Nonlinear Dyn (2013) 71:201–211
DOI 10.1007/s11071-012-0652-3

ORIGINAL PAPER

A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps

Cheng-Chi Lee · Che-Wei Hsu

Received: 16 July 2012 / Accepted: 17 October 2012 / Published online: 30 October 2012

Abstract Recently, biometric-based remote user authentication schemes along with passwords have drawn considerable attention in research. In 2011, Das proposed an improvement on an efficient biometric-based remote user authentication scheme using smart cards and claimed his scheme could resist various attacks. However, there are some weaknesses in Das's scheme such as the privileged insider attack and the off-line password guessing attack. Besides, Das's scheme also cannot provide user anonymity. To overcome these weaknesses, we shall propose a secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps. The proposed scheme not only can resist the above-mentioned attacks, but also provide user anonymity.

Keywords Anonymity · Biometric · Chaotic maps · Mutual authentication · Smart cards

1 Introduction

With regard to the client/server system, the password-based authentication scheme is an essential technique used in order to identify the validity of a remote user [3, 11, 12, 25, 29]. Sun et al. [28] pointed out that password-based authentication schemes have a major problem in that humans are not experts in memorizing text strings. Hence, most users would probably choose easy-to-remember passwords even if they know the passwords might be unsafe. In 2005, Hwang and Liu [7] and Lee and Chiu [14] proposed respectively their traditional remote identity-based authentication schemes. The security of their schemes is only based on the passwords. Consequently, the adversary can use brute force attacks or dictionary attacks to break the passwords if users select weak passwords [13, 15, 16, 26]. In order to solve this problem, cryptographic secret keys and passwords are used in remote user authentication schemes. But the cryptographic secret keys and passwords still have some problems such as the use of long and random keys which are difficult to memorize so that the keys must be stored somewhere, and maintaining the long cryptographic keys is expensive. The cryptographic secret keys and passwords also cannot provide non-repudiation. Because the keys may be forgotten, lost or they may be shared with other people, there is no way to know who the actual user is.

Recently, some biometric-based remote user authentication schemes have been proposed by researches [9, 18, 20]. The biometric system is basically a pattern recognition system which operates by obtaining biometric data from an individual, extracting a feature set from the obtained data and comparing this feature set with the template set in the database [8, 19, 21, 24].

C.-C. Lee (✉) · C.-W. Hsu
Department of Library and Information Science, Fu Jen Catholic University, 510 Jhongjheng Rd., Sinjhuang City, New Taipei City 24205, Taiwan, R.O.C.
e-mail: cclee@mail.fju.edu.tw

Das [2] pointed out the following advantages of biometric keys:

- Biometric keys cannot be lost or forgotten.
- They are very difficult to copy or share.
- They are extremely hard to forge or distribute.
- They cannot be guessed easily.
- They are not easy to break.

As mentioned above, biometric-based remote user authentication schemes are more reliable and secure than traditional password-based remote user authentication schemes. In 2010, Li and Hwang [18] proposed an efficient biometric-based remote authentication scheme using smart cards. Later, Das [2] pointed out that Li and Hwang's scheme has some flaws and proposed an improvement of Li and Hwang's scheme to remedy their flaws. Unfortunately, we found that the Das's scheme was vulnerable to privileged insider attacks, off-line password guessing attacks and also cannot provide user anonymity. To overcome these weaknesses, we propose a secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps. In recent years, cryptography based on chaos theory has been studied widely, such as symmetric encryption schemes [27, 32, 33], S-boxes [31], and hash functions [34, 35]. The proposed a scheme based on chaos theory that can allow the user to anonymously communicate with the server and provide mutual authentication between user and server. The security and performance analysis show that the proposed scheme has low computation and communication cost and also can resist these attacks, which was found in the Das's scheme.

The remainder of this paper is organized as follows. In Sect. 2 we introduce the definitions of Chebyshev chaotic maps and review the Das's scheme, and in Sect. 3 we will show the weaknesses of the Das's scheme. Then the proposed scheme is presented in Sect. 4. Next, we analyze the proposed scheme and show that the scheme can resist several attacks in Sect. 5. Our conclusion is given in Sect. 6.

2 Preliminaries

In this section, we briefly introduce the Chebyshev polynomial used in the proposed scheme and review the Das's scheme.

2.1 Chebyshev chaotic maps

From now on, we briefly describe the Chebyshev polynomials [22] as follows. The Chebyshev polynomial $T_n(x)$ is a polynomial in x of degree n. Let n be an integer, and let x be a variable taking value over the interval $[-1,\ 1]$. The Chebyshev polynomial $T_n(x)$: $[-1,\ 1] \to [-1,\ 1]$ is defined as

$$T_n(x) = \cos\bigl(n \cdot \arccos(x)\bigr).$$

The recurrence relation of the Chebyshev polynomial is defined as

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n \geq 2,$$
$$T_0(x) = 1,$$
$$T_1(x) = x.$$

The $\cos(x)$ and $\arccos(x)$ are the trigonometric functions [1]. They are defined as cos: $R \to [-1, 1]$ and arccos: $[-1, 1] \to [0, \pi]$.

The Chebyshev polynomials exhibit the following two important properties [4, 17]: the semigroup property and the chaotic property.

(1) The semigroup property:

$$\begin{aligned} &T_r\bigl(T_s(x)\bigr) \\ &\quad = \cos\bigl(r\cos^{-1}\bigl(\cos\bigl(s\cos^{-1}(x)\bigr)\bigr)\bigr) \\ &\quad = \cos\bigl(rs\cos^{-1}(x)\bigr) = T_{sr}(x) = T_s\bigl(T_r(x)\bigr), \end{aligned}$$

where r and s are positive numbers and $x \in [-1,\ 1]$.

