



**HAL**  
open science

# Chaotic Chebyshev Polynomials Based Remote User Authentication Scheme in Client-Server Environment

Toan-Think Truong, Minh-Triet Tran, Anh-Duc Duong, Isao Echizen

► **To cite this version:**

Toan-Think Truong, Minh-Triet Tran, Anh-Duc Duong, Isao Echizen. Chaotic Chebyshev Polynomials Based Remote User Authentication Scheme in Client-Server Environment. 30th IFIP International Information Security Conference (SEC), May 2015, Hamburg, Germany. pp.479-494, 10.1007/978-3-319-18467-8\_32 . hal-01345140

**HAL Id: hal-01345140**

**<https://inria.hal.science/hal-01345140v1>**

Submitted on 13 Jul 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Chaotic Chebyshev polynomials based remote user authentication scheme in client-server environment

Toan-Thinh TRUONG<sup>1</sup>, Minh-Triet TRAN<sup>1</sup>  
Anh-Duc DUONG<sup>2</sup>, Isao ECHIZEN<sup>3</sup>

<sup>1</sup>University of Science, VNU-HCM, Hochiminh city, Vietnam  
{ttthinh, tmtriet}@fit.hcmus.edu.vn

<sup>2</sup>University of Information Technology, VNU-HCM, Hochiminh city, Vietnam  
ducda@uit.edu.vn

<sup>3</sup>National Institute of Informatics, Tokyo city, Japan  
iechizen@nii.ac.jp

**Abstract.** Perfect forward secrecy is considered as the most important standard to evaluate a strong authentication scheme. There are many results researched to achieve this property without using hard problems. Recently, the result of Chang et al has some advances such as, the correctness of schemes mutual authentication and session key agreement demonstrated in BAN-logic or the overheads reduction of system implementation. However, in this paper, we prove that their scheme is still vulnerable to impersonation attacks and session key leakage. To overcome those limitations and be practical, we use different notion to propose time efficient scheme conducted in experiment. Our proposed method can be applied for remote user authentication in various scenarios, including systems with user authentication using mobile or wearable devices.

**Keywords:** Authentication, Anonymity, Impersonation, Session key, Chaotic Chebyshev polynomials

## 1 Introduction

Nowadays, wireless communication is the necessary fundamental. With non-stop growth of handheld and wearable devices, there are many online services widely deployed on the Internet. Customers demand an immediate response, privacy and cryptography in their transactions with service providers. Therefore, incorporating mathematical results into user authentication schemes is an inevitable trend.

User authentication is the first task which any online service needs to perform. It is said that two basic standards a scheme should achieve are the security and time efficiency. However, simultaneously obtaining those goals is a difficult mission. As for security, there are many criteria and one of them is exactly user identification. Basic method [1] is storing a verification table including records (identity/password) on server side. When a user logs, the server checks the

existence of identity and password in the table. Although simple, this method is vulnerable to stolen verification attack. Furthermore, providing static identity through common channel is not suitable for some applications, such as mobile pay-TV [2] or online voting. To overcome those limitations, some authors proposed the notion of dynamic identity [3, 4, 5], but these results still have some drawbacks such as, symmetric message easy to replay attack or poor design easy to information injection attack. In general, most schemes employ one-way hash function which does not provide scheme with strong security. To enhance security, however, a method of using hard problems is more and more given consideration.

Typically, RSA [6] is one of popular methods incorporated into user authentication scheme, but main disadvantage is using certificates leading to additional computations to its verification. Clearly, this is not suitable for resource-limited handheld or wearable devices. Some authors publish the results [7, 8, 9] based on elliptic curve are considered reasonable for time efficiency and security. However, those results use a special kind of hash function, Map-To-Point which has non-negligible cost and is not standardized. Also, Chebysev polynomial is given consideration [10, 11] and its semi-group property is widely applied in global mobile networks environment [12] or public key based cryptosystems [13]. There are some algorithms [14] used in public key cryptosystem based on this approach. It is said that authentication scheme using Chebysev polynomial is better way to keep the tradeoff between time efficiency and security.

In 2013, Chang et al proposed the time efficiency scheme [15] with one-way hash assumption about collision resistant. Besides, the correctness of the scheme is proved based on BAN-logic [16]. Their scheme truly has some successes, for example, providing mutual authentication, achieving session key establishment and without using time-synchronized mechanism. However, their basic limitations are that challenge is only derived from server side and distribution of common secret information to all valid members. In this paper, we prove that Chang et al.'s scheme does not resist impersonation attack and fail to protect session key. Furthermore, it does not provide users anonymity in their transactions. Next, we apply semi-group of Chebysev polynomial for tradeoff balance and session key protection in generic client-server environment in which this approach has not been considered. In addition, our design has challenges derived from two parties, client and server, to make the fairness in transaction. Also, our scheme is proven correct according to BAN-logic. It is said that our result truly is enhanced security and efficiency in practice, including systems with user authentication using handheld or wearable devices to create smart interactive environments.

The remainder of this paper is organized as follows: section 2 quickly reviews Chang et al.'s scheme and discusses its limitations. Then, proposed scheme is presented in section 3, while section 4 discusses the security and efficiency of proposed scheme. Finally, our conclusions are presented in section 5.

## 2 Review and Cryptanalysis of Chang et al.'s Scheme

In this section, we review Chang et al.'s scheme [15] and show that their scheme is vulnerable to impersonation attack. Besides, it cannot provide user's anonymity.

### 2.1 Review of Chang et al.'s Scheme

In this subsection, we review Chang et al.'s scheme. Their scheme includes four phases: registration phase, authentication phase, password change phase, and lost card revocation phase. Below are some important notations in this scheme:

- $U_i$ :  $i^{th}$  user.
- $id_i$ :  $U_i$ 's identity.
- $pw_i$ :  $U_i$ 's password.
- $S$ : Remote server.
- $id_s$ :  $S$ 's identity.
- $x, y$ : The secret keys of remote server.
- $h(\cdot)$ : A cryptographic one-way hash function.
- $sn_i$ : Smart card's serial number.
- $SK$ : Common session key.
- $SC$ : Smart-card.
- $\oplus$ : exclusive-or operation.
- $\parallel$ : concatenation operation.

