Cryptanalysis and Improvement of an Efficient Password Authentication Scheme Based on Smart Card

Raylin Tso[1] Yi-Shiuan Liu[1] Mu-En Wu[2]

[1] Department of Computer Science, National Chengchi University

Taipei 11605, Taiwan, ROC

{raylin, 101753031}@cs.nccu.edu.tw

[2] Department of Mathematics, Soochow University

Taipei, Taiwan, ROC

mnasia1@gmail.com

Received 4 July 2014; Revised 1 November 2014; Accepted 4 December 2014

Abstract. User authentication is an important technology to guarantee that only the legal users can access resources from the remote server. In 2013, based on chaotic maps, Guo and Chang proposed an efficient mutual authentication protocol with user anonymity for the smart card. Unfortunately, this study will demonstrate their scheme could not achieve the user anonymity property, and do not allow changing password freely for the user. Then, we proposed a new method to remedy the weaknesses. The proposed method is secure even if the secret information stored in the smart card is compromised. Only one-way hash function and simple polynomial computations are involved in our protocol. It is more suitable for practice implementation.

Keywords: anonymity, authentication, chaotic map, smart card

1 Introduction

With rapid development of the network technology, password based authentication has been widely used in many areas, such as the remote access control systems, medical systems, banking and payment systems and so on [1]. Currently, due to the cryptographic capacity, low cost, and the portability, the smart card based authentication scheme is becoming more and more important and functional [2-7]. There are many remote user authentication protocols with smart card being proposed to improve security, efficiency, and functionality extensively by many scholars in recent years [2-14]. Moreover, the compromise of user's identity would lead to the tracing of the previous network communications for the same user. To protect from the risk of ID-theft, the user anonymity property is required for the privacy protection user [9,10].

1.1 Motivation and Contributions

In 2008, Juang et al.'s [12] proposed a new password-authenticated key agreement protocol based on elliptic curve cryptosystems. Their scheme not only could provide identity protection but also construct the session key agreement and enhance efficiency by using elliptic curve cryptosystems. Unfortunately, Sun et al.'s proposed an improved scheme to overcome the weakness of Juang et al.'s, including inability of the password-changing and the session key problem [13]. Later, there are many password based authentications with smart card have been proposed to achieve the user anonymity [9,10,11,16,17,20].

Due to the smart card usually does not support powerful computation capability, new secure authentication protocols with less calculation in the smart cards are required[12,13,16,17,20]. In 2013, based on Chebyshev chaotic maps, Guo and Chang [20] firstly proposed password-authenticated key agreement protocol using smart card. Their scheme is efficient since no time-consuming modular exponential computing and scalar multiplication on elliptic curve cryptosystem are involved in the authentication processes. They claimed that their protocol is able to provide user anonymity even though the adversary could extract the data stored in the smart card. However, we will show that Guo and Chang's scheme is still vulnerable to the impersonation attack by using data extracted from his own smart card, and do not allow changing password freely for the user. Moreover, their scheme cannot provide the user anonymity. There, in this paper, we will propose improved method to overcome Guo and Chang's security weaknesses. And our improved scheme needs not to create public key cryptosystems in advance.

The remainder of this paper is organized as follows. In the next section, we give some related works. We present the improved scheme in section 3. In section 4, the security analyses and performance of the proposed scheme are introduced. Finally, some conclusions will be made in the last section.

2 Preliminaries

2.1 Chebyshev Chaotic Maps

Let *n* be an integer number and *x* be a variable with the interval $[-1,1]$; Chebyshev polynomial map $T_n : R \to R$ of degree *n* is defined by the following recurrent relation:

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

For example: some of the Chebyshev polynomials are

$$T_2(x) = 2x^2 - 1, \quad T_3(x) = 4x^3 - 3x, \text{ and } T_4(x) = 8x^4 - 8x^2 + 1.$$

One of the most important properties is that Chebyshev polynomials establishes that $T_r(T_s(x)) = T_{rs}(x)$ and the consequence property is the computation under composition $T_r(T_s(x)) = T_s(T_r(x))$. In order to enhance the security, Zhang [18] proved that property holds for Chebyshev polynomials defined on interval $(-\infty, \infty)$. In this paper, we use the enhanced Chebyshev polynomials: $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \bmod N$, where $n \geq 2$, $x \in (-\infty, \infty)$, and *N* is a large prime number. And it is obvious that $T_r(T_s(x)) = T_s(T_r(x)) = T_{rs}(x)$.

