C. Then, without user's password and personal biometrics, the attacker can succeed in either impersonating the user or obtaining the session key. In these schemes, once the template f_i is leaked, the biometrics authentication is facing a dilemma of how to identify a forgery. In addition, they suffer from replay attacks and DoS attacks. We remedy this situation by suggesting an enhanced scheme. We also demonstrate how the enhanced scheme is efficient. Furthermore, the security of the enhanced scheme will be demonstrated by formal proofs.

The structure of this paper is organized as follows. In Section 2, we review Li and Hwang's and Li et al's schemes, and point out the weaknesses of these schemes. In Section 3, we propose an enhanced biometrics-based remote user authentication scheme. In Section 4 and 5, security and performance analysis are given, respectively. Finally, we conclude this paper in Section 6.

2 Security analysis for Li-Hwang's scheme and its improvement

2.1 Review for Li-Hwang's scheme

Li-Hwang's scheme [9] is composed of four phases namely; the registration phase, the login phase, the authentication phase and password change phase. In their scheme, there are three participants, the registration center (R), the server (S_i) and the user (C_i), where R is assumed to be a trusted party. R chooses the master secret key X_S and distributes it to S_i via a secure channel.

Registration phase When client C_i wants to perform his registration, he requests registration center R with his personal biometrics B_i, password PW_i, and identity ID_i. After receiving the request, R computes $r_i = h(PW_i||f_i)$ and

e_i=$h(ID_i||X_S) \bigoplus r_i$, where $f_i = h(B_i)$, and X_S is the secret information generated by S_i. After personalizing the smart card with parameters $(ID_i, h(\cdot), f_i, e_i)$, R returns the smart card to C_i.

Login phase When C_i wants to login to the remote server S_i, he inserts his smart card in the terminal and inputs his personal biometrics B_i. If $h(B_i) = f_i$, the smart card requires C_i to key the PW_i. Then, it outputs the message M_2, where M_2=$M_1 \bigoplus R_C$, M_1=$e_i \bigoplus r'_i$, r'_i=$h(PW_i||f_i)$, and R_C is a random number generated by the user. Finally, C_i sends the message (ID_i, M_2) to S_i.

Authentication phase Referring to Fig. 1, its authentication process is described below. After receiving the login request, S_i checks the format of ID_i, and then sends $M_5 = h(ID_i||X_S) \oplus R_S$ and $M_6 = h(M_2||(M_2 \oplus h(ID_i||X_S)))$ back to C_i, where R_S is a random number chosen by S_i. C_i verifies the legality of S_i according to the relation M_6=$h(M_2||R_C)$, and sends back $M_8 = h(M_5||(M_5 \oplus M_1))$ to S_i. If $M_8 = h(M_5||R_s)$, C_i and S_i authenticate each other successfully.

Password changing phase This phase is invoked whenever C_i wants to change her/his password PW_i to a new password PW_i^n. First, C_i inserts smart card into the terminal device, and inputs personal biometrics B_i. After passing the biometrics verification (i.e., $h(B_i)$=f_i), C_i is required to enter the old password PW_i and a new one PW_i^n. By first computing $e'_i = e_i \bigoplus h(PW_i||f_i)$ and then setting $e''_i = e'_i \oplus h(PW_i^n)$, the smart card replaces e_i with e''_i.

2.2 The improvement of Li-Hwang's scheme

Li-Hwang's scheme is very efficient in terms of communication and storage space, but it suffers from the impersonation attacks and the man-in-the-middle attack [10]. Based on this weakness, an improvement is discussed in [10]. A secret random number y and a master key X_S are distributed to S_i by R via a secure channel. The improvement is also composed of four phases, and the password change phase is the same as that of Li-Hwang's scheme. We next start with a brief review of the improvement.

Registration phase By first generating a random number N and then setting RPW_i=$h(N||PW_i)$, client C_i sends his registration information (B_i, RPW_i, ID_i) to R. After receiving the request, R personalizes a smart card with parameters $(f_i, e_i, h(\cdot), y)$, and returns the smart card to C_i, where $f_i = h(B_i)$, e_i=$h(ID_i||X_S) \bigoplus r_i$, and $r_i = h(RPW_i||f_i)$.