(2) The chaotic property:

When the degree $n > 1$, the Chebyshev polynomial map $T_n(x)$: $[-1,\ 1] \to [-1,\ 1]$ of degree n is a chaotic map with its invariant density $f^*(x) = 1/(\pi\sqrt{1-x^2})$, for positive Lyapunov exponent $\lambda = \ln n > 0$.

In 2008, Zhang [36] proved that the semigroup property holds for the Chebyshev polynomials defined on interval $(-\infty, +\infty)$, which can enhance the property, as follows:

$$T_n(x) \equiv \bigl(2xT_{n-1}(x) - T_{n-2}(x)\bigr) \bmod p$$

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

$$T_r\bigl(T_s(x)\bigr) \equiv T_{sr}(x) \equiv T_s\bigl(T_r(x)\bigr) \bmod p,$$

so the semigroup property still holds and the enhanced Chebyshev polynomials also commute under composition.

The Chebyshev polynomials have the following two problems, which are assumed to be difficult to handle within polynomial time:

(1) Given two elements x and y, the task of the discrete logarithm problem (DLP) is to find the integer r, such that $T_r(x) = y$.
(2) Given three elements x, $T_r(x)$, and $T_s(x)$, the task of the Diffie–Hellman problem (DHP) is to compute the element $T_{rs}(x)$.

2.2 Review of the Das's scheme

In this section, we describe the Das's scheme [2]. The notation throughout the Das's scheme is summarized in Table 1.

There are four phases in the Das's scheme including the registration phase, login phase, authentication phase, and password change phase. The Das's scheme uses the biometric template pattern matching to perform the user's biometric verification [8]. The user's biometric will be matched against the template pattern stored in the system when the user inputs his/her biometric template. The user will pass the biometric verification if there is a match. We explain the details of each phase in the following.

Table 1 The notation used in the Das's scheme

Notation	Definition
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
$h(\cdot)$	a secure one-way hash function
X_s	a secret information maintained by the server
R_c	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

2.2.1 *Registration phase*

When the remote user C_i wants to login into the system, as shown in Fig. 1, he/she needs to perform the following steps:

(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.
(2) The registration center R_i computes the following:

$$f_i = h(B_i),$$

$$r_i = h(PW_i) \oplus f_i,$$

$$e_i = h(ID_i \parallel X_s) \oplus r_i.$$

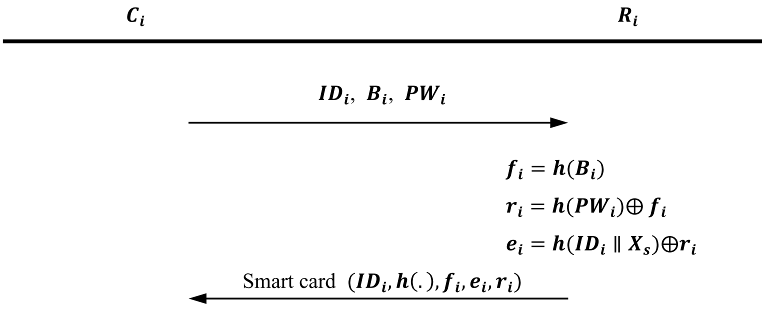
X_s is a secret information generated by the server and is not disclosed to any other for all secure future communications.
(3) R_i embedded $(ID_i, h(.), f_i, e_i, r_i)$ in the user's smart card and sends the card to the user C_i via a secure channel.

2.2.2 *Login phase*

In this phase, when a user C_i wants to login into the server S_i, as shown in Fig. 2, he/she needs to perform the following steps:

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

Fig. 1 Registration phase of the Das's scheme


C_i
R_i
ID_i, B_i, PW_i
$f_i = h(B_i)$
$r_i = h(PW_i) \oplus f_i$
$e_i = h(ID_i \parallel X_s) \oplus r_i$
Smart card $(ID_i, h(.), f_i, e_i, r_i)$

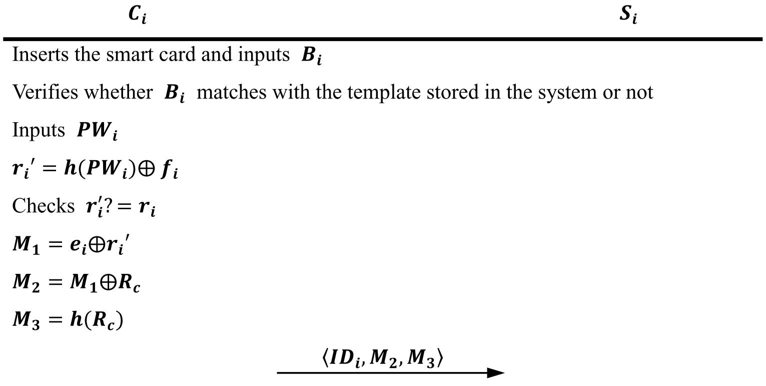


Fig. 2 Login phase of the Das's scheme

C_i | S_i

Inserts the smart card and inputs B_i

Verifies whether B_i matches with the template stored in the system or not

Inputs PW_i

$r_i' = h(PW_i) \oplus f_i$

Checks $r_i' ?= r_i$

$M_1 = e_i \oplus r_i'$

$M_2 = M_1 \oplus R_c$

$M_3 = h(R_c)$

$\xrightarrow{\langle ID_i, M_2, M_3 \rangle}$

(2) C_i verifies whether B_i matches with the template stored in the system or not.

(3) If the above verification does not hold, then C_i does not pass the biometric verification. As a result, the remote user authentication is terminated. Otherwise, if the above verification holds, C_i passes the biometric verification and inputs his/her password PW_i to perform the following step 4.

(4) The smart card computes $r_i' = h(PW_i) \oplus f_i$. The client terminates the session if $r_i' \neq r_i$.