**Registration Phase**  $U_i$  freely chooses a fixed length  $id_i$  and  $pw_i$ . Then  $U_i$  has to submit his/her  $id_i, pw_i$  to  $S$  through a secure channel. When receiving  $U_i$ 's message,  $S$  performs following steps.

- $S$  randomizes 128-bit sized integer  $r_i$ . Then,  $S$  computes  $R_1 = h(id_i \parallel x \parallel r_i)$ ,  $R_2 = g^{xy} \text{ mod } p$ , where  $p$  is a large prime number and  $g$  is a primitive element in  $Z_p^*$ , and  $R_3 = h(id_i \parallel R_2) \oplus h(pw_i)$ .
- $S$  issues a  $SC$  with a 32-bit sized  $sn_i$ , where  $sn_i$  has a specific format. Then,  $S$  combines  $sn_i$  with  $U_i$ 's  $id_i$  as  $SID_i = (id_i \parallel sn_i)$ .
- Finally,  $S$  saves  $R_1, R_2, R_3, SID_i$  and  $h(\cdot)$  into  $SC$  and send it to  $U_i$  via a secure channel.

In this registration phase, we see that there are some problems: Because  $U_i$  sends plain  $pw_i$  to  $S$ ,  $S$  knows user's true password and may try using it in another system. Furthermore, using two secret keys  $x$  and  $y$  is more security, but we should use only one with high entropy for enough security. Therefore, we will change this in our registration.

**Authentication Phase** When  $U_i$  accesses  $S$ ,  $U_i$  inserts  $SC$  into terminal device and provides  $id_i$  and  $pw_i$ . Then  $SC$  performs following steps.

- $SC$  computes  $C_1 = R_3 \oplus h(pw_i)$  and  $V_1 = R_1 \oplus C_1$ .

- Next,  $SC$  randomly generates a 160-bit sized integer  $n_1$ , then computes and  $DID_i = h(R_2 \parallel n_1) \oplus SID_i$ .
- Finally,  $SC$  sends  $m_1 = \{DID_i, V_1, n_1\}$  to  $S$  via common channel.
- Upon receiving  $m_1$  from  $U_i$ ,  $S$  re-computes  $SID_i = DID_i \oplus h((g^{xy} \bmod p) \parallel n_1)$ . Then,  $S$  retrieves  $id_i$  and  $sn_i$  and checks their format. If they are valid,  $S$  continues to compute  $R_1^* = V_1 \oplus h(id_i \parallel (g^{xy} \bmod p))$  and randomly generates 160-bit sized integer  $n_2$ .
- Next,  $S$  computes  $V_2 = h(R_1^* \parallel id_s \parallel n_1)$ ,  $V_3 = h(h(id_i \parallel (g^{xy} \bmod p)) \parallel n_1) \oplus n_2$  and send  $m_2 = \{id_s, V_2, V_3\}$  to  $U_i$  via common channel.
- Upon receiving  $m_2$  from  $S$ ,  $SC$  computes  $V_2^* = h(R_1 \parallel id_s \parallel n_1)$  and check if  $V_2^* \neq V_2$ . If it holds,  $S$  is successfully authenticated; otherwise, the connection is terminated.
- $SC$  obtains random value  $n_2 = V_3 \oplus h(C_1 \parallel n_1)$  and generates  $SK = h(n_1 \parallel SID_i \parallel R_2 \parallel n_2)$ .
- Finally,  $SC$  computes  $V_4 = h(SK \parallel (n_2 + 1))$  and send  $m_4 = \{V_4\}$  to  $S$  via common channel.
- After receiving  $m_4$  from  $U_i$ ,  $S$  computes  $SK = h(n_1 \parallel SID_i \parallel (g^{xy} \bmod p) \parallel n_2)$  and  $V_4^* = h(SK \parallel (n_2 + 1))$ . Next,  $S$  check if  $V_4^* \neq V_4$ . If it holds,  $U_i$  is successfully authenticated. Otherwise, the connection is terminated.

In their authentication phase, we see that only  $S$  generates random value  $n_2$  to challenge  $U_i$ , while  $U_i$ 's  $n_1$  is opened in a common channel. This design will limit random value's power in scheme. Furthermore, user's identity can be leaked because their scheme distributes  $g^{xy} \bmod p$  to all users. We will analyze in next section.

**Password Change Phase** When  $U_i$  wants to change his/her  $pw_i$ ,  $U_i$  can perform following steps.

- $U_i$  inserts  $SC$  into another terminal device, and enters  $id_i, pw_i$ .
- $SC$  computes  $Q_1 = h(id_i \parallel R_2)$  and  $Q_1^* = R_3 \oplus h(pw_i)$  and compares with each other. If  $Q_1 = Q_1^*$ ,  $SC$  goes to next step; otherwise, the procedure is terminated.
- $SC$  computes  $R_3' = h(id_i \parallel R_2) \oplus h(pw_i) \oplus h(pw_i) \oplus h(pw_i')$  and replace  $R_3$  with  $R_3'$ .

In their password change phase, we see that password update is performed without interacting with  $S$ . In our scheme, we will inherit this idea from [15].

**Lost Card Revocation Phase** When  $U_i$  discovers  $SC$ 's information is leaked,  $U_i$  can request  $S$  to revoke  $SC$  via a secure channel. When receiving revocation request,  $S$  validates  $U_i$  by checking  $U_i$ 's secret personal information. After successfully validation,  $S$  saves  $sn_i$  of revoked  $SC$  in the database and issue a new  $SC$  with new  $sn_i'$  for  $U_i$ . Finally,  $U_i$  chooses a new  $pw_i$  similarly to the steps in registration phase.

## 2.2 Cryptanalysis of Chang et al.'s Scheme

In this subsection, we present our results on Chang et al.'s scheme. We demonstrate that their scheme is vulnerable to impersonation and session-key stolen attacks. Besides, their scheme does not provide user's anonymity.

**Inability to Protect User Anonymity** In Chang et al.'s scheme, we see that another user sends  $m_1 = \{DID_i, V_1, n_1\}$ . However, important information ( $g^{xy} \bmod p$ ) is distributed to all valid users. Hence, anyone who is legitimate user can steal other users' identity by performing following steps:

- Malicious user captures  $m_1 = \{DID_i, V_1, n_1\}$
- Next, he/she obtains  $SID_i = DID_i \oplus h((g^{xy} \bmod p) \parallel n_1)$
- Finally, he/she extracts  $id_i$  and  $sn_i$  from  $SID_i$  and knows who is authenticating with  $S$ .