Login phase In the login phase, the system authenticates C_i's personal biometrics B_i by matching the biometric template f_i, and generates a request (ID_i, M_2, M_4, M_5) to S_i. Here, M_2=$e_i \oplus h(RPW_i||f_i) \oplus R_C$, M_4=$RPW_i \oplus h(y||R_C)$, M_5=$h(M_2||h(y||R_C)||M_4)$, RPW_i=$h(N||PW_i)$, and R_C is a random number chosen by C_i.

Authentication phase Referring to Fig. 2, S_i authenticates C_i by first computing M_8=$h(y||h(M_2 \oplus h(ID_i||X_S))$ and then checking if M_5=$h(M_2||M_8||M_4)$. If C_i is trustworthy, S_i sends the response (M_{10}, M_{11}) to C_i, where M_{10}=$h(M_9||SID_i||y) \oplus M_8 \oplus R_S$, M_{11}=$h(h(ID_i||X_S)||M_9||y||R_S)$, and M_9=$M_4 \oplus M_8$. By computing M_{12}=$h(RPW_i||SID_i||y) \bigoplus M_3 \bigoplus M_{10}$ and verifying if M_{11}=$h(M_1||RPW_i||y||M_{12})$, C_i can authenticate S_i. C_i and S_i finally establish a session key SK when they authenticate each other successfully.

Password changing phase Whenever C_i wants to replace her/his password PW_i with a new password PW_i^{new}, this phase is performed. , After inserting smart card into the terminal device, C_i firstly inputs personal biometrics B_i. Then, C_i is required to enter the old password PW_i and a new one PW_i^n if she/he passes the biometrics verification. Finally, the smart card replaces e_i with e_i^{new}, where $e_i^{new} = e_i' \oplus h(RPW_i^{new})$, $e_i' = e_i \bigoplus h(RPW_i||f_i)$, RPW_i^{new}=$h(N||PW_i^{new})$, and RPW_i=$h(N||PW_i)$.

C_i		S_i
$(ID_i, h(\cdot), f_i, e_i)$ (B_i, PW_i)		(X_S)
	$\xrightarrow{\quad ID_i, M_2 \quad}$	
	$\xleftarrow{\quad M_5, M_6 \quad}$	
	$\xrightarrow{\quad M_8 \quad}$	

r'=$h(PW_i||f_i)$, M_2=$e_i \oplus r_i' \oplus R_C$
M_5=$h(ID_i||X_S) \oplus R_S$,
M_6=$h(M_2||(M_2 \oplus h(ID_i||X_S)))$
M_8=$h(M_5||(M_5 \oplus M_1))$

Fig. 1. The message flow of the authentication phase in [9]

2.3 Security analysis

Li et al.'s scheme are more secure than Li-Hwang's. Two schemes can be further improved by examining the following three cases:

(1) The leak of $\tilde{R}_C$ used in some obsolete request $\{ID_i, \tilde{M_2}\}$ makes the secret key ck visible to an attacker.

In Li-Hwang's scheme, M_2=$e_i \oplus r_i' \oplus R_C = h(ID_i||X_S) \oplus R_C$. It is true for any request message (R_C, M_2) from C_i. For the version employing randomly chosen number R_C, we do not see a way of getting it according to M_2 and ID_i. Note that R_C may be not ephemeral in S_j. In the nonce-based authentication schemes, the client's ephemeral random number is recovered and usually stored in S_j's database. The aim is to check the freshness of random number in the C_i request. For example, the scheme in [10] stores (ID_i, M_7) in the S_j's database, where

C_i		S_i
$(f_i, e_i, h(\cdot), y, N)$ (ID_i, B_i, PW_i)		(X_S, y)
	$\xrightarrow{ID_i, M_2, M_4, M_5}$	
	$\xleftarrow{M_{10}, M_{11}}$	

$RPW_i{=}h(N||PW_i)$, $M_2{=}e_i \oplus h(RPW_i||f_i) \oplus R_C$
$M_4{=}RPW_i \oplus h(y||R_C)$, $M_5{=}h(M_2||h(y||R_C)||M_4)$
$M_8{=}h(y||h(M_2 \oplus h(ID_i||X_S))$, $M_9{=}M_4 \oplus M_8$
$M_{10}{=}h(M_9||SID_i||y) \oplus M_8 \oplus R_S$
$M_{11}{=}h(h(ID_i||X_S)||M_9||y||R_S)$
$SK{=}h(RPW_i||h(y||R_C)||R_S||SID_i)$

