International Journal of Security and Its Application
Vol.11, No.3 (2017), pp.1-14
http://dx.doi.org/10.14257/ijsia.2017.11.2.01

# A Novel Remote Authentication Scheme with Smart Cards Based On Chaotic Maps

Kai Chain [1*], Wen-Chung Kuo [2] and Jar-Ferr Yang [3]

*[1*] Department of Computer and Information Science, R.O.C. Military Academy, Kaohsiung, Taiwan, R.O.C.*
*[2] Department of Computer Science and Information Engineering, National Yunlin University of Science & Technology, Taiwan, R.O.C.*
*[3] Institute of Computer and Communication Engineering, Department of Electrical Engineering, National Cheng Kung University, R.O.C.*
*[1]chainkai@mail2000.com.tw, [2]simonkuo@yuntech.edu.tw,*
*[3] jfyang@ee.ncku.edu.tw*

***Abstract***

*This article proposes a novel remote authentication scheme with smart card based on chaotic maps, where BAN-Logic is used to verify the security of the structure. Just like an RSA system, the chaotic map-based code system suffers from chosen-message forgery attacks. However, such attacks will not influence the security of the proposed scheme. The proposed scheme, which has higher security than the existed methods has the following advantages: 1) the user can select password at will; 2) any legal owner can use the smart card to perform online registration at the password center; 3) there is a one-to-one relationship between each user and his smart card so creating an identification code for each user is unnecessary; 4) the authentication server (AS) can confirm the remote login request from a user without any difficulties even without the verification table or any private information; 5) no forger can successfully cheat the scheme by resending stolen registration information; 6) no illegal owner can successfully log in to the scheme using the smart card.*

***Keywords****: remote authentication, smart card, chaotic maps, information security, BAN-Logic*

## 1. Introduction

In the current network system, once the servers are connected to it, they are usually shared by many authorized users. According to the conventional password verification method, each user should register his own public identification code (ID) and secret password (PW) to the authentication server (AS). Then, the AS will save the registered IDs and PWs in the password file. When a user wants to log in to the system, he should provide his personal ID and PW to the AS; then, the AS will compare the ID and PW with data saved in the password file to determine whether or not to allow the user to log in to the system. However, this method saves secret information in a password file, which can result in serious security problems. Moreover, any attacker may capture the verification information from legal users and cheat the system by resending the information. In order to overcome the aforementioned problem, we use the one way hash function and other encryption algorithms to encrypt user passwords as test patterns (TPs) and store the registered IDs and TPs in a public inventory, where the inventory is in a so-called verification table. A user can encrypt his password as a test pattern (TPs) before logging in to the system and then send his ID and TP to the AS. The AS will check whether or not the user's ID and TP are consistent with those stored in the system to determine whether

ISSN: 1738-9976 IJSIA

to accept the login request of the user or not. Most previous password verification methods use the above process [20]. However, all these methods are vulnerable to hackers resending stolen verification information to the system. Furthermore, maintaining and managing the verification table creates an additional cost for the authentication center.

Now, smart card is another tool for security access of commerce and computer network communication applications. Many studies have already proposed various methods that do not need to save the verification table in the authentication center [4, 8, 19]. In this way, a user can remotely log in to the system whether the network is safe or not. Of course, many security problems are still waiting to be solved. Over the past decades, many studies have dealt with encryption systems based on the chaos theory [16]. In particular, the chaotic system is used to design communication protocols with high security, or the symmetric encryption protocol [7, 13, 18] and hashing function [2, 17] are used in chaotic maps. Recently, chaotic maps have begun to also be used in digital signatures [1].

Some research has already addressed smart card applications based on chaotic maps [5, 6], but all of them lack security. Thus, we propose a novel smart card remote authentication application based on the characteristics of chaotic maps in order to increase the security and completeness of smart cards and to make chaotic map applications more comprehensive.

The rest of the paper is organized as follows: In Section 2, we explain the features of chaotic maps. In Sections 3 and 4, we propose our scheme and prove its security by BAN-Logic, respectively. In Section 5, we analyze the security and computation performances of the proposed scheme and compare them with those of the other schemes. Finally, we conclude this paper in Section 6.

## 2. Preliminaries

In this section, we first introduce the definitions and related characteristics of chaotic maps [9, 10].

**Definition 1:** Chebyshev polynomial, $T_n(x)$, is a polynomial, where the integer, $n$ denotes the degree of $x$, which is a variable within the interval $[-1, 1]$. Chebyshev polynomial can be derived as:

$$T_n(x) = \cos(n \cdot ar\cos(x)) \quad (-1 \le x \le 1), \tag{1}$$

According to (1), $T_n(x)$ can be further as:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \ (n \ge 2), \tag{2}$$

where $T_0(x) = 1$ and $T_1(x) = x$.

From (1) and (2), we can obtain its characteristic equation as:

$$f(x) = t^2 - 2xt + 1, \tag{3}$$

Then, with respect to $t$ of (3), we can solve its roots as:

$$\alpha = \frac{x + \sqrt{x^2 - 1}}{2} \text{ and } \beta = \frac{x - \sqrt{x^2 - 1}}{2}, \tag{4}$$

Therefore, Chebyshev polynomial can be given by

$$T_n(x) = \frac{(x + \sqrt{x^2 - 1})^n + (x - \sqrt{x^2 - 1})^n}{2} \bmod p. \tag{5}$$

Chebyshev polynomial has two important properties:

*The semi-group property:*

 

$T_r(T_s(x)) = \cos(r\cos^{-1}(\cos(s\cos^{-1}(x)))) = \cos(rs\cos^{-1}(x)) = T_{sr}(x) = T_s(T_r(x))$,
(6)
where *r* and s are positive integer numbers and $x \in [-1, 1]$.
*The chaotic property:*
When the degree $n > 1$, Chebyshev polynomial mapping: $T_n(x)$: $[-1, 1] \rightarrow [-1, 1]$ of degree *n* is a chaotic map with the invariant density $f^*(x) = \frac{1}{\pi\sqrt{1-x^2}}$ for Lyapunov exponent $\lambda = \ln n > 0$.