(5) If $r_i' = r_i$, the smart card computes the following:
$M_1 = e_i \oplus r_i'$, which is equal to $h(ID_i \parallel X_s)$,
$M_2 = M_1 \oplus R_c$, which is equal to $h(ID_i \parallel X_s) \oplus R_c$ and
$M_3 = h(R_c)$, where R_c is a random number generated by the user.

(6) Finally, C_i sends the message $\langle ID_i, M_2, M_3 \rangle$ to the remote server S_i.

2.2.3 Authentication phase

After receiving the login request messages $\langle ID_i, M_2, M_3 \rangle$, the server S_i performs the following steps, as shown in Fig. 3.

(1) S_i first checks the format of C_i's ID_i.

(2) If the format is valid, S_i then computes the following:
$M_4 = h(ID_i \parallel X_s)$ using the secret value maintained by the server.
$M_5 = M_2 \oplus M_4$, which needs to be R_c.
S_i verifies $h(M_5) ? = M_3$. If it does not hold, S_i rejects C_i's login request. Otherwise, if the verification is successful, S_i computes the following:
$M_6 = M_4 \oplus R_s (= h(ID_i \parallel X_s) \oplus R_s)$,
$M_7 = h(M_2 \parallel M_5)(= h((h(ID_i \parallel X_s) \oplus R_c) \parallel R_c))$,
$M_8 = h(R_s)$.

(3) S_i sends the messages $\langle M_7, M_6, M_8 \rangle$ to C_i.

(4) After receiving the messages $\langle M_7, M_6, M_8 \rangle$, C_i verifies $M_7 ? = h(M_2 \parallel R_c)$. Thus, C_i terminates the session if the verification does not pass. Otherwise, C_i computes $M_9 = M_6 \oplus M_1$ and verifies $h(M_9) ? = M_8$. If $h(M_9) \neq M_8$, C_i terminates the session. Otherwise, C_i computes $M_{10} = h(M_6 \parallel M_9)(= h((h(ID_i \parallel X_s) \oplus R_s) \parallel R_s))$ and sends the message $\langle M_{10} \rangle$ to the server S_i.

(5) After receiving C_i's message, S_i verifies $M_{10} ? = h(M_6 \parallel R_s)$.

(6) S_i rejects C_i's login request if the above mentioned does not hold.

(7) Thus, S_i accepts C_i's login request if the verification is successful.

2.2.4 Password change phase

In this phase, the smart card always verifies the old entered password by the user before updating the new changed password. In order to change the password, the user performs the following steps:

(1) Inserts the smart card and offers B_i.

(2) Verifies whether B_i matches with the template stored in the system or not.

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

(4) The smart card computes the following:

$$r_i' = h\left(PW_i^{\text{old}}\right) \oplus f_i.$$

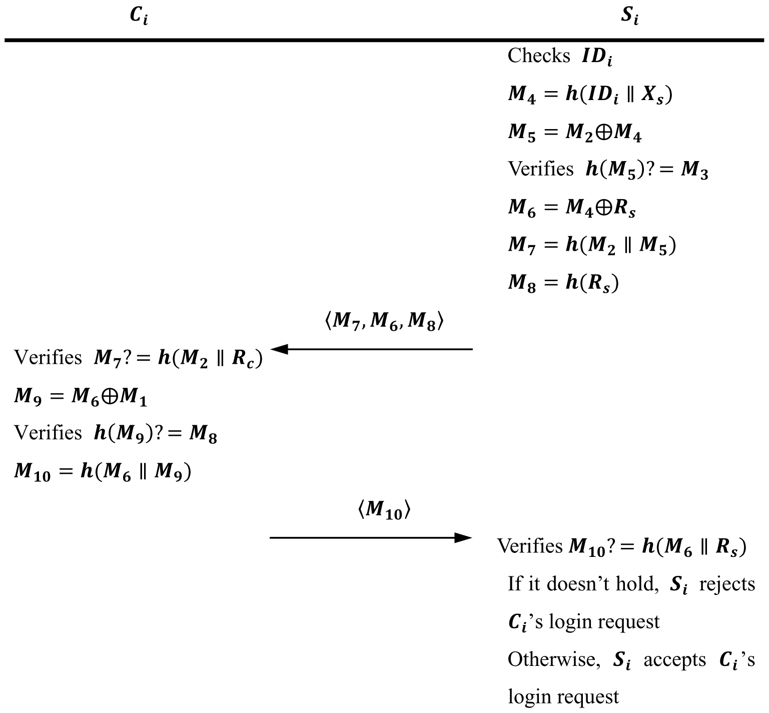

C_i
S_i
Checks ID_i
$M_4 = h(ID_i \parallel X_s)$
$M_5 = M_2 \oplus M_4$
Verifies $h(M_5)? = M_3$
$M_6 = M_4 \oplus R_s$
$M_7 = h(M_2 \parallel M_5)$
$M_8 = h(R_s)$
$\langle M_7, M_6, M_8 \rangle$
Verifies $M_7? = h(M_2 \parallel R_c)$
$M_9 = M_6 \oplus M_1$
Verifies $h(M_9)? = M_8$
$M_{10} = h(M_6 \parallel M_9)$
$\langle M_{10} \rangle$
Verifies $M_{10}? = h(M_6 \parallel R_s)$
If it doesn't hold, S_i rejects C_i's login request
Otherwise, S_i accepts C_i's login request

Fig. 3 Authentication phase of the Das's scheme

If $r_i' \neq r_i$, it means that C_i enters the wrong old password and the password change phase is terminated. If $r_i' = r_i$, then the smart card computes

$$r_i'' = h\left(PW_i^{\text{new}}\right) \oplus f_i,$$

$$e_i' = e_i \oplus r_i'\left(= h(ID_i \parallel X_s)\right),$$

$$e_i'' = e_i' \oplus r_i''.$$

(5) Finally, replaces the e_i with e_i'' and r_i with r_i'' on the smart card.