Clearly, their scheme does not defend user's anonymity against attackers.

**Impersonation Attack** Because of inappropriate design, Chang et al.'s scheme is vulnerable to server and user impersonation attack. First of all, we present the steps which another malicious user employs to masquerade as the server:

- Similarly to above steps, malicious user obtains another user's  $id_i$  and  $sn_i$ .
- With  $id_i$ , he/she computes  $R_1^* = V_1 \oplus (id_i \parallel (g^{xy} \bmod p))$  and  $V_2^* = h(R_1^* \parallel id_S \parallel n_1)$ , which  $V_1$  and  $n_1$  belongs to  $m_1 = \{DID_i, V_1, n_1\}$  which is captured by him/her.
- Next, he/she generates a random value  $n_2^*$  and computes  $V_3^* = h(h(id_i \parallel (g^{xy} \bmod p)) \parallel n_1) \oplus n_2^*$ .
- Finally, he/she sends  $m_2^* = \{id_S, V_2^*, V_3^*\}$  to user.

Upon receiving  $m_2^*$ ,  $U_i$  re-computes  $V_2 = h(R_1 \parallel id_S \parallel n_1)$  and compares it with  $V_2^*$ . Clearly, they are equal and malicious user successfully impersonates  $S$ . Furthermore, he can impersonate another  $U_i$  authenticating with  $S$ . Following are some steps to masquerade as legitimate user.

- Malicious user captures  $m_1 = \{DID_i, V_1, n_1\}$ , he/she extracts  $SID_i$  by computing  $SID_i = DID_i \oplus h(R_2 \parallel n_1)$ , where  $R_2$  is his/her smartcard's information.
- Afterwards, he/she generates a random value  $n_1^*$  and re-computes  $DID_i^* = h(R_2 \parallel n_1^*) \oplus SID_i$ .
- Next, he/she sends  $m_1^* = \{DID_i^*, V_1, n_1\}$  to  $S$ .
- After receiving  $m_1^*$ ,  $S$  computes and re-sends  $m_2 = \{id_S, V_2, V_3\}$  to him/her. In this time, he/she computes  $n_2 = V_3 \oplus h(h(id_i \parallel R_2) \parallel n_1^*)$ , where  $id_i$  is obtained by him/her.
- With  $n_2$ , he/she computes  $SK^* = h(n_1^* \parallel SID_i \parallel R_2 \parallel n_2)$  and  $V_4^* = h(SK^* \parallel (n_2 + 1))$ .
- Finally, he/she sends  $m_3^* = \{V_4^*\}$  to  $S$ .

After receiving  $m_3^*$ ,  $S$  computes  $V_4$  and compares it with  $V_4^*$ . Clearly, they are equal and malicious user successfully impersonate another legitimate  $U_i$ .

**Session Key Attack** Another malicious user can observe outside and compute common session-key  $SK$  by performing following steps:

- First of all, he captures three packages  $m_1$ ,  $m_2$  and  $m_3$  in common channel.
- Next, he computes  $SID_i = DID_i \oplus h((g^{xy} \bmod p) \parallel n_1)$  and extracts  $id_i$ .
- Afterwards, he obtains  $n_2$  by performing  $n_2 = V_3 \oplus h((id_i \parallel R_2) \parallel n_1)$ .
- Finally, he computes  $SK = h(n_1 \parallel SID_i \parallel R_2 \parallel n_2)$ .

Clearly, all data encrypted with session-key will be revealed.

### 3 Proposed Scheme

At first, we depict Chebyshev polynomial [17] which is our scheme's security foundation. Chebyshev polynomial has the form:  $T_n(x) = \cos(n * \arccos(x))$ , where  $n$  is an integer degree and  $x \in [-1, 1]$ . Besides, we have its recurrent formulas:

- $T_0(x) = 1$
- $T_1(x) = x$
- ...
- $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$  and  $n \geq 2$ .

Moreover, our scheme utilizes polynomial's semi-group property:  $T_q(T_w(x)) = \cos(q * \arccos(\cos(w * \arccos(x)))) = \cos(qw * \arccos(x)) = \cos(wq * \arccos(x)) = \cos(w * \arccos(\cos(q * \arccos(x)))) = T_w(T_q(x))$ .

Next, we propose an improved scheme that eliminates aforementioned security problems. Before presenting each phase, we present general ideas in our scheme. In registration phase, our main objective includes providing authentication key  $h(X_S \parallel e)$  and storing  $h(id_i) \oplus X_S$  in server's database to check identity's validity. Especially, random value  $e$  helps to create different keys at different time. In login and authentication phases, we use two random values  $R_U$  and  $R_S$  combined with Chebyshev polynomials for challenge. In addition, we employ three-way challenge-response handshake technique to better resist replay and impersonation attacks [9]. Eventually, it is essential to obtain  $SK$  for encrypting data transmitted between user and server after successfully authentication phase. Our scheme is also divided into five phases of registration, login, mutual authentication, password update and lost card revocation.

#### 3.1 Registration Phase

Before presenting this phase, we suggest three conditions which registration phase should satisfy: Firstly, user's password should be concealed from the server. In our scheme, although the server generates user's password, the user will change his/her password after receiving it from the server. Secondly, the server must provide different authentication keys at different time. By using random value, our scheme completely achieves this requirement. Thirdly, the server should store

user's identity for later checking in next phases such as login or authentication phase. Our scheme is designed to achieve these fundamentals.

When  $U_i$  registers to  $S$ , he/she must submit his/her chosen  $id_i$  via a secure channel. When receiving this information,  $S$  performs following steps:

1.  $S$  generates  $pw_i$  and random value  $e$ .
2. Next,  $S$  computes authentication key  $K = h(X_S \parallel e)$ , masked key  $M = K \oplus h(id_i \parallel pw_i)$  and confirmation  $L = h(K \parallel id_i \parallel pw_i)$ , where  $X_S$  is  $S$ 's master key.
3. Afterwards,  $S$  stores  $h(id_i) \oplus X_S$  in  $S$ 's database for later checking.
4. Finally,  $S$  sends  $\{pw_i, SC(M, L, e, h(\cdot), T_s(x))\}$  via a secure channel, where  $\{x, T_s(x)\}$  is  $S$ 's public information.