In order to enhance the property, Zhang[21] proved that the time interval $(-\infty, +\infty)$ that maintains the semi-group property as the definition of Chebyshev polynomial is as follows:

$$T_n(x) \equiv (2xT_{n-1}(x) - T_{n-2}(x)) \bmod P \quad (7)$$

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

$$T_{rs}(x) \equiv T_r(T_s(x)) \equiv T_s(T_r(x)) \bmod P \quad (8)$$

**Definition 2:** For given two elements *x* and *y*, the task of the discrete logarithm problem is to find an integer *s*, such that $T_s(x) = y$.

**Definition 3:** For given three elements *x*, $T_r(x)$, and $T_s(x)$, the task of the Diffie-Hellman problem is to compute element $T_{rs}(x)$.

## 3. The Proposed Scheme

In this section, we propose a novel smart card remote authentication scheme based on chaotic maps. The scheme can be divided into four phases: system initialization, smart card registration, and user login and authentication server verification. These four phases have four main characters: password center (*PC*), user, smart card and authentication server (*AS*).

In the system initialization phase, the *PC* will establish some necessary public and secret parameters. First, the *PC* will assign an initial password to each smart card, and issue a smart card and the corresponding password to a user before the smart card registration phase. Other parameters will be stored in the smart card and only the smart card can read these parameters. In the smart card registration phase, the user can perform on-line registration for his smart card (using the initial password) and select a desired password to register it on the *PC*. Next, the *PC* will generate a test code that corresponds to the password selected by the user, and then store the test code in the smart card. The identification codes of all successfully registered users will be saved in a registration status file maintained by the *PC*. In the user login phase, the user should insert the smart card into any terminal connected to the *AS* and then enter his password. Next, the smart card will create a remote login request message and transmit it to the *AS*. In the authentication server verification phase, the *AS* can examine the remote login requests from users without the verification table or any secret information. In Section 3.1, we describe the symbols that will be used later, and in Section 3.2, we illustrate the four respective phases.

### 3.1. Notations

In this section, notations used in our scheme are described in **Table 1**.

**Table 1. Notations Descriptions**

| *Notations* | *Descriptions* |
|---|---|
| *PC* | Password center |
| *SC* | Smart card |
| $U_i$ | User *i* |
| *AS* | Authentication server |
| $P_C$, $P_S$ | Two big prime numbers, and $P_C < P_S$ |
| $e_S$, $e_C$ | Integers, $e_S \in Z_{P_S}$ , $e_C \in Z_{P_C}$ |
| $d_S$, $d_C$ | $d_S = e_S^{-1} \bmod P_S$ (secret key) · $d_C = e_C^{-1} \bmod P_C$ |
| *h* | One-way hash function |
| *ID* | Identification code |
| *PW* | Password |
| *t'* | The time when the *PC* receives the registration request from a smart card |
| *t"* | The time when the *AS* receives a login verification message |
| $Time_{reg}$, $Time_{log}$ | Timestamps |
| $\delta$ | Transmission delay time between the *PC* and $U_i$ logged in to server |
| $\varepsilon$ | Transmission delay time between the *AS* and $U_i$ logged in to server |
| $A_C$ | Test code |

### 3.2. Description of Each Phase

*A. System initialization phase*

In this phase, the following steps will be performed by the *PC* to prepare the system:

Step 1: Define $P_S$, $e_S$ and $d_S$ for the system.

Step 2: Define $P_C$, $e_C$ and $d_C$ for the smart card, where $P_C < P_S$.

Step 3: Select a one-way hash function.

Step 4: Perform the following steps for the smart card:

4-1: Assign an identification code $ID_C$ and a password $PW_C$ for the smart card.

4-2: Store $ID_C$, ($P_S$, $e_S$), ($P_C$, $e_C$), *h* and $d_C$ in the smart card.

4-3: Calculate a test code $A_C$ for $PW_C$, and store it in the smart card. The definition of the symbol, ⨁ , is exclusive or computation.

$$A_C = T_{d_S}(T_{e_C}(h^2(PW_C)) \oplus h(ID_C) \bmod P_C) \bmod P_S \quad (9)$$

Step 5: Publish $P_S$, $e_S$, $P_C$, and *h*; and store $d_S$ as the secret key.

It is note that all smart cards can have the same $P_S$, $e_S$, $P_C$, and *h*, but have different $ID_C$, $A_C$, $e_C$ and $d_C$. The secret key $d_S$ is used to establish a verification test code for a smart card and its user during the smart card registration phase. If a user wants to join the

 

system, he should have a legal smart card and an initial password $PW_C$ given by the *PC* in advance. The initial password of a smart card can only be used one time when registering the smart card. The system initialization phase is also shown in **Figure 1**.