3 Weaknesses of the Das's scheme

In this section, we analyze the security of the Das's scheme. We show that the Das's scheme is vulnerable to privileged insider attack and the off-line password guessing attack. In addition, the Das's scheme cannot provide a user anonymity. We now describe the details in the following.

3.1 Privileged insider attack

In a real environment, it is a common practice that many users use the same password to access different applications or servers for convenience in remembering long passwords and ease-of-use whenever required [6]. However, if a privileged insider of the registration center knows the password of the user C_i, he/she may try to impersonate C_i for accessing other servers where C_i could be a registered user. In the Das's scheme, the user C_i sends his/her real identity ID_i and password PW_i to the registration center R_i directly in the registration phase. Hence, the privileged insider could get C_i's password and use it to impersonate C_i for accessing different applications or servers. Consequently, the Das's scheme is vulnerable to the privileged insider attack.

3.2 Off-line password guessing attack

Kocher et al. [10] and Messerges et al. [23] have pointed out that all the information in smart cards could be extracted by the side channel attack. We assume that an adversary has stolen user C_j's smart card and extracted the information $(ID_j, h(.), f_j, e_j, r_j)$ of the smart card in the Das's scheme. Using the extracted f_j and r_j, the adversary could find the password PW_j of user C_j through the following steps.

(1) The adversary uses f_j and r_j to compute $h(PW_j) = f_j \oplus r_j$.

(2) Then, the adversary chooses a password PW'_j and verifies $h(PW'_j)? = h(PW_j)$.
(3) If $h(PW'_j) = h(PW_j)$, the guess was correct. Otherwise, the adversary can make another guess and repeat the process.

As mentioned above, we show that an adversary can get the password of user C_j and use it to impersonate C_i for accessing different applications or servers. Hence, the Das's scheme is vulnerable to off-line password guessing attack.

3.3 Inability of providing user anonymity

In the Das's scheme, the user C_i sends his/her real identity ID_i to the server S_i directly in the login phase. All other users also send their real identity to the server S_i directly in the login phase. Hence, an adversary can get the real identity of any user by intercepting the messages $\{ID_i, M_2, M_3\}$ transmitted between the user and the server. Consequently, the Das's scheme cannot provide user anonymity.

4 The proposed scheme

In this section, we present our proposed scheme using extended chaotic maps. The notation used in our scheme is summarized in Table 2.

In the beginning, the registration center R_i selects a random number s, a random integer X_s, and computes $SPUB \equiv T_{X_s}(s) \bmod p$. The registration center R_i keeps the master secret key X_s secretly. There are four phases in our scheme: the registration phase, login phase, authentication phase, and password change phase. The detailed steps of these phases are described in the following subsections.

4.1 Registration phase

When the remote user C_i wants to register and become a new legal user in the system, as shown in Fig. 4, he/she needs to perform the following steps:

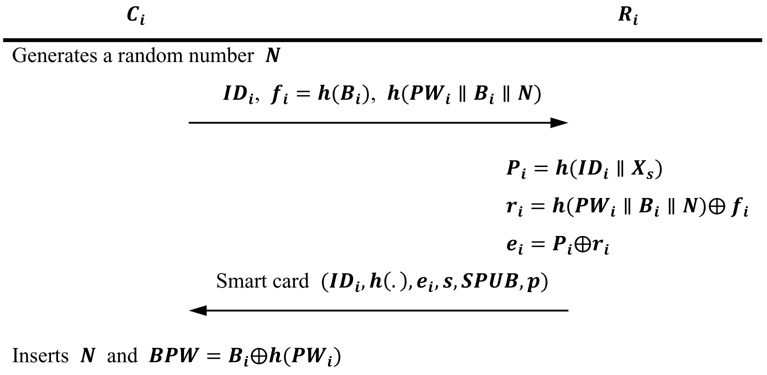
(1) The user offers his/her password PW_i, the identity ID_i, generates a random number N, and also inputs his/her personal biometric B_i on a specific device and computes $f_i = h(B_i)$. C_i then sends $\{ID_i, f_i = h(B_i), h(PW_i \parallel B_i \parallel N)\}$ to the registration center R_i via secure channel.
(2) The registration center R_i computes the following:

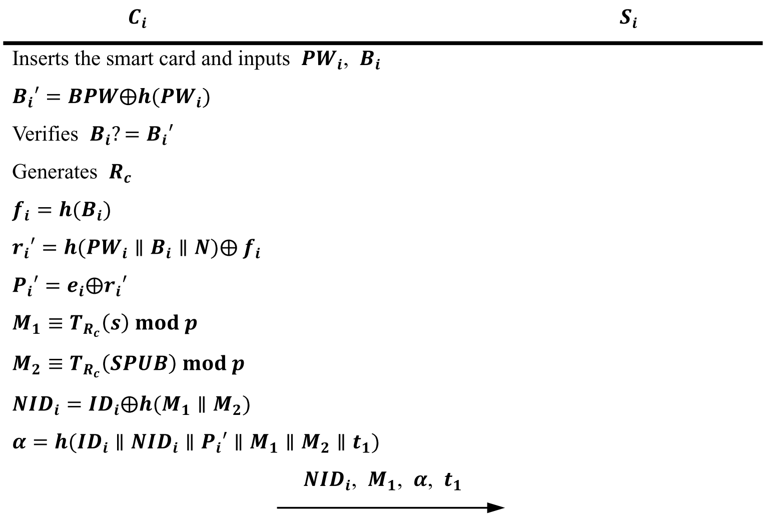
$$P_i = h(ID_i \parallel X_s),$$

Table 2 The notation used in our scheme

Notation	Definition
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
p	a large prime number
X_s	a random integer chosen by the registration center
s	a random number chosen by the registration center
$SPUB$	the public key of R_i, where $SPUB \equiv T_{X_s}(s) \bmod p$
R_c, R_s	two random integers
t_i	the time-stamp
$h(\cdot)$	a secure one-way hash function
$\parallel$	the concatenation operation
$\oplus$	the exclusive-or (XOR) operation