After receiving  $SC$  and  $pw_i$ ,  $U_i$  updates  $pw_i$  via our password-update phase.

### 3.2 Login Phase

In login phase, checking user's identity and password must be performed at client side to prevent the attackers from overwhelming the server with a false identity and password in order to busy the server for a long time. Besides, login-message should be dynamic at different time to protect user's information especially identity. Our login phase is also designed to satisfy these requirements.

When  $U_i$  inputs his/her  $id_i$  and  $pw_i$  to login  $S$ , then  $SC$  performs:

1.  $SC$  computes  $K = M \oplus h(id_i \parallel pw_i)$  and  $L^* = h(K \parallel id_i \parallel pw_i)$ .
2. Next, it compares  $L^*$  with  $L$ . If they are the same,  $id_i$  and  $pw_i$  are correct and  $SC$  goes to next steps; otherwise, it terminates the session.
3. Afterwards,  $SC$  generates a random large integer  $r_U$ , computes  $R_U = T_{r_U}(x)$ ,  $DID_i = id_i \oplus h(R_U \parallel K)$  and  $R_2 = h(K \parallel id_i \parallel R_U)$ .
4. Finally, it sends  $\{e, R_U, R_2, DID_i\}$  to  $S$  via common channel.

### 3.3 Authentication and Session Key Agreement Phase

In authentication phase, both user and server must challenge each other to prove their legitimacy. Additionally, they should obtain common session-key after successful authentication. Our phase has these two important features.

In this session, after receiving  $U_i$ 's  $\{e, R_U, R_2, DID_i\}$  in login phase.  $S$  performs the steps to authenticate  $U_i$ .

1.  $S$  computes  $h(X_S \parallel e)$  and extracts  $id_i = DID_i \oplus h(R_U \parallel h(X_S \parallel e))$ .
2. Next,  $S$  check  $id_i$  by performing  $h(id_i) \oplus X_S$ , and searches its existence in  $S$ 's database. If it exists,  $id_i$  is valid; otherwise,  $S$  terminates the session.
3. Afterwards,  $S$  computes  $R_2^* = h(h(X_S \parallel e) \parallel id_i \parallel R_U)$  and compares  $R_2^*$  with  $R_2$ . If they are the same,  $S$  goes to next step; otherwise,  $S$  terminates the session.
4.  $S$  generates  $r_S$ , computes  $R_S = T_{r_S}(x)$ ,  $SK = h(T_{r_S}(R_U) \parallel h(X_S \parallel e) \parallel id_i)$  and  $R_4 = h(h(X_S \parallel e) \parallel id_i \parallel SK)$ .



5. Finally,  $S$  sends  $\{R_S, R_4\}$  to  $U_i$  via common channel.
6. After receiving  $S$ 's  $\{R_S, R_4\}$ ,  $U_i$  re-computes  $SK = h(T_{r_U}(R_S) \parallel K \parallel id_i)$ ,  $R_4^* = h(K \parallel id_i \parallel SK)$  and compares  $R_4^*$  with  $R_4$ . If they are the same,  $U_i$  successfully authenticates  $S$ .
7.  $U_i$  computes  $R_5 = h(SK)$  and sends to  $S$  via common channel.
8. After receiving  $U_i$ 's  $\{R_5\}$ ,  $S$  re-computes  $R_5^* = h(SK)$  and compares it with  $R_5$ . If they are the same,  $S$  successfully authenticates  $U_i$ .

### 3.4 Password Update Phase

When  $U_i$  wants to change  $pw_i$ ,  $U_i$  performs:

1.  $U_i$  inserts  $SC$  and inputs  $id_i$  and  $pw_i$ .
2. Next,  $SC$  computes  $K = M \oplus h(id_i \parallel pw_i)$  and  $L^* = h(K \parallel id_i \parallel pw_i)$ .
3. Afterwards,  $SC$  compares  $L^*$  with  $L$  stored in it. If they are the same,  $SC$  accepts user's request; otherwise, it terminates the session.
4.  $U_i$  inserts new password  $pw_{inew}$ . Then,  $SC$  computes  $M_{new} = K \oplus h(id_i \parallel pw_{inew})$  and  $L_{new} = h(K \parallel id_i \parallel pw_{inew})$ .
5. Finally,  $SC$  replaces  $L, M$  with  $L_{new}, M_{new}$ .

### 3.5 Lost Card Revocation Phase

If  $U_i$  loses his/her  $SC$ ,  $U_i$  must notify  $S$ . Then,  $S$  will re-issue new  $SC$  with the old  $U_i$ 's  $id_i$ .

1.  $U_i$  re-submits  $id_i$  and *request-re-issue-smart-card* to  $S$  via a secure channel.
2. After receiving  $U_i$ 's request,  $S$  computes  $h(id_i) \oplus X_S$  and searches its existence in  $S$ 's database. If it exists,  $S$  accepts  $U_i$ 's request; otherwise,  $S$  terminates the session.
3. Next,  $S$  generates a new random value  $e_{new}$  and performs steps which are the same as registration phase's. Finally,  $S$  re-issues new  $SC$  to  $U_i$  via a secure channel.

## 4 Security and Efficiency Analysis

In this section, we analyze our scheme on two aspects: security and efficiency. Before further analysis, we introduce three basic computational assumptions which proposed scheme employs, that are one-way hash function ([15] for more details), Chebysev discrete logarithm problem (**CDLP**) and Diffie-Hellman problem (**CDHP**)([18, 11] for more details).

- Chebysev Discrete Logarithm Problem: Given  $x \in [-1, 1]$ ,  $T_n(x)$ , where  $n, s \in \mathbf{N}$ , the discrete logarithm problem is to find unknown degree  $n$ .
- Chebysev Diffie-Hellman Problem: Given  $x \in [-1, 1]$ ,  $T_q(x)$  and  $T_s(x)$ , where  $q, s \in \mathbf{N}$ , the computational Diffie-Hellman problem is to find  $T_{q*s}(x)$  or  $T_{s*q}(x)$ , where  $T_{q*s}(x) = T_q(T_s) = T_s(T_q) = T_{s*q}(x) \in [-1, 1]$ .