Fig. 2. The message flow of the authentication phase in [10]

$M_7 = R_C$. This gives an attacker a chance to obtain a copy of R_C. Therefore, it is possible that an attacker, who makes the attack stealthy in terms of not getting noticed by server S_j, will obtain the random $\tilde{R}_C$ used in some obsolete request $\{ID_i, \tilde{M}_2\}$. If such a $\tilde{R}_C$ is acquired, the attacker can compute $ck = \tilde{R}_C \oplus \tilde{M}_2$. Once the ck is found, the attacker can clearly succeed in impersonating either party without C_i's password PW_i and personal biometrics B_i.

Notice first that an attacker in [10], can also get the common secret key ck as above. Next, in order to impersonate the parties, she/he need to be able to get hold of another key y. According to the operational approach at the registration phase, it is known to the attackers that $(f_i, e_i, h(\cdot), y)$ are stored in C_i's smart card by R. Since y is present in plaintext, a copy of y can be done from C_i's smart card in a background to avoid user attention, more stealthy techniques are also possible. Knowledge of the y and ck can help the attacker find the session key $SK = h(PRW_i||h(y||R_C)||R_S||SID_i)$. Concretely, from the request message (ID_i, M_2, M_3, M_4), the attacker can obtain $R_C = ck \oplus M_2$ and $RPW_i = M_4 \oplus h(y||R_C)$. In addition, $R_S = h(PRW_i||SID_i||y) \oplus h(y||R_C) \oplus M_{10}$, and it can be computed by the response (M_{10}, M_{11}) from S_i.

(2) Replay attacks

S_i in [9], only checks the format of the user's identity ID_i, and does not verify the validity of login message M_2. This could lead to some attacks against the server S_i, like denial-of-services (DoS), replay attacks, and man-in-the-middle attacks.

Let $(ID_i, M_2^{(l)}, M_4^{(l)}, M_5^{(l)})$ be C_i's login messages which passed the test of the authentication phase in [10], where $l = 1, 2, \cdots, \tau$. After receiving $(ID_i, M_2^{(\tau+1)}, M_4^{(\tau+1)}, M_5^{(\tau+1)})$ from C_i, S_i computes $M_7^{(\tau+1)}{=}\ h(ID_i||S_X) \oplus M_2$ and $M_8^{(\tau+1)}$. Then, he verifies if $M_5^{(\tau+1)}{=}h(M_2^{(\tau+1)}||M_8^{(\tau+1)}\ ||M_4^{(\tau+1)})$, at the same time, checks if $M_7^{(\tau+1)}$ is equal to $M_7^{(\tau)}$ in the database. If both are true, S_i deletes $(ID_i, M_7^{(\tau)})$ and stores $(ID_i, M_7^{(\tau+1)})$ to protect against a replay attack. We note that $M_7^{(\tau+1)}{=}R_C^{(\tau+1)}$, and $R_C^{(\tau+1)}$ is a one-time random number. One potential issue here is that an attacker may replay the outdated login messages, $(ID_i, M_2^{(l)}, M_4^{(l)}, M_5^{(l)})$, $l = 1, 2, \cdots, \tau$.

The server S_j cannot authenticate an outdated request immediately after receiving it. This means that S_j has to generate its response before properly authenticating it. As a result, an attacker can force S_j to process a large number of outdated requests to eventually exhaust its resource. There is no way to prevent such an attacker from launching DoS attack to S_j.

(3)Insecure protection for personal biometrics B_i.

In [9] and [10], the biometric information, B_i, is acquired at the time of initial registration. The feature termed a template f_i is extracted and stored in the smart card, where $f_i = h(B_i)$. B_i remains unchanged through C_i's life and cannot be changed easily in contrast to the password and the encrypted key. Accordingly, in case the template f_i is leaked, their schemes arises a problem that the biometrics authentication cannot distinguish between genuine and fake template f_i.