3 Review of Guo and Chang's Scheme

This section briefly reviews Guo and Chang's chaotic maps-based password-authenticated key agreement protocol and then analysis the weakness of the scheme.

3.1 Guo and Chang's Scheme Revisited [20]

This protocol consists of four phases: (1) the parameter generation phase; (2) the registration phase; (3) the authentication phase; and (4) the password change phase.

Parameter Generation Phase: In this phase, a server *S* needs to choose some parameters as follows:

(1) The server chooses a public key scheme based on Chebyshev chaotic maps, and its public key is $(x, T_s(x))$ and its private key is *s*.

(2) The server *S* selects a secure one-way hash function *h*( ).

(3) The server *S* selects a symmetric key cryptosystem with encryption $E_k(\ )$ and decryption $D_k(\ )$, where *k* is the corresponding of symmetric key.

Registration Phase: If the user *U* with identity *ID* would like to register or reregister with the server *S*, *U* and *S* will perform the following steps:

(1) The user *U* selects a password *pw* and a random number *b* and computes $h(pw \| b)$, where "||" is string concatenation operator. Then, the user *U* submits his/her identity *ID* and the value $h(pw \| b)$ to the server *S* for registration over a secure channel.

(2) If the *ID* is valid, the server *S* computes $IM = E_{KS}(ID \| h(pw \| b))$, where *KS* is the master key of the server *S*.

(3) *S* stores the data {*IM*, *h*( ), $E_k(\ )$, $x, T_s(x)$} into a new smart card, and issues this smart card to the user *U* through a secure channel.

(4) *U* stores the random number *b* into the smart card.

Authentication Phase: After completing this phase, the user *U* and the server *S* can achieve the mutual authentication and establish a common session key used in communication. If *U* wants to log into the server *S*, then *U* and the server *S* will perform the following steps:

(1) The user *U* first inserts his/her smart card into a card reader and inputs the password *pw*.

(2) The smart card selects a random number u and computes $KA = T_u(T_s(x))$. The smart card further computes $E_{KA}(h_{pw} \| IM \| T_1)$, where $h_{pw} = h(ID \| h(pw \| b))$. Then the smart card sends the message $\{IM, T_u(x), E_{KA}(h_{pw} \| IM \| T_1)\}$ to the server S, where T_1 is the current timestamp of the smart card.

(3) Upon receiving the message $\{IM, T_u(x), E_{KA}(h_{pw} \| IM \| T_1)\}$, the server S computes $KA = T_s(T_u(x)) = T_u(T_s(x))$ using his/her private key s, and further decrypts $E_{KA}(h_{pw} \| IM \| T_1)$ with the key KA. Then, the server S checks whether $T^* - T_1 \leq \Delta T$, where T^* is the time of receiving the login message and ΔT is the time threshold.

(4) The server S decrypts IM using his/her master key KS and obtains ID' and $h'(pw \| b)$. Then, the server S computes $h'_{pw} = h(ID' \| h'(pw \| b))$ and checks whether $h'_{pw} = h_{pw}$. If the verification is false, the server S terminates this session. If the verification is successful, the server S believes that the user U is valid.

(5) The server S selects a random number s' and computes $T_{s'}(x)$. Then, the server S computes $E_{KA}(T_{s'}(x) \| h(ID \| T_2) \| T_2)$ and sends this message to the smart card, where T_2 is the server S's current timestamp of the system.

(6) After receiving the message $E_{KA}(T_{s'}(x) \| h(ID \| T_2) \| T_2)$, the smart card decrypts this message using KA and obtains T_2. Then it compares whether the delay time is acceptable. The smart card further computes $h'(ID \| T_2)$, and compares whether $h'(ID \| T_2) = h(ID \| T_2)$. If the equation holds, the server is authenticated by the smart card.

(7) The user U and the server S could compute the common session key $SK = T_u(T_{s'}(x)) = T_{s'}(T_u(x))$, respectively.

Password Change Phase: In Guo and Chang's scheme, if the user U wants to change his/her password, he/she performs the following steps:

(1) The user U inserts his/her smart card into a card reader, enters the old password pw, and requests to change the password. Then, the user U enters the new password pw^*.