#### 4.1 Correctness Proof

To correct evaluate about authentication scheme, we employ BAN-logic [16] proposed by Burrows. We introduce some basic symbols used in this method as follows: symbols  $P$  and  $Q$  stand for principals,  $X$  and  $Y$  range over statements, and  $K$  represent the cryptographic key. For more details about the notations and postulates, please refer to Burrows' result. In the following, we use BAN-logic to prove proposed scheme achieves correct mutual authentication and session key agreement. In stead of using  $P$ ,  $Q$ , we let  $U_i$ ,  $S$  stand for user and server participating in the scheme. Furthermore, we formalize our goals denoted as  $\mathbf{G}_j$ , where  $j \in [1, 8]$  as follows:

1.  $U_i \mid\equiv U_i \stackrel{id_i}{\leftrightarrow} S$
2.  $U_i \mid\equiv S \mid\equiv U_i \stackrel{id_i}{\leftrightarrow} S$
3.  $S \mid\equiv U_i \stackrel{id_i}{\leftrightarrow} S$
4.  $S \mid\equiv U_i \mid\equiv U_i \stackrel{id_i}{\leftrightarrow} S$
5.  $U_i \mid\equiv U_i \stackrel{SK}{\leftrightarrow} S$
6.  $U_i \mid\equiv S \mid\equiv S \stackrel{SK}{\leftrightarrow} U_i$
7.  $S \mid\equiv S \stackrel{SK}{\leftrightarrow} U_i$
8.  $S \mid\equiv U_i \mid\equiv U_i \stackrel{SK}{\leftrightarrow} S$

Then, we idealize proposed scheme as follows:

- $\mathbf{DID}_i = \langle U_i \stackrel{id_i}{\leftrightarrow} S, T_{r_U}, U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S \rangle$
- $\mathbf{R}_2 = \langle U_i \stackrel{id_i}{\leftrightarrow} S, U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S, T_{r_U} \rangle$
- $\mathbf{R}_4 = \langle U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S, U_i \stackrel{id_i}{\leftrightarrow} S, U_i \stackrel{SK}{\leftrightarrow} S \rangle$
- $\mathbf{R}_5 = \langle T_{r_S}(R_U), U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S, U_i \stackrel{id_i}{\leftrightarrow} S \rangle$

Next, we give some assumptions (denoted as  $\mathbf{A}_t$ , where  $t \in [1, 8]$ ) about proposed scheme's initial states

1.  $U_i \mid\equiv U_i \stackrel{id_i}{\leftrightarrow} S$
2.  $U_i \mid\equiv U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S$
3.  $U_i \mid\equiv S \Rightarrow U_i \stackrel{SK}{\leftrightarrow} S$
4.  $S \mid\equiv U_i \Rightarrow U_i \stackrel{id_i}{\leftrightarrow} S$
5.  $S \mid\equiv U_i \Rightarrow U_i \stackrel{SK}{\leftrightarrow} S$
6.  $S \mid\equiv S \stackrel{h(X_S \parallel e)}{\leftrightarrow} U_i$
7.  $U_i \mid\equiv \#(T_{r_S})$
8.  $S \mid\equiv \#(T_{r_U})$

Finally, with  $\mathbf{A}_t$  and BAN-logic's postulates, we demonstrate our scheme successfully achieves  $\mathbf{G}_j$ .

- $U_i$  registers  $id_i$  with  $S$ , so we achieve  $\mathbf{G}_1$

$$U_i \mid \equiv U_i \stackrel{id_i}{\leftrightarrow} S$$

- With **A**<sub>6</sub> and *DID*<sub>i</sub>, applying the message-meaning rule to derive

$$\frac{S \mid \equiv S \stackrel{h(X_S \parallel e)}{\leftrightarrow} U_i, S \triangleleft (U_i \stackrel{id_i}{\leftrightarrow} S, T_{r_U}, U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S)_{h(X_S \parallel e)}}{S \mid \equiv U_i \mid \sim U_i \stackrel{id_i}{\leftrightarrow} S, T_{r_U}, U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S} \quad (1)$$

- With **A**<sub>8</sub> and applying freshness rule to infer

$$\frac{S \mid \equiv \#(T_{r_U})}{S \mid \equiv \#(U_i \stackrel{id_i}{\leftrightarrow} S, T_{r_U}, U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S)} \quad (2)$$

- With (1) and (2), applying the nonce - verification rule to derive

$$\frac{(1), (2)}{S \mid \equiv U_i \mid \equiv U_i \stackrel{id_i}{\leftrightarrow} S, T_{r_U}, U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S} \quad (3)$$

- With (3), applying believe rule to derive

$$\frac{(3)}{S \mid \equiv U_i \mid \equiv U_i \stackrel{id_i}{\leftrightarrow} S} \quad (\mathbf{G}_4)$$

- With **G**<sub>4</sub> and **A**<sub>4</sub>, applying jurisdiction rule to infer

$$\frac{S \mid \equiv U_i \Rightarrow U_i \stackrel{id_i}{\leftrightarrow} S, S \mid \equiv U_i \mid \equiv U_i \stackrel{id_i}{\leftrightarrow} S}{S \mid \equiv U_i \stackrel{id_i}{\leftrightarrow} S} \quad (\mathbf{G}_3)$$

- With **A**<sub>2</sub> and *R*<sub>4</sub>, applying the message-meaning rule to derive

$$\frac{U_i \mid \equiv U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S, U_i \triangleleft (U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S, U_i \stackrel{id_i}{\leftrightarrow} S, U_i \stackrel{SK}{\leftrightarrow} S)_{h(X_S \parallel e)}}{U_i \mid \equiv S \mid \sim U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S, U_i \stackrel{id_i}{\leftrightarrow} S, U_i \stackrel{SK}{\leftrightarrow} S} \quad (4)$$

- With (4) and **A**<sub>7</sub>, applying the freshness rule to derive

$$\frac{(4), U_i \mid \equiv \#(T_{r_S})}{U_i \mid \equiv \#(U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S, U_i \stackrel{id_i}{\leftrightarrow} S, U_i \stackrel{SK}{\leftrightarrow} S)} \quad (5)$$