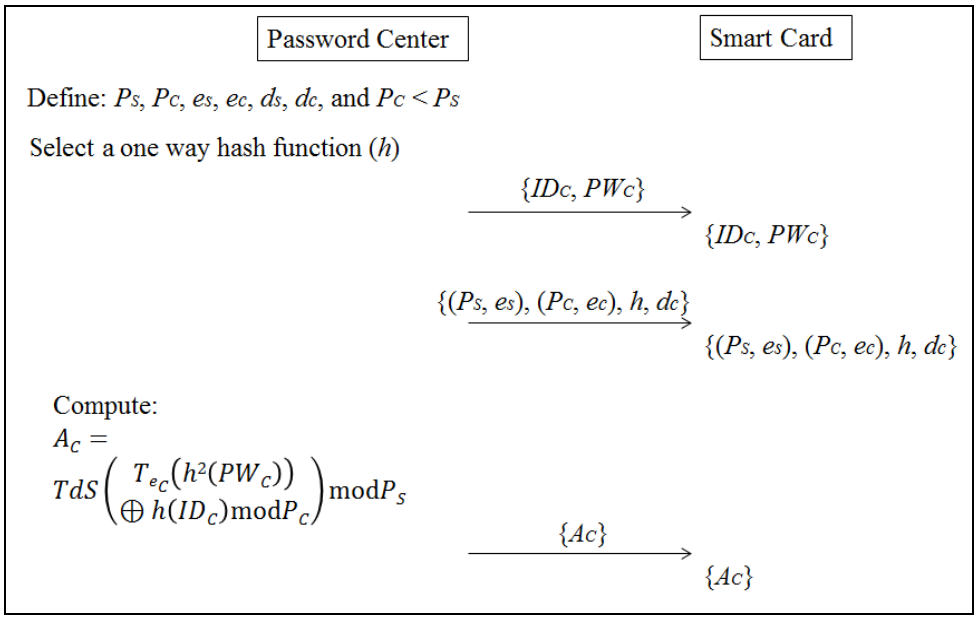


**Figure 1. System Initialization Phase**

*B. Smart card registration phase*

If a user ($U_i$) wants to use his own smart card to join the system, $U_i$ should insert his smart card into a server connected to the *PC*. The on-line registration of the smart card of $U_i$ and the password $PW_i$ selected by $U_i$ are as follows:

Step 1: The smart card asks $U_i$ to input the initial password $PW_C$ and the password $PW_i$ selected by $U_i$.

Step 2: The smart card requests a timestamp ($Time_{reg}$) from the *PC*.

Step 3: The smart card calculates:

$$Y_C = T_{d_C}(h(PW_C) \oplus h(Time_{reg})) \bmod P_C \quad (10)$$

$$Y_i = T_{e_S}(T_{d_C}(h(PW_i) \oplus h(PW_C) \oplus h(Time_{reg})) \bmod P_C \bmod P_S \quad (11)$$

Step 4: The smart card sends {$ID_C$, $Y_C$, $Y_i$, $A_C$, $e_C$, $Time_{reg}$} to the *PC* via a secret path.

Step 5: The *PC* checks whether the deadline of the received timestamp, $Time_{reg}$, is legal by the equation, $| t' - Time_{reg} | \leqq \delta$, where $t'$ is the time when the *PC* received the registration message of the smart card, and $\delta$ is the transmission delay time between the *PC* and the server that $U_i$ logged in to.

If the above equation cannot hold, the registration request will be rejected.

Step 6: The *PC* checks the registration status file to determine whether the received $ID_C$ has been registered by another person.

If the $ID_C$ has been registered by another person, the registration request will be ended.

Step 7: The *PC* checks whether the following equation holds:

$$T_{e_S}(A_C) \bmod P_S = T_{e_C}(h(T_{e_C}(Y_C) \oplus h(Time_{reg}))) \oplus h(ID_C) \bmod P_C \quad (12)$$

If the above equation cannot hold, the registration request will be ended.

Step 8: The *PC* calculates:

$$Z_i = T_{e_C}(T_{d_S}(Y_i) \bmod P_S) \oplus T_{e_C}(Y_C) \bmod P_C$$

(13)

$$A_i = T_{d_S}(T_{e_C}(Z_i) \oplus h(ID_C) \bmod P_C) \bmod P_S$$

(14)

And transmit messages $\{ID_C, A_i\}$ to the smart card.

Step 9: The smart card checks whether $A_i$ is correct by the following equation:

$$T_{e_S}(A_i) \bmod P_S = T_{e_C}(h(PW_i)) \oplus h(ID_C) \bmod P_C$$

(15)

If the above equation cannot hold, the registration request will be ended or will replace the original $A_c$ with $A_i$ in the smart card and ask the *PC* to record *IDc* in the registration file.

The smart card registration phase is shown in **Figure 2.** Step 5 is used to examine, whether they received timestamp, $Time_{reg}$, is legal, to prevent attackers impersonating a legal user ($U_i$) to register on a *PC* with stolen legal registration messages. Via a complete on-line registration stage, each registered user will have a unique one-to-one relation with his own smart card. As the verification message $A_i$ is used as the test code of $ID_C$ and $PW_i$, only the user and his own smart card can create a legal remote verification message.

We can see that the registration message communicated between the smart card and the *PC* does not leak the initial password or the plain text selected by the user; moreover, the *PC* only knows the coded form of the user's password. The reason that the *PC* maintains the registration file is to prevent an illegal card owner from using the card to perform registration again. Therefore, each user can only register his smart card once.