3 The proposed scheme

In this section, we propose a secure and efficient biometrics-based user authentication scheme for remote access. The registration center (R) is presented as a trusted third party which is invoked only in the registration phase. An authentication system can be described formally with the help of the message space $\mathcal{M}$, the master key spaces $\mathcal{X}$, the identity set ID, a family $\mathcal{H}$ of hash function from $\{0,1\}^*$ to $\{0,1\}^l$, and a related family $\mathcal{MAC}$ of message authentication code from $\{0,1\}^\kappa \times \{0,1\}^*$ to $\{0,1\}^l$.

We denote the enrolled biometric template as f_i, and the input biometric data after image processing at login phase as f_i^*. To measure the similarity, we definite a normalized distance between two strings as $\rho(f_i, f_i^*)$=$1 - \frac{d_H(f_i, f_i^*)}{N}$, where d_H is a Hamming distance comparison between two binary strings, and N is the length of binary string. A larger value of ρ=$\rho(f_i, f_i^*)$ means that the two strings are more similar. It is noted that ρ is between 0 and 1. The distance for perfect matching is one.

The biometrics-based remote user authentication scheme consists of five phases: 1) initialization; 2) user registration; 3) user login; 4) remote authentication, and 5) password and template update. Detailed steps of these phases of the proposed scheme are described as follows.

3.1 Initialization phase

The proposed initialization phase contains two steps: 1) system setup and 2) server enrollment. System setup is implemented once by the R to setup the overall enrollment system. Let ρ be a matching algorithm for user's biometrics. In this step, given the security parameter κ, the R determines a hash function $h(\cdot) \in \mathcal{H}$ and a message authentication code $MAC_{(\cdot)}(\cdot) \in \mathcal{MAC}$, and publicizes them. In the server enrollment step, a legal server S_j is provided a master secret key $X_S \in \mathcal{X}$ by R, where X_S is shared between R and the server S_j.

3.2 User registration phase

A user C_i with identifier ID_i should first carry out this phase once before she/he can use any of the services provided by the server S_j. Users may use their medium access control or network layer address as an identity when contacting R for the authorization for their demands. In this phase, C_i needs to perform the following steps.

Step (1): Firstly, user C_i inputs his/her personal biometrics, B_i, on the specific device, and provides the password, PW_i, identity of the user, ID_i, to R via a secure channel.

Step (2): Next, R reads its current timestamp T_R, and computes $f_i{=}h(B_i \oplus h(PW_i||T_R))$, $r_i{=}h(B_i||PW_i||f_i)$ and $e_i{=}h(ID_i||X_S) \oplus r_i$.

Step (3): Lastly, R stores $(ID_i, h(\cdot), \rho(\cdot), MAC_{(\cdot)(\cdot)}, f_i, e_i, T_R)$ on the C_i's smart card and sends it to C_i via a secure channel.

3.3 User login phase

Whenever C_i wants to login a server S_j with identifier SID_j, she/he must perform the following steps:

Step (1): After inserting her/his smart card into the card reader, C_i inputs the PW_i and personal biometrics, B_i, on the specific device. Then, the smart card computes $f_i^*{=}h(B_i \oplus h(PW_i||T_R))$.

Step (2): The smart card checks if the matching score $\rho(f_i, f_i^*)$ is not beyond a predefined threshold value. If true, C_i passes the biometric verification, and performs the following steps.

Step (3): C_i inputs ID_i. Then, the smart card computes the following messages:

$$
\begin{aligned}
&r_i = h(B_i||PW_i||f_i)\\
&ck = e_i \oplus r_i\\
&tk = h(ck||t_C)\\
&M_C = tk \oplus R_C\\
&A_C = MAC_{ck}(R_C||t_C||\mathrm{SID}_j)
\end{aligned}
$$

where t_C is the C_i's current timestamp, R_C is a random number generated by the user, and $||$ is a concatenation operation for two bit strings. Here, user C_i and server S_j need not have synchronized clocks, and t_C is treated as the nonce generated by C_i. The message authentication code A_C is introduced to authenticate the legitimacy of C_i.

Step (4): Finally, C_i sends the message (ID_i, t_C, M_C, A_C) to the remote server S_j, and stores (R_C, t_C).