- With (4) and (5), applying the nonce - verification rule to derive

$$\frac{(4), (5)}{U_i \mid \equiv S \mid \equiv U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S, U_i \stackrel{id_i}{\leftrightarrow} S, U_i \stackrel{SK}{\leftrightarrow} S} \quad (6)$$

- With (6), applying the believe rule to derive

$$\frac{(6)}{U_i \mid \equiv S \mid \equiv U_i \stackrel{id_i}{\leftrightarrow} S} \quad (\mathbf{G}_2)$$

With **G**<sub>1</sub>, **G**<sub>2</sub>, **G**<sub>3</sub>, and **G**<sub>4</sub>, we prove *U*<sub>i</sub> and *S* can mutually authenticate with dynamic identity. Next, we demonstrate *U*<sub>i</sub> and *S* can share *SK* as follows.

- With *R*<sub>4</sub> and **A**<sub>2</sub>, applying the message-meaning rule to derive

$$\frac{U_i \mid \equiv U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S, U_i \triangleleft (U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S, U_i \stackrel{id_i}{\leftrightarrow} S, U_i \stackrel{SK}{\leftrightarrow} S)_{h(X_S \parallel e)}}{U_i \mid \equiv S \mid \sim U_i \stackrel{h(X_S \parallel e)}{\leftrightarrow} S, U_i \stackrel{id_i}{\leftrightarrow} S, U_i \stackrel{SK}{\leftrightarrow} S} \quad (7)$$

- With *R*<sub>4</sub> and **A**<sub>7</sub>, applying the freshness rule to derive

$$\frac{U_i | \equiv \#(T_{r_S})}{U_i | \equiv \#(U_i \stackrel{h(X_S \| e)}{\leftrightarrow} S, U_i \stackrel{id_i}{\leftrightarrow} S, U_i \stackrel{SK}{\leftrightarrow} S)} \quad (8)$$

– With (7) and (8), applying the nonce - verification rule to derive

$$\frac{(7), (8)}{U_i | \equiv S | \equiv U_i \stackrel{h(X_S \| e)}{\leftrightarrow} S, U_i \stackrel{id_i}{\leftrightarrow} S, U_i \stackrel{SK}{\leftrightarrow} S} \quad (9)$$

– With (9), applying the believe rule to derive

$$\frac{(9)}{U_i | \equiv S | \equiv S \stackrel{SK}{\leftrightarrow} U_i} \quad (G_6)$$

– With **A**<sub>3</sub> and **G**<sub>6</sub>, we apply the jurisdiction rule to infer

$$\frac{U_i | \equiv S \Rightarrow U_i \stackrel{SK}{\leftrightarrow} S, U_i | \equiv S | \equiv U_i \stackrel{SK}{\leftrightarrow} S}{U_i | \equiv U_i \stackrel{SK}{\leftrightarrow} S} \quad (G_5)$$

– With **R**<sub>5</sub> and **A**<sub>6</sub>, applying the message-meaning rule to derive

$$\frac{S | \equiv S \stackrel{h(X_S \| e)}{\leftrightarrow} U_i, S \triangleleft (T_{r_U * r_S}, U_i \stackrel{h(X_S \| e)}{\leftrightarrow} S, U_i \stackrel{id_i}{\leftrightarrow} S)_{h(X_S \| e)}}{S | \equiv U_i | \sim T_{r_U * r_S}, U_i \stackrel{h(X_S \| e)}{\leftrightarrow} S, U_i \stackrel{id_i}{\leftrightarrow} S} \quad (10)$$

– With **R**<sub>5</sub> and **A**<sub>8</sub>, applying the freshness rule to derive

$$\frac{S | \equiv \#(T_{r_U})}{S | \equiv \#(T_{r_U * r_S}, U_i \stackrel{h(X_S \| e)}{\leftrightarrow} S, U_i \stackrel{id_i}{\leftrightarrow} S)} \quad (11)$$

– With (10) and (11), applying the nonce - verification rule to derive

$$\frac{(10), (11)}{S | \equiv U_i | \equiv T_{r_U * r_S}, U_i \stackrel{h(X_S \| e)}{\leftrightarrow} S, U_i \stackrel{id_i}{\leftrightarrow} S} \quad (12)$$

– With (12) and **A**<sub>6</sub>, applying the believe rule to infer

$$\frac{(12), S | \equiv S \stackrel{h(X_S \| e)}{\leftrightarrow} U_i}{S | \equiv U_i | \equiv U_i \stackrel{SK}{\leftrightarrow} S} \quad (G_8)$$

– With (12) and **A**<sub>5</sub>, applying the message-meaning rule to infer

$$\frac{(12), S | \equiv U_i \Rightarrow U_i \stackrel{SK}{\leftrightarrow} S}{S | \equiv T_{r_U * r_S}, U_i \stackrel{h(X_S \| e)}{\leftrightarrow} S, U_i \stackrel{id_i}{\leftrightarrow} S} \quad (13)$$

– With (13), applying the believe rule to derive

$$\frac{(13)}{S | \equiv S \stackrel{SK}{\leftrightarrow} U_i} \quad (G_7)$$

With **G**<sub>5</sub>, **G**<sub>6</sub>, **G**<sub>7</sub> and **G**<sub>8</sub>, we prove both  $S$  and  $U_i$  believe the other believes  $SK$  shared between  $U_i$  and  $S$ . Below are common kinds of attacks proposed scheme can withstand.

## 4.2 Resistance to Common Attacks

In this subsection, we prove our scheme can withstand many common kinds of attacks based on above two basic assumptions. Our context is that both server and user are authenticating in open channel. Hence, anyone is capable of intercepting all messages transmitted between them. Besides, we assume anyone can obtain  $SC$ 's information.

**Replay Attack** In this kind of attack, adversary captures the user's old messages for next transaction. It is hard to perform in proposed scheme. For example, when adversary sends package  $\{e, R_U, R_2, DID_i\}$  at another session to cheat the server, he/she needs to resend  $R_5$  at the end of the session. Clearly, knowing  $U_i$ 's  $r_U, id_i$  and  $h(X_S \parallel e)$  is impossible to adversary. It is said that proposed scheme can withstand replay attack.