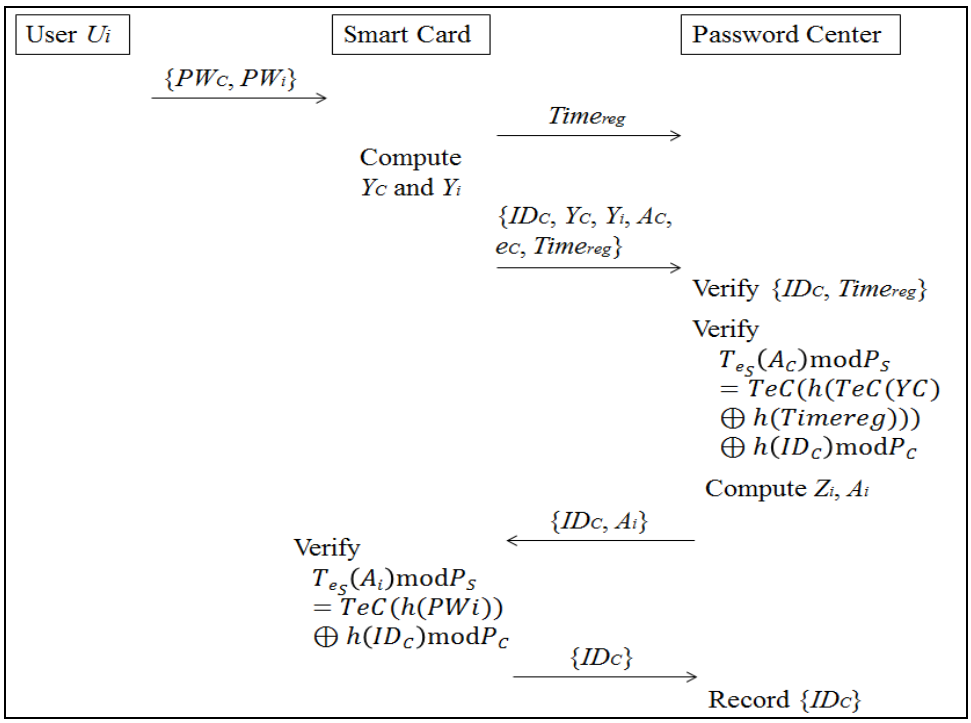


**Figure 2. Smart Card Registration Phase**

In the smart card registration phase, Step 7 is used to examine $A_c$ in order to make the *PC* believe that the smart card to be registered is legal; afterward, Step 9 examines $A_i$ to make the smart card believe that the user password $PW_i$ has been successfully registered. The examination equation in Step 7 of the smart card registration phase can be derived

 

from (9) of Step 4 in the system initialization phase in Subsection 3. Next, we illustrate why the examination equation of Step 9 of the smart card registration phase can hold.

First, the following equation can be derived from Step 3:

$$T_{d_S}(Y_i) \bmod P_S = T_{d_C}(h(PW_i) \oplus h(PW_C) \oplus h(Time_{reg})) (\bmod P_C)$$

(16)

Next, the follow equation can be derived from Step 8:

$$\begin{aligned} Z_i &= T_{e_C}(T_{d_S}(Y_i) \bmod P_S) \oplus T_{e_C}(Y_C) \bmod P_C \\ &= T_{e_C}(T_{d_S}(h(PW_i) \oplus h(PW_C) \oplus h(Time_{reg}))) \oplus h(PW_C) \oplus h(Time_{reg}) \bmod P_C \\ &= h(PW_i) \oplus h(PW_C) \oplus h(Time_{reg}) \oplus h(PW_C) \oplus h(Time_{reg}) \bmod P_C \\ &= h(PW_i) \bmod P_C \end{aligned}$$

(17)

Therefore,

$$\begin{aligned} A_i &= T_{d_S}(T_{e_C}(Z_i) \oplus h(ID_C) \bmod P_C) \bmod P_S \\ &= T_{d_S}(T_{e_C}(h(PW_i)) \oplus h(ID_C) \bmod P_C) \bmod P_S \end{aligned}$$

(18)

From this result, we can ensure that the examination equation in Step 9 can hold.

*C. User login phase*

When a user ($U_i$) wants to log in to the system, $U_i$ should insert his smart card into any terminal connected to the *AS* and then input his password $PW_i$. Next, the smart card will perform the following steps:

Step 1: Request a timestamp, $Time_{log}$, from the *AS*.

Step 2: Calculate a test code related to $PW_i$ and $Time_{log}$, which is as follows:

$$C_i = T_{d_C}(h(PW_i) \oplus h(Time_{\log})) \bmod P_C$$

(19)

Step 3: Create a verification message $\{ID_C, A_i, C_i, Time_{log}, e_C\}$ only for *Ui*, and transmit the message to the *AS*.

The above steps are illustrated in **Figure 3**.