3.4 Remote authentication phase

A user performs the remote authentication phase based on the login message for authentication as long as it visits the server. Without the clock synchronization

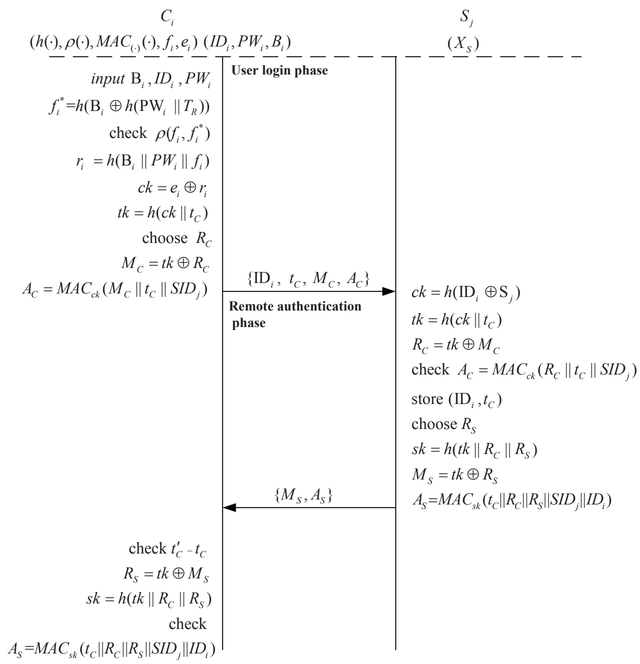

Fig. 3. The mutual authentication between C_i and S_j in the proposed scheme.

assumption, C_i and S_j perform the following steps to achieve mutual authentication and to establish a session key.

Step (1): After receiving the login message (ID_i, t_C, M_C, A_C), S_j checks whether the format of ID_i is valid or not. If true, S_j retrieves $R_C{=}M_C \oplus tk$ by computing $ck{=}h(ID_i||X_S)$ and setting $tk{=}h(ck||t_C)$, and then authenticates C_i by using the attached message authentication code A_C.

Step (2): If $A_C{\neq}MAC_{ck}(R_C||t_C||\mathrm{SID}_j)$, S_j rejects the login request and terminates the session; otherwise, S_j stores (ID_i, t_C) in the database. When receiving C_i's next request login message $(ID_i, \bar{t}_C, \bar{M}_C, \bar{A}_C)$, S_j compares $\bar{t}_C$ with the stored t_C. if $\bar{t}_C \leq t_C$, S_j reject it since it is a replay message. If $(ID_i, \bar{t}_C, \bar{M}_C, \bar{A}_C)$ is valid, S_j deletes t_C and stores $\bar{t}_C$. This mechanism can resist the replay attacks and man-in-the-middle attacks.

Step (3): S_j chooses a random number R_S, and then generates the session key $sk{=}h(tk||R_C||R_S)$ and message (M_S, A_S), where $tk{=}h(ck||t_C)$, $R_C{=}tk \oplus M_2$, $M_S{=}tk \oplus R_S$, and $A_S{=}$ $MAC_{sk}(t_C||R_C||R_S||\mathrm{ID}_j||ID_i)$.

Step (4): S_j sends the response message (M_S, A_S) to C_i.

Step (5): After receiving S_j's response message at time t'_C, C_i first checks if $t'_C - t_C$ is beyond a predefined delay. If true, C_i rejects the response message, and terminates the session.

Step (6): C_i restores $R_S{=}tk \oplus M_S$ according to tk in the user login phase. Then, C_i computes the session key $sk{=}h(tk||R_C||R_S)$, and checks if A_S =

$MAC_{sk}(t_C||R_C||R_S||\mathrm{ID}_j||ID_i)$. If they are equal, then C_i authenticates S_j and believes the share session key sk.

The message flow of the remote authentication phase is described in Fig. 3.

3.5 password and template update phase

C_i updates her/his password PW_i and template f_i in two steps. First, C_i inserts the smart card, and inputs his/her old password PW_i and personal biometrics, B_i, on the specific device. The biometrics verification is performed by checking the matching score $\rho(f_i, f_i^*)$, where $f_i^*{=}h(B_i \oplus h(PW_i||T_R))$. In the second step, C_i who passes the biometrics verification, inputs a new password PW_i^*. Then, the smart card computes $r_i{=}h(B_i||PW_i||f_i))$, $r_i^*{=}h(B_i||PW_i^*||f_i^*))$, and $e_i^*{=}e_i \oplus r_i \oplus r_i^*$. Finally, e_i^* and f_i^* are stored in the smart card while e_i and f_i are deleted.