**User And Server Impersonation Attack** In this kind of attack, adversary has two options, which are user and server impersonation. Firstly, we consider the case of user impersonation. In the users login message, only two messages that adversary can forge are  $e$  and  $R_U = T_{r_U}(x)$  because they do not include identity information. Consequently, adversary randomly chooses  $r_U^*$  to compute  $R_U^* = T_{r_U^*}(x)$ , where  $e^*$  is adversary's own random value. Finally, he/she sends  $\{e^*, R_U^*, R_2, DID_i\}$  to server. When receiving, server computes  $h(X_S \parallel e^*)$  and extracts  $id_i$  by computing  $DID_i \oplus h(R_U^* \parallel h(X_S \parallel e^*)) = id_i \oplus h(R_U \parallel h(X_S \parallel e)) \oplus h(R_U^* \parallel h(X_S \parallel e^*))$ . Clearly, we see the result of this computation is nonsense. Therefore, server will detect and terminate this session. Secondly, we consider the case of server impersonation. We see that adversary needs to successfully compute  $\{R_S, R_4\}$  and this is impossible because  $R_4 = h(h(X_S \parallel e) \parallel id_i \parallel SK^*)$ , where  $SK^*$  is random session key computed from adversary's random value  $r_S^*$ . Hence, adversary needs  $U_i$ 's  $h(X_S \parallel e)$  and  $id_i$ . In short, proposed scheme can resist two-side impersonation attack.

**User Anonymity Protected** In this kind of attack, adversary wants to know whose transaction this is. Therefore, he/she will find the way to extract identity from the message  $DID_i$ . We see that user's identity is combined with random value  $R_U$  and key  $K = h(X_S \parallel e)$ . With two values, adversary has no chance to extract true identity. Specially,  $DID_i$  is different at each session due to random value  $R_U$ . Also, adversary does not know whether or not  $DID_i$  and  $DID_i'$  belong to the same person. Hence, proposed scheme achieves strong user anonymity.

**Perfect forward secrecy (PFS)** In this kind of attack, assume that long-term key of the server and all users is leaked, so the system is broken. However, the previously transactions should be secured from the adversary and this means that generated session keys should be secured. In proposed scheme, in case of leakage of server  $S$ 's  $X_S$  and user  $U_i$ 's  $h(X_S \parallel e_i)$ , the adversary has  $R_U = T_{r_U}(x)$ ,  $R_S = T_{r_S}(x)$  and  $id_i$ . Nevertheless, computing  $T_{r_S} * r_U(x)$  is the same as computing the **CDHP**. It is said that proposed scheme can achieve **PFS** based on **CDHP**.

Chang et al.'s ideas are inherited by proposed scheme. For example, no using password or state table due to the increase of computational overload, or using random value instead of time-stamp to save time-synchronization mechanism cost. Likewise, using cryptographic hash function allows the users to freely choose their password without worrying about bit-length. In short, those properties

**Table 1.** The comparison between our scheme and previous ones for security

Items	Das's[19]	Wang's[20]	Chang's[15]	Ours
Mutual authentication	No	Yes	Yes	Yes
Password chosen by users	Yes	No	Yes	Yes
User anonymity	Yes	No	No	Yes
Without registration table	Yes	Yes	Yes	Yes
Withstand impersonation attack	No	No	No	Yes
Without time-synchronized mechanism	No	No	Yes	Yes
Session key establishment	No	No	Yes	Yes
Perfect forward secrecy	No*	No*	No	Yes
* Do not provide session key establishment				

completely exist in proposed scheme. Table 1 is the comparison between our scheme and previous schemes including Chang et al.'s for security.

### 4.3 Efficiency Analysis

To compare efficiency between our scheme and previous ones, we let  $H$  be the hash operation,  $\uparrow$  be modular exponentiation operation,  $\oplus$  be exclusive-or operation and  $T$  be computational operation of polynomial. At registration phase, Das's scheme needs  $1 \times \oplus, 2 \times H$ ; Wang's needs  $2 \times \oplus, 2 \times H$ ; Chang's needs  $1 \times \oplus, 3 \times H, 1 \times \uparrow$ ; Ours needs  $2 \times \oplus, 4 \times H$ . At login and authentication phases, Das's scheme needs  $14 \times \oplus, 7 \times H$ ; Wang's needs  $14 \times \oplus, 6 \times H$ ; Chang's needs  $7 \times \oplus, 10 \times H, 1 \times \uparrow$ ; Ours needs  $4 \times \oplus, 14 \times H, 4 \times T$ . Compared with previous schemes, our scheme's computational cost increases perceptibly. However, this is essential because of enhancement of security. Furthermore, in according to [14], we believe if practical implemented, our scheme will be still efficient enough. The theoretical comparison of cost at this phase is presented in Table 2.

Let  $t_H, t_{\oplus}, t_T, t_{\uparrow}$  denote running-time corresponding to each operation  $H, \oplus, T, \uparrow$ . We see that  $t_{\oplus} \ll t_H \ll t_{\uparrow} < t_T$ , so we only compare between two algorithms, modular exponentiation and Chebysev polynomial which are used in Chang's scheme and ours. To relatively compare, we re-implement  $T_n(g) \bmod p$  using BigInteger class in Java. Also, we re-use 'ModPow' function in Java to stand for  $g^n \bmod p$ . Our experiment is conducted in personal computer, Intel Core 2 Quad CPU 2.66GHz. By measuring running-time between two algorithms with prime numbers which range from 10 to 400 digits, we propose using 512-bit

**Table 2.** A comparison of computation costs

Items	Authentication	Login	Registration
Das[19]	$3 \times H, 7 \times \oplus$	$4 \times H, 7 \times \oplus$	$2 \times H, 1 \times \oplus$
Wang[20]	$4 \times H, 10 \times \oplus$	$2 \times H, 4 \times \oplus$	$2 \times H, 2 \times \oplus$
Chang[15]	$8 \times H, 4 \times \oplus, 1 \times \uparrow$	$2 \times H, 3 \times \oplus$	$3 \times H, 1 \times \oplus, 1 \times \uparrow$
Ours	$10 \times H, 2 \times \oplus, 3 \times T$	$4 \times H, 2 \times \oplus, 1 \times T$	$4 \times H, 2 \times \oplus$