(2) The smart card first computes the $h'(pw \| b)$ and the new $h(pw^* \| b)$. Then the smart card selects a random number u' and computes $E_{KA'}(h'(pw \| b), h(pw^* \| b), IM)$, where $KA' = T_{u'}(T_s(x))$. Next, the smart card sends the message $\{E_{KA'}(h'(pw \| b), h(pw^* \| b), IM), T_{u'}(x)\}$ to the server.

(3) Upon receiving this message $\{E_{KA'}(h'(pw \| b), h(pw^* \| b), IM), T_{u'}(x)\}$, the server computes $KA' = T_s(T_{u'}(x))$ and decrypts $E_{KA'}(h'(pw \| b), h(pw^* \| b), IM)$ and IM using KA' and KS, respectively. Then the server checks whether $h'(pw \| b) = h(pw \| b)$. If the equation holds, the server computes $IM^* = E_{KS}(ID \| h(pw^* \| b))$, and then replaces IM with IM^* for the smart card.

3.2 Security Analysis of Guo and Chang's Scheme

Based on Chebyshev chaotic maps, Guo and Chang proposed their password-authenticated key agreement protocol. In the enhanced Chebyshev polynomials: $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \bmod N$, where $n \geq 2$, $T_0(x) = 1$, $T_1(x) = x$, $x \in (-\infty, \infty)$, and N is a larger prime number. It is obvious that if given x and s, it is easy to compute $T_s(x) = y$; however, given y, it is very hard to find the exact parameters x and s such that $y = T_s(x)$. There are many pairs x and s such that $y = T_s(x)$. The probability of obtaining the exact x and s are equivalent to performing an exhaustive search on $y = T_s(x)$. Therefore, without the knowledge of x and s, it is very difficult for someone to impersonate the server and the user. On the other hand, from the recurrent relation $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \bmod N$, given y and x, it is computationally tractable to obtain s such that $y = T_s(x)$ when s is not large enough.

In Guo and Chang's scheme [20], they claimed that their scheme could achieve user identity anonymity property. It means that the adversary cannot find out the true user's ID by intercepting the communication message

$\{IM, T_u(x), E_{KA}(h_{pw} \| IM \| T_1)\}$ in their authentication phase, where $IM = E_{KS}(ID \| h(pw \| b))$. The user's *ID* information is encrypted by using the server's master *KS* and common session key *KA*. Even though the secret data $\{IM, h(), E_k(), x, T_s(x)\}$ stored in the smart card are compromised, the adversary also cannot obtain any information about the user's *ID* from *IM*. Since the user's *ID* combined with a random number *b* is encrypted by server's master key *KS*. The corresponding message $IM = E_{KS}(ID \| h(pw \| b))$ is stored in the smart card.

However, in their scheme, when *U* sends the login request message $\{IM, T_u(x), E_{KA}(h_{pw} \| IM \| T_1)\}$ to remote server *S*, *IM* is always kept the same from *U*. The attacker can use *IM* to distinguish each user and to be seen as user's identification. Then, the adversary is able to find any relation between any readings of the same user. Therefore, their scheme uses *IM* as an identity (*ID*), it cannot actually provide the advantage of the user's anonymity property.

In the registration phase of Guo and Chang's scheme [19], the server *S* stores the data $\{IM, h(), E_k(), x, T_s(x)\}$ into a new smart card, and issues this smart card to the user *U*, where *s* is server's private key. Now suppose that the adversary (e.g., U_a) could extract the data *x* and $T_s(x)$ which stored on $U_a{}'s$ smart card. With the message $\{IM, T_u(x), E_{KA}(h_{pw} \| IM \| T_1)\}$ Then, it is possible to find *u* from given the values *x* and $T_u(x)$, so *u* must be very large. With the communication message $\{IM, T_u(x), E_{KA}(h_{pw} \| IM \| T_1)\}$ then the adversary could compute $KA = T_u(T_s(x))$ and decrypt $E_{KA}(h_{pw} \| IM \| T_1)$ with the key *KA* when *u* is not large enough. Thus, the adversary could obtain h_{pw} and impersonate the user *U* for the future communication. Hence, the adversary could attack and access the system. In this situation, Guo and Chang could not provide secure password-authenticated key agreement protocol based on Chebyshev chaotic maps. Therefore, Guo and Chang's protocol also relies on the hard problem of high-degree polynomials when the smart card data *x* and $T_s(x)$ are extracted.