Fig. 4 Registration phase of our scheme


C_i
R_i
Generates a random number N
$ID_i,\ f_i = h(B_i),\ h(PW_i \parallel B_i \parallel N)$
$P_i = h(ID_i \parallel X_s)$
$r_i = h(PW_i \parallel B_i \parallel N) \oplus f_i$
$e_i = P_i \oplus r_i$
Smart card $(ID_i, h(.), e_i, s, SPUB, p)$
Inserts N and $BPW = B_i \oplus h(PW_i)$



C_i S_i

Inserts the smart card and inputs PW_i, B_i

$B_i' = BPW \oplus h(PW_i)$

Verifies $B_i ?= B_i'$

Generates R_c

$f_i = h(B_i)$

$r_i' = h(PW_i \parallel B_i \parallel N) \oplus f_i$

$P_i' = e_i \oplus r_i'$

$M_1 \equiv T_{R_c}(s) \bmod p$

$M_2 \equiv T_{R_c}(SPUB) \bmod p$

$NID_i = ID_i \oplus h(M_1 \parallel M_2)$

$\alpha = h(ID_i \parallel NID_i \parallel P_i' \parallel M_1 \parallel M_2 \parallel t_1)$

$\xrightarrow{NID_i,\ M_1,\ \alpha,\ t_1}$

Fig. 5 Login phase of our scheme

$$r_i = h(PW_i \parallel B_i \parallel N) \oplus f_i,$$

$$e_i = P_i \oplus r_i.$$

R_i embedded $(ID_i, h(.), e_i, s, SPUB, p)$ in the user's smart card and sends the card to the user C_i via a secure channel.

(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 the smart card and finishes the registration.

4.2 Login phase

In this phase, when a legal user C_i wants to access the server S_i, as shown in Fig. 5, he/she needs to perform the following steps:

(1) C_i inserts his/her smart card into the card reader and offers both his/her personal biometric template B_i and password PW_i on a specific device.

(2) The smart card computes $B_i' = BPW \oplus h(PW_i)$ 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 R_c and computes

$$f_i = h(B_i),$$

$$r_i' = h(PW_i \parallel B_i \parallel N) \oplus f_i,$$

$$P_i' = e_i \oplus r_i',$$

$$M_1 \equiv T_{R_c}(s) \bmod p,$$

$$M_2 \equiv T_{R_c}(SPUB) \bmod p,$$

$$NID_i = ID_i \oplus h(M_1 \parallel M_2),$$

$$\alpha = h\big(ID_i \parallel NID_i \parallel P_i' \parallel M_1 \parallel M_2 \parallel t_1\big).$$

(4) The user C_i sends $\{NID_i, M_1, \alpha, t_1\}$ to S_i.

4.3 Authentication phase

After receiving the login request messages, the server S_i performs the following steps to access mutual authentication, as shown in Fig. 6.

(1) Upon receiving $\{NID_i, M_1, \alpha, t_1\}$, S_i first checks the validity of t_1 by checking whether the equation $t' - t_1 > \Delta t$ holds, where t' is the time when the server receives the messages from C_i. and Δt denotes the predetermined legal time interval of transmission delay. If the equation holds, S_i rejects C_i.

(2) S_i computes $M_2' \equiv T_{X_s}(M_1) \bmod p$, $ID_i' = NID_i \oplus h(M_1 \parallel M_2')$ and checks the validity of ID_i'.

(3) S_i computes $P_i'' = h(ID_i' \parallel X_s)$ and $\alpha' = h(ID_i' \parallel NID_i \parallel P_i'' \parallel M_1 \parallel M_2' \parallel t_1)$.

(4) Then S_i verifies whether α' equals to α. If $\alpha' \neq \alpha$, S_i stops the session.

(5) If $\alpha' = \alpha$, S_i randomly chooses an integer R_s and computes $M_3 \equiv T_{R_s}(s) \bmod p$ and $\beta = h(ID_i' \parallel P_i'' \parallel M_2' \parallel M_3 \parallel t_2)$. Then, S_i sends $\{M_3, \beta, t_2\}$ to C_i.