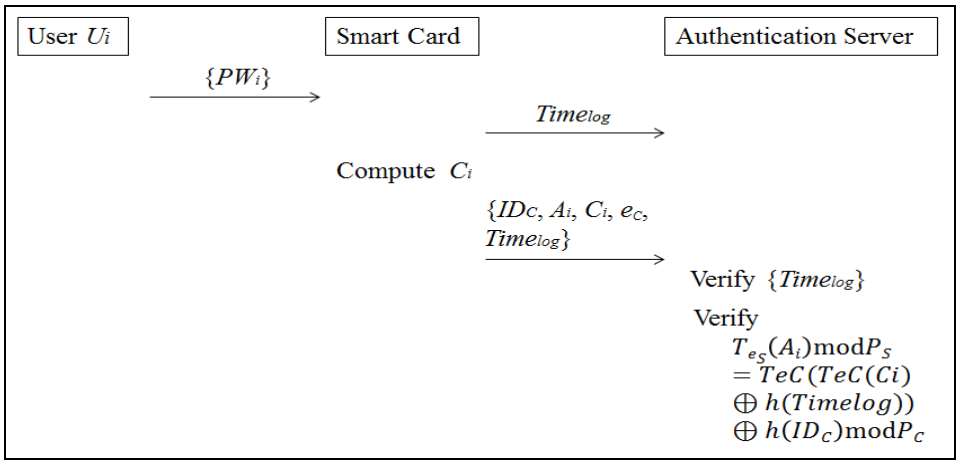


**Figure 3. User Login Phase and Authentication Server Verification Phase**

*D. Authentication server verification phase*

After receiving the message $\{ID_C, A_i, C_i, Time_{log}, e_C\}$, the *AS* performs the following steps to confirm whether $U_i$ is legal or not:

Step 1: Check whether the deadline of the received timestamp, $Time_{log}$, is legal. Let $t''$ be the time when the *AS* received the registration verification message, and then

check whether the equation, $| t'' - Time_{log} | \leqq \varepsilon$, holds or not, where $\varepsilon$ is the transmission delay time between the *AS* and the terminal that *Ui* logged in to.

If the above equation holds, the registration request of $U_i$ will be accepted; otherwise, the registration request of $U_i$ will be denied.

Step 2: Check whether the following equation holds or not:

$$T_{e_S}(A_i) \bmod P_S = T_{e_C}(T_{e_C}(C_i) \oplus h(Time_{\log})) \oplus h(ID_C) (\bmod P_C) \quad (20)$$

If the above equation holds, the registration request of $U_i$ will be accepted; otherwise, the registration request of $U_i$ will be denied.

The authentication server verification phase is shown in **Figure 3**, which explains that when verifying a remote login request, the *AS* does not have to use the verification table or any secret information. The confirmation equation of step 2 of the authentication server verification phase can be derived from the following process:

The following equation can be derived from step 9 of the smart card registration stage:

$$T_{e_S}(A_i) \bmod P_S = T_{e_C}(h(PW_i)) \oplus h(ID_C) \bmod P_C \quad (21)$$

Therefore, the following equation can be derived from Steps 2-3 of the user login phase:

$$\begin{aligned} & T_{e_C}(T_{e_C}(C_i) \oplus h(Time_{\log})) \oplus h(ID_C) \bmod P_C \\ & = T_{e_C}(h(PW_i) \oplus h(Time_{\log}) \oplus h(Time_{\log})) \oplus h(ID_C) \bmod P_C \\ & = T_{e_C}(h(PW_i)) \oplus h(ID_C) \bmod P_C = (T_{e_S}(A_i) \bmod P_S) \bmod P_C \end{aligned} \quad (22)$$

In other words, the confirmation equation of Step 2 of the authentication server verification phase holds.

## 4. BAN-Logic Verification

### 4.1 Introduction of BAN-Logic

Ban-Logic is mainly used to verify the security of the mutual verification process between the password center (*PC*), user ($U_i$), smart card (*SC*) and authentication server (*AS*). In the proposed scheme, the following five main items need to be verified:

(1) *SC* believes $U_i$ is true.
(2) *PC* believes *SC* is true.
(3) *PC* believes $U_i$ is true.
(4) *AS* believes *SC* is true.
(5) *AS* believes $U_i$ is true.

According to the characteristics of the security analysis of BAN-Logic [3, 11], several symbol expressions should be observed, which are as follows:

Characteristic 1: (*X*, *Y*): *X* and *Y* are the members of (*X*, *Y*).
Characteristic 2: <*X*>*Y*: can obtain *X* via the secret parameter *Y*.
Characteristic 3: {*X*}*K*: can encrypt *X* via the key *K*.
Characteristic 4: $\xrightarrow{K} U$ : *K* is the public key of the entity *U*.
Characteristic 5: *P*←-*K*-→*Q*: *P* and *Q* can use the commonly shared key *K* to communicate with each other; any third parties except for *P* and *Q* cannot know about the existence of *K*.
Characteristic 6: *P*<=*S*=>*Q*: *S* is only known by *P* and *Q*; therefore, *P* and *Q* can use *S* to verify one another's identity.

The several inference rules for the security analysis of BAN-Logic are as follows:

Inference rule 1: Freshness conjunction rule

 

Definition: Remove the secret parameter $Y$ or key $K$ from the outermost expression rule of the original message.
Inference rule 2: Break conjunction rule
Definition: When a message is completely trusted, all the parameters n the message can be trusted.
Inference rule 3: Message-mean rule
Definition: Remove the secret parameter $Y$ and key $K$ from the remaining message.
Inference rule 4: Nonce-verification rule
Definition: If there are independent random numbers, they can be removed according to the rule.
Inference rule 5: Jurisdiction rule
Definition: If $A$ trusts $B$ and $B$ trusts $C$, $A$ can directly trust $C$.