Moreover, in Guo and Chang's scheme, the user cannot freely change his/her password without the remote server joining this password change phase

4 Our New Scheme

In this section, we will propose a more secure user authentication protocol based on Chebyshev chaotic maps. To illustrate the protocol clearly, the Chebyshev chaotic maps $T_n(x)$ used in the improved protocol are the same as Guo and Chang's scheme [20]. In the initialization, the server *S* chooses two secret key *s* and *d*, where *s* and *d* are large numbers. And *h*( ) is a secure one-way hash function with fixed-length output. Our scheme consists of three phases: the registration, the authentication, and the updated password. The details of these phases will be described as follows.

Registration Phase: If the user U_i with identity ID_i would like to register or reregister with the server *S*, U_i and *S* will perform the following steps:

(1) The user U_i selects a password pw_i and a random number b_i and computes $h(pw_i \| b_i)$. Then, the user U_i submits his/her identity ID_i and $h(pw_i \| b_i)$ to the server *S* for registration over a secure channel.

(2) If the ID_i is valid, the server *S* computes $x_i = h(ID_i \| d)$, $T_s(x_i)$, and $V_i = h(ID_i \| d) \oplus h(pw_i \| b_i)$.

(3) *S* stores the data $\{h(), V_i, T_s(x_i)\}$ into a new smart card, and issues this smart card to the user U_i through a secure channel.

(4) (4) U_i stores the random number b_i into the smart card.

Authentication Phase: If U_i wants to log into the server *S*, then U_i and the server *S* will perform the following steps:

(1) The user U_i first inserts his/her smart card into a card reader and inputs the ID_i and password pw_i. Then, the smart card selects a random number *u* and computes $h(ID_i, d) = V_i \oplus h(pw_i \| b_i)$, $KA_i = T_u(T_s(x_i))$, $CID_i = ID_i \oplus KA_i$, $T_u(h(ID_i \| d))$, and $Y_i = h(h(ID_i \| d) \| T_u(h(ID_i \| d)) \| T_1)$ Next, the smart card sends the message $\{CID_i, T_u(h(ID_i \| d)), Y_i, T_1\}$ to the server *S*, where T_1 is the current timestamp of the smart card.

(2) Upon receiving the message $\{CID_i, T_u(h(ID_i \| d)), Y_i, T_1\}$, the server S checks whether $T^* - T_1 \le \Delta T$, where T^* is the time of receiving the login message and ΔT is the time threshold. Then, the server S computes $KA_i = T_s(T_u(h(ID_i \| d))$, $ID_i = CID_i \oplus KA_i$, and $x_i = h(ID_i \| d)$ using his/her secret keys s and d. Next, the server S computes $Y_i' = h(x_i \| T_u(h(ID_i \| d)) \| T_1)$ and checks whether $Y_i' = Y_i$. If the verification is false, the server S terminates this login request for a period of time. If the verification is successful, the server S believes that the user U_i is valid.

(3) The server S selects a random number v and computes $T_v(x_i)$, $KS_i = T_v(T_u(h(ID_i \| d))$ and $R_i = h(ID_i \| x_i \| KS_i \| T_2)$, where $x_i = h(ID_i \| d)$. Then, S sends the message T_2, $T_v(x_i)$, and R_i to the user U_i, where T_2 is the server S's current timestamp of the system.

(4) After receiving T_2, $T_v(x_i)$, and R_i, the smart card compares whether the time T_2 is acceptable. Then, it further computes $KS_i = T_u(T_v(x_i))$ and $R_i' = h(ID_i \| h(ID_i \| d) \| KS_i \| T_2)$, and compares whether $R_i' = R_i$. If the equation holds, the server is authenticated by the user U_i.