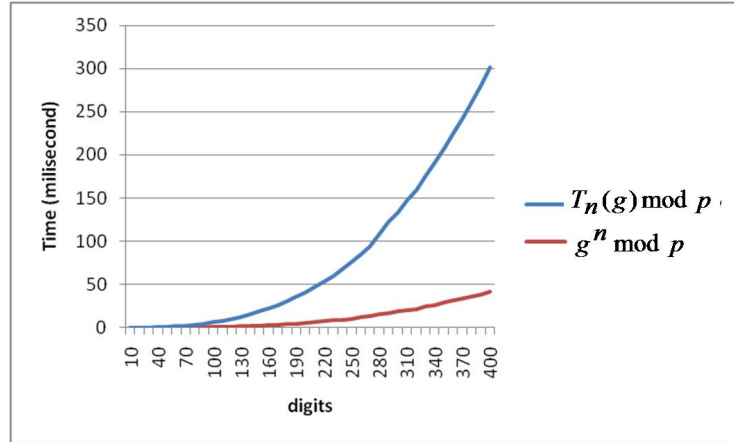


Fig. 1. Comparison of time cost between  $T_n(g) \bmod p$  and  $g^n \bmod p$

prime number to guarantee time efficiency ( $\approx 150\text{ms}$ ) and security because of solution space up to  $2^{512}$  when facing **CDLP**. Although running-time between  $g^n \bmod p$  used by Chang's scheme and  $T_n(g) \bmod p$  used by ours is a little different, practical running-time of our scheme  $\sum_{i=1}^4 t_T \approx 0.6\text{s}$  when using prime number with appropriate bit amount. Therefore, it is said that our scheme is still enough efficiency when practically implemented. Experiment's result with different prime numbers is presented in Figure 1.

## 5 Conclusions

In this paper, we review Chang et al.'s scheme. Although their scheme has some positive characteristics but it is vulnerable to impersonation attack. Furthermore, it cannot provide user's anonymity and does not have the property of perfect forward secrecy. Hence, we suggest a different improved scheme using Chebyshev polynomial to overcome such pitfalls. Compared with Chang's scheme schemes, our scheme has the following main advantages; (1) A user need not choose the password at first. (2) It provides user's anonymity. (3) It does not maintain verification table. (4) It provides property of perfect forward secrecy.

From our security evaluation, our proposed method can resist known methods of attacks. As the proposed scheme can be used in various client-server environment for remote user authentication, it can be applied for systems that accept user authentication with mobile or wearable devices to create smart interactive environments. Furthermore we also study to integrate biometric features into Chebyshev polynomial-based authentication scheme.

**Acknowledgment** This research is supported by National Institute of Informatics (Japan) and Vietnam National University-Ho Chi Minh city (Vietnam) to develop new schemes for user authentication with mobile and wearable devices.

## Bibliography

- [1] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.
- [2] T. H. Chen, Y. C. Chen, W. K. Shih, and H. W. Wei, "An efficient anonymous authentication protocol for mobile pay-tv," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1131–1137, 2011.
- [3] S. Shin, K. Kim, K.-H. Kim, and H. Yeh, "A remote user authentication scheme with anonymity for mobile devices," *International Journal of Advanced Robotic Systems*, vol. 9, no. 13, pp. 1–7, 2012.
- [4] I. E. Liao, C. C. Lee, and M. S. Hwang, "Security enhancement for a dynamic id-based remote user authentication scheme," *International Conference on Next Generation Web Services Practices*, vol. 6, no. 2, pp. 517–522, 2005.
- [5] E. J. Yoon and K. Y. Yoo, "Improving the dynamic id-based remote mutual authentication scheme," *First International Workshop on Information Security*, vol. 4277, pp. 499–507, 2006.
- [6] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
- [7] J.-H. Yang and C.-C. Chang, "An id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *Computers and Security*, vol. 28, no. 3-4, pp. 138–143, 2009.
- [8] E.-J. Yoon and K.-Y. Yoo, "Robust id-based remote mutual authentication with key agreement scheme for mobile devices on ecc," *IEEE International Conference on Computational Science and Engineering*, vol. 2, pp. 633–640, 2009.
- [9] S. H. Islam and G. P. Biswas, "A more efficient and secure id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *Journal of Systems and Software*, vol. 84, no. 11, pp. 1892–1898, 2011.
- [10] 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 system," *Journal of Physics Letters A*, vol. 360, no. 2, pp. 259–262, 2006.
- [11] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Journal of Chaos, Solitons & Fractals*, vol. 37, no. 3, pp. 669–674, 2008.
- [12] C. Guo, C.-C. Chang, and C.-Y. Sun, "Chaotic maps-based mutual authentication and key agreement using smartcards for wireless communications," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 2, pp. 99–109, 2013.
- [13] K. Prasad, K. Ramar, and R. Gnanajeyaraman, "Public key cryptosystems based on chaotic chebyshev polynomials," *Journal of Engineering and Technology Research*, vol. 1, no. 7, pp. 122–128, 2009.



- [14] L. Zhi-hui, C. Yi-dong, and X. Hui-min, "Fast algorithms of public key cryptosystem based on chebyshev polynomials over finite field," *The Journal of China Universities of Posts and Telecommunications*, vol. 18, no. 2, pp. 86–93, 2010.
- [15] C.-C. Chang and C.-Y. Lee, "A smart card-based authentication scheme using user identify cryptography," *International Journal of Network Security*, vol. 15, no. 2, pp. 139–147, 2013.
- [16] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer System*, vol. 8, pp. 18–36, 1990.
- [17] D. Xiao, X. Liao, and K. Wong, "An efficient entire chaos-based scheme for deniable authentication," *Journal of Chaos, Solitons & Fractals*, vol. 23, no. 4, pp. 1327–1331, 2005.
- [18] P. Bergamo, P. Arco, A. Santis, and L. Kocarev, "Security of public key encryption technique based on multiple chaotic system," *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 7, pp. 1382–1393, 2005.
- [19] 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.
- [20] Y. Y. Wang, J. Y. Kiu, F. X. 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.