4 Performance analysis

Performance is a key factor for popularizing the services in network communication systems. Especially, almost all of the remote users pay much attention to the performance issue due to the limited computation capabilities of their devices. Among the biometrics-based remote user authentication schemes proposed in the literatures [7–10, 4], [9] is one of efficient ones.

We adopt SHA-256, which has a 256-bit output, to implement the one-way hash function. We also implement the random-number generator and the message authentication code function by SHA-256 in the scheme. In general, the length of the identity of every remote user is usually less than 128 bits. Thus, we let the length of the user's identity be 128 bits. Besides, the length of every random number produced by the random-number generator is 256 bits and the length of every timestamp is about 60 bits. In the following, the comparisons of our scheme and other related schemes are summarized in Table 1. From Table 1, the proposed scheme is designed that guarantees not only resilient against man-in-the-middle attacks and DoS attacks at low communication costs, but also the secure protection for common secret key and personal biometrics with a few hashing function computations. This feature makes the proposed scheme practical.

The proposed scheme provides the following security guarantees.

Secure protection for ck and B_i: During the login phase, C_i first uses the timestamp t_C to generate a one-time temporary key tk with $tk = h(ck||t_C)$. Next, R_C is selected at random, and M_C is determined using $M_C = tk \oplus R_C$. An attacker has many more ways of obtaining tk. However, it is difficult for him to get ck from tk and t_C since $h(\cdot)$ is a one-way function.

During user registration phase, personal biometrics B_i is transformed into a template f_i with its current password PW_i. Here, $f_i{=}h(B_i \oplus h(PW_i||T_R))$. According to the above method, the server cannot know the original biometrics B_i even during authentication, and the privacy of individual can be protected. Further, even if the template f_i is leaked, the security of the scheme can be

Table 1. Comparison with other related schemes

	Li-Hwang (2010b)	Li et al (2011)	Ours
Computational cost in registration phase	$3h$	$4h$	$4h$
Computational cost in user login phase	$2h$	$4h$	$4h+\rho$
Computational cost in user authentication	$5h$	$6h$	$5h$
Communication cost in user login phase	384bits	896bits	696bits
Communication cost in user authentication	512	512	512
Change template freely	No	No	Yes
Common secret key protection	No	No	Yes
Resilient for replay attacks and DoS attacks	No	No	Yes
Resilient for man-in-the-middle attacks	No	Yes	Yes
Session key agreement	No	Yes	Yes

guaranteed by changing the password PW_i, preparing a template again and registering it.

Mutual authentication: Let $C_i \xleftrightarrow{sk} S_j$ denote that C_i and S_j share the common session key sk. To demonstrate that the proposed scheme satisfies mutual authentication, we need to argue that C_i believes that S_j believes $C_i \xleftrightarrow{sk} S_j$, and that S_j believes that C_i believes $C_i \xleftrightarrow{sk} S_j$ for the transaction [2, 5].

Consider Fig.3. C_i receives (M_S, A_S) as a response after sending (ID_i, t_C, M_C, A_C) to the server S_j. By recovering R_S from M_S and tk=$h((e_i \oplus r_i)||t_C)$, C_i can get the session key sk=$h(tk||R_C||R_S)$, and check whether $MAC_{sk}(t_C||R_C ||R_S||\mathrm{ID}_j||ID_i)$ matches the received value A_S. If true, C_i believe $C_i \xleftrightarrow{sk} S_j$. Since $r_i = h(B_i||PW_i||f_i)$ is computed by C_i in the user login phase, and the nonce t_C is picked by the user himself, C_i believes that tk is fresh and can only be recovered by S_j using the common secret key $h(ID_i||X_S)$. Thus, C_i believes that S_j believes that $C_i \xleftrightarrow{sk} S_j$ due to the fact that only S_j has the knowledge of X_S to compute tk and sk from ID_i and t_C. and validate that A_C matches $MAC_{ck}(R_C||t_C||\mathrm{SID}_j)$. If true, S_j will compute the session key sk =$h(h(ck||t_C)||R_C||R_S)$, and believe that $C_i \xleftrightarrow{sk} S_j$. Since S_j himself decides the random number R_S, he believes that R_S is fresh. On the receipt of M_C from C_i, S_j is sure via the aforementioned verification that R_C and t_C are correct, and then believes that C_i believes that $C_i \xleftrightarrow{sk} S_j$.