From the above authentication phase, the user U_i and the server S could compute the common session key $KS_i = T_v(T_u(h(ID_i \| d)) = T_u(T_v(x_i))$, where $x_i = h(ID_i \| d)$. This authentication process is briefly illustrated in Fig. 1

U_i		Server
(1) *timestamp* T_1		
$h(ID_i \|\| d) = V_i \oplus h(pw_i \|\| b_i)$		
$T_u(h(ID_i \|\| d)), KA_i = T_u(T_s(x_i))$		
$CID_i = ID_i \oplus KA_i$		
$Y_i = h(h(ID_i \|\| d), T_u(h(ID_i \|\| d)), T_1)$		
	$\{CID_i, T_u(h(ID_i \|\| d)), Y_i, T_1\}$ →	(2) *timestamp* $T^*, (T^* - T_1) \le \Delta T$
		$KA_i = T_s(T_u(h(ID_i \|\| d))$
		$ID_i = CID_i \oplus KA_i$, $x_i = h(ID_i \|\| d)$
		$Y_i' = h(h(ID_i \|\| d) \|\| T_u(h(ID_i \|\| d) \|\| T_1)$
		check $Y_i' = Y_i$
(3)*check timestamp* T_2	← $\{T_v(x_i), R_i, T_2\}$	$T_v(x_i), KS_i = T_v(T_u(x_i))$
$KS_i = T_u(T_v(x_i))$		$R_i = h(ID_i \|\| h(ID_i \|\| d) \|\| KS_i \|\| T_2)$
$R_i' = h(ID_i \|\| h(ID_i \|\| d) \|\| KS_i \|\| T_2)$		
check $R_i' = R_i$		

Fig. 1. Authentication Process

Updated Password Phase: In our remedy, if a user wants to arbitrarily update his/her password pw_i, he/she does not need to register with the server *S*. It is very convenient for the user to change his password. Now suppose user U_i would like to change his/her password, he/she only requires to perform the following steps.

(1) U_i inserts the smart card into the smart card reader and then inputs ID_i and his/her old PW_i. Then, the smart card computes $V_i' = V_i \oplus h(pw_i \| b_i) \oplus h(pw_i' \| b_i)$.

(2) Replace V_i with V_i' on the memory of the smart card.

It is accepted because

$$V_i' = V_i \oplus h(pw_i \| b_i) \oplus h(pw_i' \| b_i)$$
$$= h(ID_i \| d) \oplus h(pw_i \| b_i) \oplus h(pw_i \| b_i) \oplus h(pw_i' \| b_i) = h(ID_i \| d) \oplus h(pw_i' \| b_i)$$

5 Security and Performance Analysis

In this section, we are going to explore the securities and the performances of the improvement protocol.

5.1 Security Analysis

In this section, we analyze the security of the improvement method as follows. Based on Guo and Chang's scheme [20], our scheme can overcome the weaknesses indicated previously. The security of the proposed scheme can be shown as follows:

(1) Mutual Authentication

At the beginning of authentication phase, the user U_i selects a random number *u* and sends the message $\{CID_i, T_u(h(ID_i \| d)), Y_i, T_1\}$ to the server *S*, where $h(ID_i \| d) = V_i \oplus h(pw_i \| b_i)$, $KA_i = T_u(T_s(\mathrm{x}_i))$, $CID_i = ID_i \oplus KA_i$, and $Y_i = h(h(ID_i \| d) \| T_u(h(ID_i \| d) \| T_1))$. From the value $T_u(h(ID_i \| d))$, it is very difficult for the adversary to find the exact *u* and $h(ID_i, d)$. The probability of obtaining the exact *u* and $h(ID_i \| d)$ are equivalent to performing an exhaustive search on $T_n(x)$. Without the knowledge $h(ID_i \| d)$ or the password pw_i of U_i, the adversary could not generate the legal messages CID_i and Y_i. Therefore, the server could authenticate the user U_i through checks the validity of Y_i.

With the message $\{CID_i, T_u(h(ID_i \| d)), Y_i, T_1\}$, the server generates a random number *v* and computes $T_v(x_i)$, $KS_i = T_v(T_u(x_i))$ and $R_i = h(ID_i \| x_i \| KS_i \| T_2)$, where $x_i = h(ID_i \| d)$. Then, *S* sends the messages T_2, $T_v(x_i)$, and R_i to the user U_i. Without the server's secret keys *s* and *d*, the adversary cannot obtain the exact ID_i, and $h(ID_i \| d)$ from CID_i, where $KA_i = T_s(T_u(\mathrm{x}_i))$ and $ID_i = CID_i \oplus KA_i$. Then, the user U_i could authenticate the server *S* by checking the validity of R_i. Hence, the proposed scheme could provide mutual authentication.