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

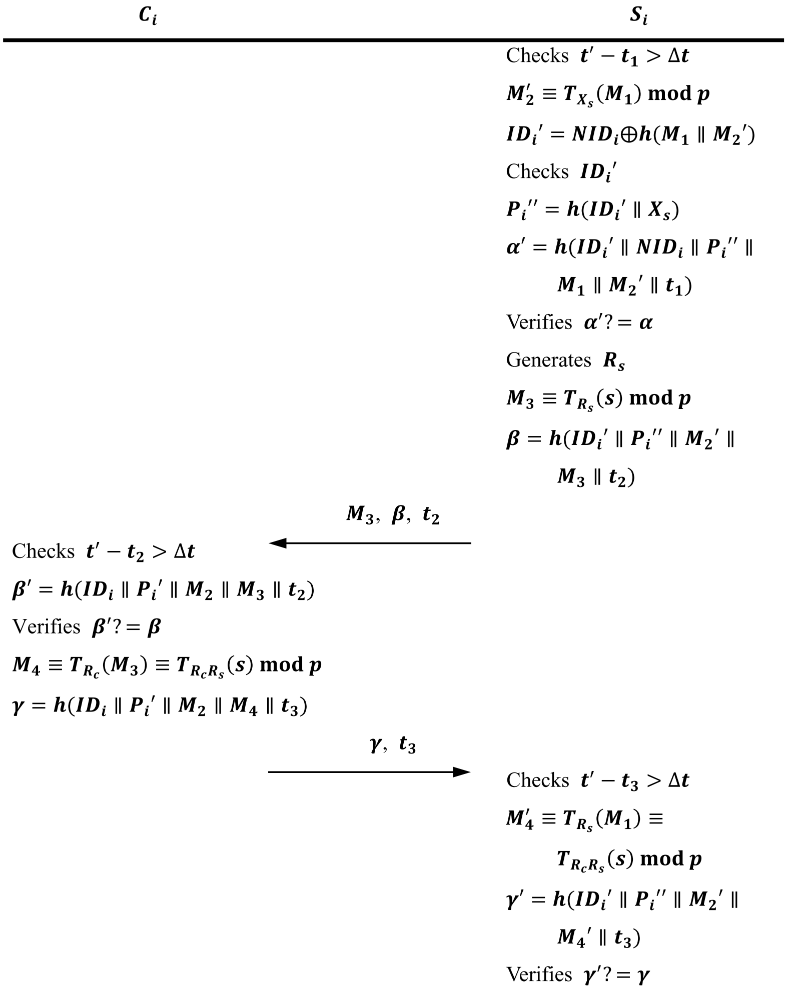


Fig. 6 Authentication phase of our scheme

S_i: Checks $t' - t_1 > \Delta t$

$M_2' \equiv T_{X_s}(M_1) \bmod p$

$ID_i' = NID_i \oplus h(M_1 \parallel M_2')$

Checks ID_i'

$P_i'' = h(ID_i' \parallel X_s)$

$\alpha' = h(ID_i' \parallel NID_i \parallel P_i'' \parallel M_1 \parallel M_2' \parallel t_1)$

Verifies $\alpha'? = \alpha$

Generates R_s

$M_3 \equiv T_{R_s}(s) \bmod p$

$\beta = h(ID_i' \parallel P_i'' \parallel M_2' \parallel M_3 \parallel t_2)$

C_i: Checks $t' - t_2 > \Delta t$

$\beta' = h(ID_i \parallel P_i' \parallel M_2 \parallel M_3 \parallel t_2)$

Verifies $\beta'? = \beta$

$M_4 \equiv T_{R_c}(M_3) \equiv T_{R_cR_s}(s) \bmod p$

$\gamma = h(ID_i \parallel P_i' \parallel M_2 \parallel M_4 \parallel t_3)$

S_i: Checks $t' - t_3 > \Delta t$

$M_4' \equiv T_{R_s}(M_1) \equiv T_{R_cR_s}(s) \bmod p$

$\gamma' = h(ID_i' \parallel P_i'' \parallel M_2' \parallel M_4' \parallel t_3)$

Verifies $\gamma'? = \gamma$

(7) C_i computes $\beta' = h(ID_i \parallel P_i' \parallel M_2 \parallel M_3 \parallel t_2)$ and verifies whether $\beta'? = \beta$. If they are not equal, C_i stops the session. Otherwise, C_i computes $M_4 \equiv T_{R_c}(M_3) \equiv T_{R_cR_s}(s) \bmod p$ and $\gamma = h(ID_i \parallel P_i' \parallel M_2 \parallel M_4 \parallel t_3)$. C_i then sends $\{\gamma, t_3\}$ to S_i.

(8) Upon receiving $\{\gamma, t_3\}$, S_i first checks the validity of t_3 by checking whether the equation $t' - t_3 > \Delta t$ holds. If the equation holds, S_i rejects C_i. Otherwise, S_i computes $M_4' \equiv T_{R_s}(M_1) \equiv T_{R_cR_s}(s) \bmod p$ and $\gamma' = h(ID_i' \parallel P_i'' \parallel M_2' \parallel M_4' \parallel t_3)$ 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 keys M_4 and M_4' to communicate with each other by using a symmetric cryptosystem.

Since $SPUB \equiv T_{X_s}(s) \bmod p$, $M_1 \equiv T_{R_c}(s) \bmod p$, $M_2 \equiv T_{R_c}(SPUB) \bmod p$, and $M_3 \equiv T_{R_s}(s) \bmod p$, so we can derive

$$\begin{aligned} M_2' &\equiv T_{X_s}(M_1) \equiv T_{X_s}\big(T_{R_c}(s)\big) \equiv T_{R_c}\big(T_{X_s}(s)\big) \\ &\equiv T_{R_c}(SPUB) \equiv M_2 \bmod p \end{aligned}$$

and

$$\begin{aligned} M_4' &\equiv T_{R_c}(M_3) \equiv T_{R_c}\big(T_{R_s}(s)\big) \equiv T_{R_s}\big(T_{R_c}(s)\big) \\ &\equiv T_{R_s}(M_1) \equiv M_4 \bmod p. \end{aligned}$$

Therefore, the correctness of the scheme is proved.

4.4 Password change phase

In this phase, the smart card always verifies the old entered password by the user before updating the new changed password. In order to change the password, the user C_i performs the following steps:

(1) Inserts the smart card and offers both the biometric template B_i and old password PW_i.
(2) The smart card computes $B_i' = BPW \oplus h(PW_i)$ and verifies $B_i ?= B_i'$. If $B_i \neq B_i'$, it means that C_i enters the wrong old password or the wrong biometric template. Then, the smart card rejects the request.
(3) If C_i passes the biometric verification, C_i enters his/her new password PW_i^{new}.
(4) The smart card computes the following:

$$f_i = h(B_i),$$
$$r_i' = h(PW_i \parallel B_i \parallel N) \oplus f_i,$$
$$r_i'' = h\big(PW_i^{\text{new}} \parallel B_i \parallel N\big) \oplus f_i,$$
$$P_i' = e_i \oplus r_i',$$
$$e_i' = P_i' \oplus r_i''.$$

(5) Finally, replaces the e_i with e_i' on the smart card.

5 Analysis of the proposed scheme

In this section, we analyze the security and performance of our proposed scheme and show it could overcome the security weaknesses of the Das's scheme. Then, we will describe the details as in the following.

5.1 Security analysis

Here, we describe several security analyses in our proposed scheme.