The man-in-the-middle attacks: In the man-in-the-middle attacks, an attacker can impersonate a S_j and fool the previous requester C_i to connect to the attacker, instead of to the S_j. The attacker can then capture the C_i's session key. In the proposed scheme, session security is provided through the use of one-time temporary key tk and message-authentication-code. In the case that the identity of each party in the scheme is authenticated, the scheme is secure against man-in-the-middle attacks.

In the proposed scheme, the authenticity of each login output is confirmed in time. S_j verifies the message-authentication-code A_C=$MAC_{ck}(R_C||t_C||\mathrm{SID}_j)$ to guarantees the authenticity for the login output received from a registered C_i, where ck=$h(ID_i||X_S)$, R_C=$M_C \oplus tk$ and tk=$h(ck||t_C)$. If the check of C_i's identity fails, then an attacker could redirect that login output at step (1), say to S'_j, before the S_j receives it, with the subsequent result that C_i would unknowingly communicate with S'_j instead of S_j. Following the tk and R_S at step (6) in the remote authentication phase, C_i checks $A_S{=}MAC_{sk}(t_C||R_C||R_S||\mathrm{ID}_j||ID_i)$ to verify that the message really is a reply by S_j to the current temporary key $tk = h(ck||t_C)$. If the check of S_j's identity fails, the message at step (4) are redirected to another server, say to S_j'', after the S_j sends it. As a result, C_i communicates with S_j'', rather than the intended S_j.

Replay attacks and DoS attacks: In DoS attacks, the attackers may flood a large number of illegal access request messages to the server S_j. Their aim is to consume its critical resources. By exhausting these critical resources, the attacker can prevent the server from serving legitimate users. In the proposed scheme, for every access request (ID_i, t_C, M_C, S_C) from all users that have registered in the R, S_j can check the validity of the login message in time, and it only needs to perform two hash operations. Furthermore, we make use of the timestamp t_C to prevent replay attacks. Thus, our solution does not suffer from this attacks.

Secure session key establishment: As we have previously analyzed, tk is a one-time secret between C_i and S_j. In the proposed scheme, the session key sk is computed as the hash value $sk = h(tk||R_C||R_S)$, where R_C and R_S are two random numbers. Thus, the session key $sk = h(tk||R_C||R_S)$ can be shared only by C_i and the S_j. C_i confirms the validity of sk by checking if $A_S{=}MAC_{sk}(t_C||R_C||R_S||\mathrm{ID}_j||ID_i)$, and the S_j confirms by sending back (M_S, A_S). The only way for an attacker to obtain the session key is through the offline guessing attack. The attacker reconstructs $MAC_{sk'}(t_C||R_C'||R_S'||\mathrm{ID}_j||ID_i)$ and compares it with the S_j's reply A_S. If a 2-bytes user identifier and a 160-bits session key are employed, it takes at least 2.29×2^{120} years for an attacker, who can compute one billion hash operations in one second, to break the session key [14].

Password guessing attacks: Two assumptions are made about a password-based authentication protocol. One, that all sensitive information in C_i's smart card can be successfully extracted by the attacker. The second assumption is that the public key cryptosystem technology cannot be utilized to eliminate the correlation of transmitted protocol messages in a normal session. Just as the analysis in [15][page, 2558], the password guessing attacks becomes an inherent limitation of password based authentication protocol under the above assumptions. The best solution way is to reduce the success probability of password guessing attacks.

Li-Hwang's scheme and its improvement suffer from this attacks. For client C_i in Li-Hwang's scheme [9], an attacker $\mathcal{A}$ eavesdrops C_i's login request message (ID_i, M_2) and the corresponding response (M_5, M_6) from S_i. Then, by extracting f_i and e_i in C_i's smart card, $\mathcal{A}$ computes $r_i^*{=}h(PW_i^*||f_i)$, where PW_i^* is a guessed password. $\mathcal{A}$ verifies whether $M_6{=}h(M_2||(M_2 \bigoplus e_i \bigoplus r_i^*))$. If true, $\mathcal{A}$ can obtain a password PW_i^* of legal client C_i. Likewise, for legal client C_i in Li et al.'s scheme [10], $\mathcal{A}$ can also guess C_i's password PW_i^* by computing $RPW_i^*{=}h(N||PW_i^*)$ and $r_i^*{=}h(RPW_i^*||f_i)$, setting $M_8{=}h(y||h(M_2 \bigoplus e_i \bigoplus r_i^*))$ and then checking whether $M_5{=}h(M_2||M_8||M_4)$.