(2) User Anonymity

In the authentication process, the user U_i's identity ID_i is included in the message $CID_i = ID_i \oplus KA_i$, where $KA_i = T_u(T_s(\mathrm{x}_i))$ and *u* is a fresh random number. However, without the server's secret key *s*, the adversary cannot obtain the exact ID_i from CID_i. Since KA_i is different for each session, then, the adversary cannot trace the same user U_i from the information CID_i. It can protect the user from tracing over network. Even if the secret information $V_i = h(ID_i \| d) \oplus h(pw_i \| b_i)$ and $T_s(x_i)$ stored in U_i's smart card are compromised, the adversary could not easily obtain any information about the user's ID_i. Therefore, the proposed scheme can achieve user anonymity property.

(3) Replay Attack

The adversary may intercept the message $\{CID_i, T_u(h(ID_i \| d)), Y_i, T_1\}$ and replay it to the server. However, the server could find the attack through checks the validity of timestamp T_1. Similarly, the adversary may intercept the message $\{T_2, T_v(x_i), R_i\}$ and replay it to U_i. The user U_i could also find the attack through checks the timestamp T_2. Therefore, our method could withstand the replay attack.

(4) Off-line Password Guessing Attack

The proposed scheme can achieve user anonymity property. For the CID_i is different for each session, the adversary cannot trace the same user U_i from the information CID_i. Therefore, the mutual information of the interactive authentication message does not reduce the entropy of user's password and identity. Moreover, suppose that the adversary gets the data $V_i = h(ID_i \| d) \oplus h(pw_i \| b_i)$ and $T_s(x_i)$ stored in U_i's smart card, where $x_i = h(ID_i \| d)$. Then, the adversary can guess a password $pw_i{}^*$ and compute $h'(ID_i \| d) = V_i \oplus h(pw_i{}^* \| b_i)$. However, the adversary could not verify its correctness from $T_s(x_i)$ since he/she does not have the server's secret key *s*. Therefore, the proposed scheme could against the off-line password guessing attack.

(5) Impersonation Attack

In the proposed protocol, even though the secret information $V_i = h(ID_i \| d) \oplus h(pw_i \| b_i)$ and $T_s(x_i)$ stored in U_i's smart card are compromised, the adversary could not easily forge the user U_i without knowing $h(ID_i \| d)$ or the password pw_i of U_i. In addition, suppose that U_i extracts the data $V_i = h(ID_i \| d) \oplus h(pw_i \| b_i)$ and $T_s(x_i)$ from his/her smart card. The U_i could only obtain his/her $h(ID_i \| d) = V_i \oplus h(pw_i \| b_i) = x_i$. However, it is not helpful for U_i to impersonate someone U_j without knowing the information $h(ID_j \| d)$ and $T_s(x_j)$ of U_j, where $x_j = h(ID_j \| d)$. Therefore, with the message $h(ID_i \| d)$ and $T_s(x_i)$, U_i still needs to break the hash function and Chebyshev chaotic map so as to find the system's secret keys *s* and *d*, respectively. In generally, the length of *d* and *s* are about 512-1024 bits. The probability of obtaining the exact *d* and *s* are equivalent to performing an exhaustive search on $h(ID_i \| d)$ and $T_s(x_i)$, respectively. Therefore, without the knowledge of *d* and *s*, it is very difficult for someone to impersonate the server and users.

(6) The Insider Attack

In the registration of our improvement scheme, the user U_i sends the hash value $h(pw_i \| b_i)$ instead of the password pw_i to the server, where b_i is a random number generated by the user. The privileged insider A (or the server) cannot get the password since it is protected by the secure hash function and random b_i. Therefore, the proposed scheme could against the insider attack.