### 4.2 Authentication Proof based on BAN-logic

We use the security analysis of BAN-Logic to prove the security of the mutual verification process of the proposed scheme and the message communication processes in all four phases. Before the security analysis of BAN-Logic, these message communication processes should be expressed with the aforementioned basic symbols.
M1. $PC \rightarrow SC$: $\{ID_C, \mathrm{PW}_C, (P_S, e_S), (P_C, e_C), h, d_C, A_C\}$
M2. $U_i \rightarrow SC$: $\{PW_C, PW_i\}$
M3. $SC \rightarrow PC$: $\{ID_C, Y_C, Y_i, A_C, e_C, Time_{reg}\}$
M4. $PC \rightarrow SC$: $\{ID_C, A_i\}$
M5. $U_i \rightarrow SC$: $\{PW_i\}$
M6. $SC \rightarrow AS$: $\{ID_C, A_i, C_i, e_C, Time_{log}\}$

Furthermore, we should have hypothetical conditions and then use BAN-Logic to prove our hypothetical results to ensure that the results are consistent with our inference processes. The hypothetical conditions are as follows:
A1: $PC$ believes $P_S$, $e_S$, $P_C$ and $h$.
A2: $PC$ believes $SC$, and $SC$ believes $P_S$, $e_S$, $P_C$ and $h$.
A3: $PC$ believes $ID_C$, $PW_C$, $e_C$, $d_C$, and $A_C$.
A4: $PC$ believes $SC$, and $SC$ believes $ID_C$, $PW_C$, $e_C$, $d_C$, and $A_C$.
A5: $PC$ believes $A_C$.
A6: $SC$ believes $A_C$.
A7: $U_i$ believes $PW_C$ and $PW_i$.
A8: $U_i$ believes $SC$, and $SC$ believes $PW_C$ and $PW_i$.
A9: $SC$ believes $ID_C$, $Y_C$, $Y_i$, $A_C$, $e_C$, and $Time_{reg}$.
A10: $SC$ believes $PC$, and $PC$ believes $ID_C$, $Y_C$, $Y_i$, $A_C$, $e_C$, and $Time_{reg}$.
A11: $PC$ believes $A_i$.
A12: $SC$ believes $A_i$.
A13: $AS$ believes $A_i$.
A14: $U_i$ believes $PW_i$.
A15: $U_i$ believes $SC$, and $SC$ believes $PW_i$.
A16: $SC$ believes $ID_C$, $A_i$, $C_i$, $e_C$, and $Time_{log}$.
A17: $SC$ believes $AS$, and $AS$ believes $ID_C$, $A_i$, $C_i$, $e_C$, and $Time_{log}$.

The formal verification processes of the BAN-Logic analysis, which use a series of expressions to perform security analysis, are as follows:
(1) Via M1 and assuming A1, A2 and the freshness conjunction rule, $SC$ will believe $P_S$, $e_S$, $P_C$ and $h$.

Condition 1

(2) Via Condition 1 and assuming A3, A4 and the break conjunction rule, $SC$ will believe $ID_C$, $PW_C$, $e_C$, $d_C$, and $A_C$.

Condition 2

(3) Via Condition 2 and assuming A5, A6 and the message-mean rule, *SC* will believe $A_C$.

Condition 3

(4) Via M2 and condition 2 and assuming A7, A8 and the jurisdiction rule, *SC* will believe $U_i$ is true.

Condition 4

(5) Via M1 and M3 and assuming A9, A10 and the break conjunction rule, *PC* will believe $ID_C$, $Y_C$, $Y_i$, $A_C$, $e_C$, and $Time_{reg.}$

Condition 5

(6) Via Condition 5 and M4 and assuming A11, A12 and the message-mean rule, *SC* will believe $A_i$.

Condition 6

(7) Via Condition 6, M4 and the jurisdiction rule, *PC* will believe *SC* is true.

Condition 7

(8) Via M5 and assuming A14, A15 and Condition 7, *PC* will believe $U_i$ is true.

Condition 8

(9) Via M6 and assuming A16, A17 and the message-mean rule, *AS* will believe $ID_C$, $A_i$, $C_i$, $e_C$, $Time_{log}$.

Condition 9

(10) Via Condition 9 and assuming A12, A13 and the jurisdiction rule, *AS* will believe *SC* is true.

Condition 10

(11) Via Conditions 10, 8, 7, 4 and the jurisdiction rule, *AS* will believe $U_i$ is true.

Condition 11

Through the above processes and Conditions 4, 7, 8, 10 and 11, we can prove that the mutual verification processes between the password center (*PC*), user ($U_i$), smart card (*SC*) and authentication server (*AS*) are safe.

## 5. Security and Computation Cost Analyses

In order to evaluate the effectiveness of the proposed method, the security and computation cost analyses of the proposed and existed methods are addressed in the following two subsections.

### 5.1 Security Analysis

In this subsection, we analyze some attacks that may be used by attackers to crack the registration messages of legal users, legal remote registration requests, the public parameters of the system or smart cards, etc., to cheat the system. The discussions of these security problems that may take place are as follows:

*Security problem 1*: crack the secret parameters of the system and smart cards

If attackers try to crack the secret key $d_S$ through the public keys $e_S$ and *Ps* of the *PC*, they will face the discrete logarithm problem (DLP) [14]. Similarly, if attackers try to crack the secret key $d_S$ of a smart card, they will face the same difficulty.

*Security problem 2*: derive passwords from a stolen message

During the registration and logging in to the system, passwords (including initial password $PW_C$ and user password $PW_i$) are protected by the one-way hash function *h*. That is to say, these passwords are expressed by the coded forms, $h(PW_C)$ and $h(PW_i)$. Thus, any attackers, and even the *PC* or the *AS*, are incapable of deriving passwords from stolen registration or verification messages.

*Security problem 3*: an illegal smart card owner trying to re-register a smart card

No one can use a smart card to carry out new registration unless he knows the password $PW_C$ created during the system initialization stage. If a smart card has already been registered, the $A_i$ in the smart card has a one-to-one relationship with the

 

corresponding identification code $ID_C$ and password $PW_i$. Furthermore, the registration status file maintained by the *PC* will record legal smart cards and their owners. Thus, in examination Steps 6-7 of the smart card registration phase, *PC* will not allow attackers to try to re-register a registered smart card.