Our design is more secure against the password guessing attacks than Li-Hwang's and Li et al.'s. In our setting, it is difficult for an attacker to derive the client personal biometrics B_i through f_i, due to the protection of the secure one-way hash function. To resist against password guessing attacks, we simultaneously utilize two well-concealed secret values, i.e. C_i's personal biometrics and password, to protect the value r_i=$h(B_i||PW_i||f_i)$. B_i is hidden from the attacker, and so the attacker succeeds with probability at most half.

5 Conclusions

We have proposed a secure and efficient biometrics-based remote user authentication. The proposed scheme can effectively withstand the replay attack,the impersonating attack, and the man-in-the-middle attacks. Compared to the schemes in [9] and [10], not only does the proposed scheme enhance the security, but furthermore, this result reduces the communication and computation costs.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 60773083, by the Provincial Natural Science Foundation of Guangdong under Grants 2008B090500201, 2009B010-800023 and 2010B090400164, and by the Projects in the Scientific Innovation of Jinan University under Grants 11611510.

References

1. Lee W.-B., Yeh C.-K.: A New Delegation-based Authentication Protocol for Use in Portable Communication Systems. IEEE Trans. Wireless Commun., 4(1),57–64 (2005)
2. Burrow M, Abadi M, Needham R. A logic of authentication. ACM Transactions on Computer Systems 1990; 8(1): 18-36.
3. Chang -F, Chang C-C, Su Y-W. A secure improvement on the user-friendly remote authentication scheme with no time concurrency mechanism. In: Proceedings of 20th international conference on advanced information networking and applications, IEEECS, 2006.
4. Hao F, Anderson R, Daugman J. Combining crypto with biometrics effectively. IEEE Transactions on Computing 2006; 55(1): 1081-1088.
5. Juang W, Chen S, Liaw H. Robust and efficient password authenticated key agreement using smart cards. IEEE Transactions on Industrial Electronics 2008; 15(6): 2551-2556.
6. Khan M K, Zhang J, Wang X. Chaotic hash-based fingerprint biometric remote user authentication scheme on mobile devices. Chaos, Solitons and Fractals 2008; 35(3): 519-24.
7. Lee J-K, Ryu S-R, YooK-Y. Fingerprint-based remote user authentication scheme using smart cards. Electronics Letters 2002;38(12): 554-5.
8. Li C-T, Hwang M-S. An online biometrics-based secret sharing scheme for multiparty cryptosystem using smart cards. International Journal of Innovative Computing, Information and Control 2010a; 6(5), 2181-2188

9. Li C-T, Hwang M-S. An efficient biometrics-based remote user authentication scheme using smartcards. Journal of Network and Computer Applications 2010b; 33(1): 1-5.
10. Li X, Niu J-W, Ma J., Wang W-D, Liu C.-L. Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards. Journal of Network and Computer Applications 2011; 34(1):73-79.
11. Lin C-H, Lai Y-Y. A flexible biometrics remote user authentication scheme. Computer Standards and Interfaces 2004; 27(1): 19-23.
12. Matyas J V, Riha Z. Toward reliable user authentication through biometrics. IEEE Security Privacy 2003; 1(3): 45-49.
13. Uludag U, Pankanti S, Jain A K. Biometric cryptosystems: Issues and challenges. Proceedings of the IEEE 2004; 92(6): 948-960.
14. Cao X, Zeng X, Kou W, Hu L. Identity-Based Anonymous Remote Authentication for Value-Added Services in Mobile Networks. IEEE Transactions on Vehicular Technology 2009; 58(7): 3508-3517
15. Yeh K-H, Su C, Lo N W, Li Y, Hung Y-X. Two robust remote user authentication protocols using smart cards. The Journal of Systems and Software 2010; 83: 2556-2565.