(7) Forward Security

After a successful mutual authentication, session key $KS_i = T_v(T_u(x_i)) = T_u(T_v(x_i))$ is generated for legal user U_i and the server *S*. However, without the knowledge of $x_i = h(ID_i \| d)$, an adversary cannot easily to obtain the exactly nonce *u* and *v* from the transmission $T_u(x_i)$ and $T_v(x_i)$. Therefore, it is computationally intractable for the adversary to derive the session key KS_i from $T_u(x_i)$ and $T_v(x_i)$. Even if an intruder obtains the current session key KS_i, it is not easy for him to obtain the current value *u* and *v* from KS_i. Without knowing the random numbers *u* and *v*, it is exceedingly difficult for an adversary to create the session key KS_i. Moreover, the nonce *u* and *v* are used for only one time. Hence, the improved scheme can provide forward security even if the current session key KS_i has been compromised. For KS_i is used for one session only, it is not helpful for the intruder to derive from past communication or future transactions.

5.2 Performance Comparison

We use the Chebyshev polynomials to achieve the mutual authentication and establish the common session key. For the Chebyshev chaotic map [15,18,19], given *y*, it is very hard to find the exact parameters *x* and *n* such that $y = T_n(x)$. Thus, without knowing $x_i = h(ID_i \| d)$, the adversary is computationally intractable to obtain the exact *u* and *v* from $T_u(x_i)$ and $T_v(x_i)$, respectively. The security of the proposed improvement protocol no longer totally relies on the hard problem of high-degree polynomials. Therefore, it is not necessary for the user and server to select larger numbers *u* and *v* to compute $T_u(x_i)$ and $T_v(x_i)$ in the authentication phase. Even though the secret information $V_i = h(ID_i \| d) \oplus h(pw_i \| b_i)$ and $T_s(x_i)$ stored in U_i's smart card are compromised, the adversary could not easily derive the validity $x_i = h(ID_i \| d)$ and pw_i from V_i. Hence, no time-consuming modular exponential computing and scalar multiplication on elliptic are required in our authentication processes. Furthermore, the proposed scheme does not need to construct public key cryptosystem in advance. With regard to efficiency, we define related notations to analyze the computational complexity. The notation *E* means the time for one symmetric encryption or decryption, *T* denotes the time for one Chebyshev polynomial computation, and *H* denotes the time for executing the adopted one-way hash function in one's scheme. Note that the times for computing modular addition and exclusive-or are ignored, since they are much smaller than E, T, and H.

We summarize the comparisons of the proposed scheme with Guo and Chang's in Table 1. As shown in Table 1, in Guo and Chang's scheme [20], both the user and the server need to perform two hash function computations ($2H$), three Chebyshev polynomial computations ($3T$), and two symmetric encryption or decryption computations ($2E$) for the authentication phase. In our improved scheme, the computation time for each user to achieve mutual authentication is two hash function computations ($2H$) and three Chebyshev polynomial computations ($3T$). Consequently, the improved method needs three hash function computations ($3H$) and three Chebyshev polynomial computations ($3T$) to achieve mutual authentication for the server. Therefore, the improvement scheme is more efficient than Guo and Chang's scheme.

Table 1. Comparisons of computation costs

Schemes	Computation costs of user	Computation costs of server
Guo and Chang [20]	$2H + 3T + 2E$	$2H + 3T + 2E$
Our new scheme	$2H + 3T$	$2H + 3T$

6 Conclusion

In this paper, we have proposed an improvement to overcome the weaknesses of Guo and Chang's. The proposed method can provide the following characters: (1) no password table is required for the designated servers; (2) users can freely choose their own passwords; (3) users may update their passwords after registration phase; (4) it supplies mutual authentication between the user and the designated server; (5) session key is generated by the user and the remote server for each session; (6) user anonymity property is provided. Without any public key cryptosystem included in the proposed scheme, it is more efficient than that of traditional protocols for the practical applications.

Acknowledgement

The work of Raylin Tso is supported by the Ministry of Science and Technology, Taiwan, R.O.C., under Grant MOST 103-2221-E-004-009-. The work of Mu-En Wu is supported by the Ministry of Science and Technology, Taiwan, R.O.C., under Grant MOST 103-2218-E-031-001-.

References

[1] L. Lamport, "Password Authentication with Insecure Communication," *Communication of ACM*, Vol. 24, pp. 770–772, 1981.

[2] H. M. Sun, "An Efficient Remote User Authentication Scheme Using Smart Cards," *IEEE Transactions on Consumer Electronics*, Vol. 46, pp. 958–961, 2000.