*Security problem 4*: attackers trying to re-attack without smart cards

Any attacker may capture a legal verification message $\{ID_C, A_i, C_i, Time_{log}, e_C\}$ and then try to impersonate *Ui* to resend the message. However, this kind of attack will be eliminated by Step 1 of the authentication server verification phase.

*Security problem 5*: attackers trying to log in to the scheme without smart cards

Obviously, any attacker who knows the secret key $d_S$ of the *PC* and the secret key $d_C$ of a smart card can create a fake $A_i$ or $C_i$ to create a legal login message. However, the security considerations of $d_S$ and $d_C$ that will prevent this have already been discussed in security problem 1.

*Security problem 6*: an illegal smart card owner trying to log in to the scheme

It is impossible for anyone to create a correct test code $C_i$ using a smart card during Step 2 of the user login phase to pass the examination of Step 2 of the authentication server verification phase. Assume that a attacker is trying to create a fake legal registration message with a stolen smart card, but he does not know $d_S$, $d_C$ and $PW_i$. From the point of view of security problem 2, the attacker can obtain $h(PW_i)$ by eavesdropping on a login message. For example, if a attacker selects an acceptable timestamp $Time_{log}$* and acquires the signature that the secret key $d_C$ of the smart card made for the message, $h(PW_i) \oplus h(Time_{\log}*)$, he can easily create a legal registration message $\{ID_C, A_i, C_i*, Time_{log}*, e_C\}$ to pass the examination of the authentication server verification phase, which is a so-called "chosen-plaintext attack". In order to successfully realize the above "chosen-plaintext attack", the attacker should input the corresponding $PW_i$ into the smart card and impersonate the *AS* to send the timestamp $Time_{log}$* to the smart card; afterward, the smart card will use its secret key $d_C$ to make a signature for the message $h(PW_i) \oplus h(Time_{\log}*)$. The security analysis of deriving passwords from captured registration or verification messages has already been discussed in security problem 2. Hence, the attacker cannot perform a chosen-plaintext attack if he does not know $PW_i$.

Finally, the security and the functionality of our scheme comparisons with those of the related schemes [5, 12, 15] are summarized in **Table 2**.

**Table 2. Security and Functionality Comparisons between the Proposed and the Related Schemes**

| | S1 | S2 | S3 | S4 | S5 | S6 |
|---|---|---|---|---|---|---|
| Li *et al*. [12] | Yes | Yes | No | Yes | Yes | No |
| Song. [15] | Yes | Yes | No | Yes | No | No |
| Guo and Chang. [5] | Yes | Yes | Yes | Yes | Yes | No |
| Our scheme | Yes | Yes | Yes | Yes | Yes | Yes |

### 5.2 Computation Cost Analysis

If $t_{CM}$ is the time for the computation of chaotic maps and $t_H$ is the time for the execution of a one-way hash function, the computation complexity analyses of the user login stage and authentication server verification phase are as follows:
The complexity of the user login phase is $t_{CM} + 2\times t_{H,}$ and the complexity of the authentication server verification phase is $3\times t_{CM} + 2\times t_H$. Based on the concept of the pre-computation of $h(PW_i)\text{mod}P_C$, the complexity of the user login phase can be reduced to $t_{CM} + t_H$. Thus, the scheme is appropriate for smart cards.

### 5.3 Computation Cost Comparison

According to the four phases of the proposed scheme, we compare the computational cost of related schemes. **Table 3** illustrates the computational cost comparison between the proposed and related schemes.
C1: computation cost of the system initialization phase;
C2: computation cost of the smart card registration phase;
C3: computation cost the user login phase;
C4: computation cost of the authentication server verification phase;
$t_H$: time of one way hash function operation;
$t_S$: time of symmetric encryption or decryption;
$t_M$: time of scalar multiplication on elliptic curve;
$t_E$: time of modulus exponential operation;
$t_{CM}$: time of chaotic maps.

**Table 3. Computation Cost Comparisons between the Proposed and Related Schemes**

| | C1 | C2 | C3 | C4 |
|---|---|---|---|---|
| Li *et al*. [12] | $2t_H + 3t_S$ | $1t_H$ | $8t_H + 4t_S$ | $10t_H + 10t_S + 1t_M$ |
| Song. [15] | $2t_H + 1t_E$ | - | $3t_H + 1t_S$ | $3t_H + 1t_S + 1t_E$ |
| Guo and Chang. [5] | $1t_S + 1t_{CM}$ | $1t_H$ | $2t_H + 2t_S + 2t_{CM}$ | $2t_H + 3t_S + 3t_{CM}$ |
| Our scheme | $2t_H + 2t_{CM}$ | $2t_H + 2t_{CM}$ | $1t_H + 1t_{CM}$ | $2t_H + 3t_{CM}$ |

The first two phases of the proposed scheme require more computation resources; however, the computation resources spent on the proposed scheme are more efficient than other schemes after users begin to use the proposed scheme.

## 6. Conclusion

Although research regarding smart cards has been proposed before, such as the RSA system and elliptic curve cryptography system, and even many studies involving the systems combining chaotic theories and smart cards have been proposed in recent years, all of them lack security. Therefore, we propose a novel remote authentication scheme that combines smart cards and the chaotic maps technique. Through the description of security analysis, we have proven that the proposed scheme has better security and

 

efficiency. We hope that our research will make the applications of smart card systems more comprehensive.