Privileged insider attack In the registration phase of our scheme, the remote user C_i sends $h(PW_i \parallel B_i \parallel N)$ to the registration center R_i. The privileged insider cannot derive the password PW_i without B_i and N. Therefore, our scheme can resist the privileged insider attack.

Replay attack The attacker may intercept the communication messages from C_i and replay them to the server S_i in next run. However, the attacker cannot pass the verification with the incorrect timestamps. Hence, our scheme is secure against the replay attack by using the timestamps t_1, t_2, and t_3.

Off-line password guessing attack The attacker may intercept the messages $\{NID_i, M_1, \alpha, t_1\}$ and $\{M_3, \beta, t_2\}$. The attacker may also get e_i stored in the smart card. Then he/she could try to guess the password PW_i'. But the attacker cannot verify the correctness of the password PW_i' since he/she does not know the elements r_i, f_i, B_i and P_i. If the attacker wants to derive the random integers R_c and R_s, he/she will also face the DHP. Therefore, our scheme can resist the off-line password guessing attack.

User anonymity The attacker may eavesdrop on the communication between user C_i and server S_i, and try to track the user's real identity to find some information of the user. In our scheme, the real identity ID_i is protected by $M_2 \equiv M_2' \equiv T_{X_s}(T_{R_c}(s)) \bmod p$ from $PUB \equiv T_{X_s}(s) \bmod p$ and $M_1 \equiv T_{R_c}(s) \bmod p$. In order to compute M_2, the attacker will face the DHP. Therefore, our scheme can provide the user anonymity.

Mutual authentication Our scheme can achieve mutual authentication between user C_i and server S_i. In the authentication phase of our scheme, server S_i has to verify the validity of α and γ in order to authenticate C_i. The user C_i's smart card also has to verify the validity of β in order to authenticate S_i. If there is an attacker who wants to forge the messages, he/she will face the DLP and the DHP. Hence, both the user and the server can authenticate with each other, and mutual authentication between them is achieved.

Stolen-verifier attack The stolen-verifier attack means that an attacker steals the security-sensitive verification table from the server and uses it to masquerade as a legitimate user in the authentication phase. The server in our scheme does not need to maintain any security-sensitive verification table. Hence, our scheme can resist the stolen-verifier attack.

Lost smart card Assume that an attacker can extract all the information from the smart card by the side channel attack [10, 23]. The attacker may try to derive the password from the information, but the password is protected by the elements r_i, f_i, B_i and P_i that the attacker does not know. Besides, the attacker also cannot pass the biometric verification without the user's biometric template B_i. Therefore, our scheme is secure against the smart card loss problem.

Table 3 Comparison of security properties

	Tseng et al.'s scheme	Das's scheme	Lee et al.'s scheme	He et al.'s scheme	Our scheme
Privileged attack	No	No	No	No	Yes
Replay attack	Yes	Yes	Yes	Yes	Yes
Off-line guessing attack	Yes	No	Yes	Yes	Yes
User anonymity	No	No	No	Yes	Yes
Mutual authentication	No	Yes	Yes	Yes	Yes

Table 4 Comparison of performance

	Client	Server
Tseng et al.'s scheme	$2T_X + 5T_H + 1T_E + 1T_D + 1T_C$	$1T_X + 3T_H + 1T_E + 1T_D + 2T_C$
Das's scheme	$4T_X + 5T_H$	$4T_X + 8T_H$
Lee et al.'s scheme	$6T_X + 6T_H + 2T_C$	$6T_X + 6T_H + 2T_C$
He et al.'s scheme	$2T_X + 5T_H + 3T_C$	$2T_X + 5T_H + 3T_C$
Our scheme	$5T_X + 10T_H + 3T_C$	$3T_X + 7T_H + 3T_C$

5.2 Performance analysis

Here, we discuss the performance of our proposed scheme. We compare the security properties of our scheme with Tseng et al.'s scheme [30], Lee et al.'s scheme [17], He et al.'s scheme [5], and the Das's scheme [2] in Table 3. We also define some notation as follows:

- T_X: time for performing an XOR operation.
- T_H: time for performing a one-way hash function.
- T_E: time for performing a symmetric encryption operation.
- T_D: time for performing a symmetric decryption operation.
- T_C: time for performing a Chebyshev chaotic map operation.

In Table 3, we can see that our scheme is more secure than other schemes. We also compare the performance of our scheme with other schemes in Table 4. The costs of our scheme are slightly higher than of the Das's scheme. However, the Das's scheme is vulnerable to the privileged insider attack, the off-line password guessing attack, and also cannot provide user anonymity. As a result, our proposed scheme can overcome the weaknesses of the Das's scheme. Hence, our scheme is more secure than the Das's scheme.

6 Conclusions

In this article, we presented a cryptanalysis of the Das's scheme and pointed out its security weaknesses. We have shown that the Das's scheme is vulnerable to the privileged insider attack, the off-line password guessing attack, and also cannot provide user anonymity. To solve these problems, we proposed a secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps. The proposed scheme not only can resist the above-mentioned attacks, but also provide user anonymity. As a result, our scheme could solve the security problems found in the Das's scheme at the cost of increasing the computational costs slightly.

Acknowledgements This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract No.: NSC 101-2221-E-030-018.