[3] H. Y. Chien, J. K. Jan, and Y. M. Tseng, "An Efficient and Practical Solution to Remote Authentication: Smart Card," *Computer & Security*, Vol. 21, pp. 372–375, 2002.

[4] J. Y. Liu, A. M. Zhou, and M. X. Gao, "A New Mutual Authentication Scheme Based on Nonce and Smart Cards," *Computer Communications*, Vol. 31, pp. 2205–2209, June 2008.

[5] H. F. Zhu, M. Jiang, X. Hao, and Y. Zhang, "Robust Biometrics-based Key Agreement Scheme with Smart Cards towards a New Architecture," *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 6, No. 1, pp. 81–98, January 2015.

[6] C. C. Yang and R. C. Wang, "Cryptanalysis of a User Friendly Remote Authentication Scheme with Smart Cards," *Computers & Security*, Vol. 23, pp. 425–427, 2004.

[7] N. Y. Lee and Y. C. Chiu, "Improved Remote Authentication Scheme with Smart Card," *Computer Standards & Interfaces*, Vol. 27, pp. 177–180, 2005.

[8] J. Xu, W. T. Zhu, and D. G. Feng, "An Improved Smart Card Based Password Authentication Scheme with Provable Security," *Computer Standards & Interfaces*, Vol. 31, No. 4, pp. 177–180, 2009.

[9] M. L. Das, A. Saxena, and V. P. Gulati, "A Dynamic ID-Based Remote User Authentication Scheme," *IEEE Transactions on Consumer Electronics*, Vol. 50, No. 2, pp. 629–631, 2004.

[10] Y. Wang, J. Liu, F. Xiao, and J. Dan, "A More Efficient and Secure Dynamic ID-based Remote User Authentication Scheme," *Computer Communications*, Vol. 32, No. 4, pp. 583–585, 2009.

[11] D, J. He, M. Ma, Y. Zhang, C. Chen, and J. J. Bu, "A Strong User Authentication Scheme with Smart Cards for Wireless Communications," *Computer Communication*, Vol. 34, pp. 367–374, 2011.

[12] W. S. Juang, S. T. Chen, and H. T. Liaw, "Robust and Efficient Password-Authenticated Key Agreement Using Smart Card," *IEEE Transactions on Industrial Electronics*, Vol. 5, pp. 2551–2556, 2008.

[13] D. Z. Sun, J. P. Huai, J. Z. Sun, J. X. Li, J. W. Zhang, and Z. Y. Feng, "Improvements of Juang et al'.s Password-Authenticated Key Agreement Scheme Using Smart Cards," *IEEE Transactions on Industrial Electronics*, Vol. 56, pp. 2284–2291, 2009.

[14] M. S. Hwang, S, K. Chong, and T. Y. Chen, "DoS Resistant ID-Based Password Authentication Scheme Using Smart Cards," *Journal of Systems and Software*, Vol. 83, pp.163–172, 2010.

[15] D. Xiao, X. F. Liao, and S. J. Deng, "A Novel Key Agreement Protocol Based on Chaotic Maps," *Information Sciences*, Vol. 177, pp. 1136–1142, 2007.

[16] X. X.Li , W. D. Qiu, D. Zheng, K. F. Chen, and J. H. Li, "Anonymity Enhancement on Robust and Efficient Password-Authenticated Key Agreement Using Smart Cards," *IEEE Transactions on Industrial Electronics*, Vol. 57, pp. 780–793, 2010.

[17] R. Song, "Advanced Smart Card Based Password Authentication Protocol," *Computer Standards & Interfaces*, Vol. 32, pp. 321–325, 2010.

[18] E. J. Yoon and I. S. Jeon, "An Efficient And Secure Diffie-Hellman Key Agreement Protocol Based on Chebyshev Chaotic Map," *Communications in Nonlinear Science and Numerical Simulation*, Vol. 16, pp. 2383–2389, 2011.

[19] L. H. Zhang, "Cryptanalysis of the Public Key Encryption Based on Multiple Chaotic Systems," *Chaos Solitons Fract*, Vol.37, pp. 669–674, 2008.

[20] C. Guo and C. C. Chang, "Chaotic Maps-Based Password-Authenticated Key Agreement Using Smart Cards," *Communications in Nonlinear Science and Numerical Simulation*, Vol. 18, pp. 1433–1440, 2013.