## Acknowledgments

This work was supported by MOST 105-2221-E-145-002.

## References

[1] K. Chain and W.C. Kuo, "A new digital signature scheme based on chaotic maps", Nonlinear Dynamics., vol. 74, no. 4, **(2013)**, pp. 1003-1012.
[2] S. Deng, Y. Li and D. Xiao, "Analysis and improvement of a chaos-based Hash function construction", Communications in Nonlinear Science and Numerical Simulation., vol. 15, no. 5, **(2010)**, pp. 1338-1347.
[3] Y. Deng, "Based on BAN logic analysis Otway-Rees protocol", Chaohu College Journal., vol. 8, no. 3, **(2006)**, pp. 35-37.
[4] C.I. Fan, Y.C. Chan and Z.K. Zhang, "Robust remote authentication scheme with smart cards", Computers and Security., vol. 24, no. 8, **(2005)**, pp. 619-628.
[5] 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, no. 6, **(2013)**, pp. 1433-1440.
[6] C. Guo, C.C. Chang and C.Y. Sun, "Chaotic maps-based mutual authentication and key agreement using smart cards for wireless communications", Journal of Information Hiding and Multimedia Signal Processing., vol. 4, no. 2, **(2013)**, pp. 99-109.
[7] X.F. Guo and J.S. Zhang, "Secure group key agreement protocol based on chaotic Hash", Information Sciences., vol. 180, no. 20, **(2010)**, pp. 4069-4074.
[8] W.S. Juang, S.T. Chen and H.T. Liaw, "Robust and efficient password-authenticated key agreement using smart cards", IEEE Transactions on Industrial Electronics., vol. 55, no. 6, **(2008)**, pp. 2551-2556.
[9] C.C. Lee and C.W. Hsu, "A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps", Nonlinear Dynamics., vol. 71, no. 1-2, **(2013)**, pp. 201-211.
[10] C.C. Lee, C.T. Li and C.W. Hsu, "A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps", Nonlinear Dynamics., vol. 73, no. 1-2, **(2013)**, pp. 125-132.
[11] T.Y. Li, X.D. Liu, Z.G. Qin and X.F. Zhang, "Formal Analysis for Security of Otway-Rees Protocol with BAN Logic", Proceedings pf the 1st International Workshop on Database Technology and Applications, Wuhan, **(2009)**, pp.590-593.
[12] 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, no. 2, **(2010)**, pp. 793-800.
[13] L.J. Sheu, "A speech encryption using fractional chaotic systems", Nonlinear Dynamics., vol. 65, no. 1-2, **(2011)**, pp. 103-108.
[14] G.J. Simmons, "Contemporary Cryptology: An Introduction, in Contemporary Cryptology: The Science of Information Integrity". Piscatoway, N. J. IEEE Press, **(1992)**, pp.274.
[15] R.G. Song, "Advanced smart card based password authentication protocol", Computer Standards & Interfaces., vol. 32, no. 5-6, **(2010)**, pp. 321-325.
[16] K. Wang, W.J. Pei, L.H. Zou, Y.M. Cheung and Z.Y. HE, "Security of public key encryption technique based on multiple chaotic systems", Physics Letters., vol. A 360, no. 2, **(2006)**, pp. 259-262.
[17] D. Xiao, F. Shih and X. Liao, "A chaos-based hash function with both modification detection and localization capabilities", Communications in Nonlinear Science and Numerical Simulation., vol. 15, no. 9, **(2010)**, pp. 2254-2261.
[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, no. 6, **(2011)**, pp. 2383-2389.
[19] D. He, M. Ma, Y. Zhang, C. Chen and J. Bu, "A strong user authentication scheme with smart cards for wireless communications", Computer communications., vol. 34, **(2011)**, pp. 367-374.
[20] R.C. Wang, W.S. Juang and C.L. Lei, "Robust authentication and key agreement scheme preserving the privacy of secret key", Computer communications., vol. 34, no. 3, **(2011)**, pp. 274-280.
[21] L.H. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems", Chaos Solitons Fractals., vol. 37, no. 3, **(2008)**, pp. 669-674.

 

## Authors

**Kai Chain**, he received the M.S. degree in Electrical Engineering from National Taiwan University in 2003. He received the Ph.D. degree from the Institute of Computer Science and Communication Engineering at National Cheng Kung University under Profs. Chi-Sung Laih and Jar-Ferr Yang in 2015. He is an assistant professor in the Department of Computer and Information Science at the Republic of China Military Academy. His research interests include Network and Information Security, with a concentration on applied Cryptography.

**Wen-Chung Kuo** received the B.S. degree in Electrical Engineering from National Cheng Kung University and M.S. degree in Electrical Engineering from National Sun Yat-Sen University in 1990 and 1992, respectively. Then, He received the Ph.D. degree from National Cheng Kung University in 1996. Now, he is an associate professor in the Department of Computer Science and Information Engineering at National Yunlin University of Science & Technology. His research interests include steganography, cryptography, network security and signal processing.

**Jar-Ferr Yang** received his BS degree from the Chung-Yuan Christian University, Taiwan in 1977, and MS degree from the National Taiwan University, Taiwan in 1979, and Ph. D. degree from the University of Minnesota, Minneapolis, USA in 1988 all in electrical engineering. He has published over 98 journal and 146 conference papers. His research areas include multimedia processing and coding, and their applications in smart living and learning system integrations. He is a Fellow of IEEE for his contributions to fast algorithms and efficient realization of video and audio coding.