References

1. Bergamo, P., D'Arco, P., De Santis, A., Kocarev, L.: Security of public-key cryptosystems based on Chebyshev polynomials. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. **52**(7), 1382–1393 (2005)
2. Das, A.K.: Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards. IET Inf. Secur. **5**(3), 145–151 (2011)

3. Fan, L., Li, J.H., Zhu, H.W.: An enhancement of timestamp-based password authentication scheme. Comput. Secur. **21**(7), 665–667 (2002)
4. Han, S., Chang, E.: Chaotic map based key agreement with/out clock synchronization. Chaos Solitons Fractals **39**(3), 1283–1289 (2009)
5. He, D., Chen, Y., Chen, J.: Cryptanalysis and improvement of an extended chaotic maps-based key agreement protocol. Nonlinear Dyn. **69**(3), 1149–1157 (2012)
6. He, H., Wu, S., Chen, J.: Note on design of improved password authentication and update scheme based on elliptic curve cryptography. Math. Comput. Model. **55**(3–4), 1661–1664 (2012)
7. Hwang, M.S., Liu, C.Y.: Authenticated encryption schemes: current status and key issues. Int. J. Netw. Secur. **1**(2), 61–73 (2005)
8. Jain, A.K., Ross, A., Prabhakar, S.: An introduction to biometric recognition. IEEE Trans. Circuits Syst. Video Technol. **14**(1), 4–20 (2004)
9. Khan, M.K., Zhang, J., Wang, X.: Chaotic hash-based fingerprint biometric remote user authentication scheme on mobile devices. Chaos Solitons Fractals **35**(3), 519–524 (2008)
10. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Lecture Notes in Computer Science, vol. 1666, pp. 388–397. Springer, Berlin (1999)
11. Lamport, L.: Password authentication with insecure communication. Commun. ACM **24**(11), 770–772 (1981)
12. Lee, C.C.: On security of an efficient nonce-based authentication scheme for SIP. Int. J. Netw. Secur. **9**(3), 201–203 (2009)
13. Lee, C.C., Chang, Y.F.: On security of a practical three-party key exchange protocol with round efficiency. Inf. Technol. Control **37**(4), 333–335 (2008)
14. Lee, N.Y., Chiu, Y.C.: Improved remote authentication scheme with smart card. Comput. Stand. Interfaces **27**(2), 177–180 (2005)
15. Lee, C.C., Huang, K.Y., Huang, S.Y.: On-line password guessing attack on Lu-Cao key agreement protocol for secure authentication. J. Discrete Math. Sci. Cryptogr. **12**(5), 595–598 (2009)
16. Lee, C.C., Lin, T.H., Chang, R.X.: A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards. Expert Syst. Appl. **38**(11), 13863–13870 (2011)
17. Lee, C.C., Chen, C.L., Wu, C.Y., Huang, S.Y.: An extended chaotic maps-based key agreement protocol with user anonymity. Nonlinear Dyn. **69**(1–2), 79–87 (2012)
18. Li, C.T., Hwang, M.S.: An efficient biometric-based remote authentication scheme using smart cards. J. Netw. Comput. Appl. **33**(1), 1–5 (2010)
19. Li, C.T., Hwang, M.S.: An online biometrics-based secret sharing scheme for multiparty cryptosystem using smart cards. Int. J. Innov. Comput. Inf. Control **6**(5), 2181–2188 (2010)
20. Lin, C.H., Lai, Y.Y.: A flexible biometric remote user authentication scheme. Comput. Stand. Interfaces **27**(1), 19–23 (2004)
21. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition, 2nd edn. Springer, New York (2009)
22. Mason, J.C., Handscomb, D.C.: Chebyshev Polynomials. Chapman & Hall/CRC Press, London, Boca Raton (2003)
23. Messerges, T., Dabbish, E., Sloan, R.: Examining smart-card security under the threat of power analysis attacks. IEEE Trans. Comput. **51**(5), 541–552 (2002)
24. Prabhakar, S., Pankanti, S., Jain, A.K.: Biometric recognition: security and privacy concerns. IEEE Secur. Priv. **1**(2), 33–42 (2003)
25. Shen, J.J., Lin, C.W., Hwang, M.S.: Security enhancement for the timestamp-based password authentication using smart cards. Comput. Secur. **22**(7), 591–595 (2003)
26. Shen, J.J., Lin, C.W., Hwang, M.S.: A modified remote user authentication scheme using smart cards. IEEE Trans. Consum. Electron. **49**(2), 414–416 (2003)
27. Sheu, L.J.: A speech encryption using fractional chaotic systems. Nonlinear Dyn. **65**(1–2), 103–108 (2011)
28. Sun, H.M., Chen, Y.H., Lin, Y.H.: OPass: a user authentication protocol resistant to password stealing and password reuse attacks. IEEE Trans. Inf. Forensics Secur. **7**(2), 651–663 (2012)
29. Tsai, C.S., Lee, C.C., Hwang, M.S.: Password authentication schemes: current status and key issues. Int. J. Netw. Secur. **3**(2), 101–115 (2006)
30. Tseng, H.R., Jan, R.H., Yang, W.: A chaotic maps-based key agreement protocol that preserves user anonymity. In: IEEE International Conference on Communications, ICC'09, Dresden, pp. 1–6 (2009)
31. Wang, Y., Wong, K.W., Liao, X., Xiang, T.: A block cipher with dynamic S-boxes based on tent map. Commun. Nonlinear Sci. Numer. Simul. **14**(7), 3089–3099 (2009)
32. Wang, X.Y., Yang, L., Liu, R., Kadir, A.: A chaotic image encryption algorithm based on perceptron model. Nonlinear Dyn. **62**(3), 615–621 (2010)
33. Wang, X., Wang, X., Zhao, J., Zhang, Z.: Chaotic encryption algorithm based on alternant of stream cipher and block cipher. Nonlinear Dyn. **63**(4), 587–597 (2011)
34. Xiao, D., Liao, X., Deng, S.: One-way hash function construction based on the chaotic map with changeable-parameter. Chaos Solitons Fractals **24**(1), 65–71 (2005)
35. Xiao, D., Shih, F., Liao, X.: A chaos-based hash function with both modification detection and localization capabilities. Commun. Nonlinear Sci. Numer. Simul. **15**(9), 2254–2261 (2010)
36. Zhang, L.: Cryptanalysis of the public key encryption based on multiple chaotic systems. Chaos Solitons Fractals **37**(3), 669–674 (2008)