Hindawi
Security and Communication Networks
Volume 2019, Article ID 4250743, 11 pages
https://doi.org/10.1155/2019/4250743

Research Article

Improved Chebyshev Polynomials-Based Authentication Scheme in Client-Server Environment

Toan-Thinh Truong ,[1] Minh-Triet Tran,[1] and Anh-Duc Duong[2]

[1] *Faculty of Information Technology, University of Science, VNU-HCM, 700000, Vietnam*
[2] *University of Information Technology, VNU-HCM, 700000, Vietnam*

Correspondence should be addressed to Toan-Thinh Truong; ttthinh@fit.hcmus.edu.vn

Received 15 July 2018; Revised 10 December 2018; Accepted 20 December 2018; Published 17 January 2019

Guest Editor: Esther Palomar

With nonstop development of communication technologies, all aspects of social life continuously change and so do network systems. When establishing connection is easy, the convenience of online-service receives many users' attentions, for example, the patients directly access medical system to be advised by doctors at any time. Therefore, user authentication scheme is necessary when we want to provide privacy and security for working sessions. Storing a password list for verification is an old method and not secure. This list can be easily leaked, and adversary can launch an offline password-guessing attack. In addition, information exchanged between user and server needs being prevented from attacker's decryption. It can be said that current authentication schemes are unsuitable for new security standard. We need a strong user authentication scheme using new approach to overcome existing limitations and guarantee time efficiency. In this paper, we make a design with *Chebyshev* polynomial to achieve our goals and resist some kinds of attacks.

1. Introduction

User authentication is one of the first important parts in all remote services. Furthermore, after successful authentication, partners secretly exchange the messages to each other and we need a session key to encrypt all these messages. Therefore, authentication scheme also needs a session key agreement phase. Especially, when the wearable devices become popular, such as smart-glasses or smart-watch, a user wants to connect to remote service through these low-power computing devices. Therefore, in addition to security, also we consider the time efficiency which is one of the important factors. There are many proposed results using cryptography primitives to make a reasonable user authentication scheme. Lamport [1] is the pioneer using hash function with password. His method is a usage of password-table for user verification in login phase. This is a simple way and easily implemented, but his scheme is vulnerable to verification stolen attack, and inappropriately using password can result in offline password-guessing attack. Then, there are many proposed schemes to enhance security. Typically, in 2004 Das et al. [2] proposed dynamic identity to provide user anonymity in his scheme. This is a positive idea, but in his scheme, he uses password instead of real user's identity to create a dynamic login message. This causes their scheme cannot resist password-related attacks, and even the server may launch a password-guessing attack to find real user's password.

In 2006, Yoon et al. [3] proposed dynamic identity scheme using time-stamp. This scheme overcomes the reflection attack existing in Liao et al.'s scheme [4]. Clearly, Yoon's scheme has important improved ideas to isolate such problems. However, they also use password to authenticate with online server, so their scheme is still vulnerable to password-related attacks. Until now, password is still one of the most convenient factors in many authentication schemes, if only using this factor can be insecure. Using reasonable encryption scheme with block-cipher, such as *Advance Encryption Standard* (AES) or *Triple Data Encryption Standard* (T-DES), can enhance security for authentication scheme. Furthermore, if we only use hash function in scheme, this can increase authentication speed because time-cost of hash function is lower than the encryption scheme.

In addition to applying cryptography primitives, there is an approach using hard problems as security foundation, such as RSA or *Elliptic curve crypto-systems* (ECC). In 2009, Yang et al. [5] proposed a scheme in ECC. This is an efficient scheme because it uses *discrete logarithm* and *Diffie-Hellman* problems in *elliptic curve*. However, instead of using random values, they use point's coordinates to create a session key which does not satisfy *perfect forward session key secrecy* (PFS), one of the most standards to evaluate a strong authentication scheme. Therefore, some improved schemes were proposed, for instance, Islam et al. [6]. Their scheme used random values in creation of session key. However, his scheme is still vulnerable to known session-specific temporary information and denial of service attacks. In 2015 and 2016, Huang et al. [7] and Chaudhry et al. [8] proposed ECC-based authentication schemes, but these schemes cannot resist malicious user attack and does not provide PFS. Also, in 2015 Chaudhry et al. [9] proposed an authentication scheme in multiserver environment with general public key cryptography (RSA or ECC). However, their scheme needs a certificate agency (CA) to check the validity for the server's key pairs. Furthermore, all previous session keys will be recomputed if PFS appears. Compared with RSA, ECC can achieve the same security with a smaller key size. It can be said that ECC is one of the popular approaches many authors apply in authentication scheme because it offers better performance [10].

Recently, *Chebyshev* polynomial is an approach many authors pay attention to. Although this method's computational cost is more than ECC's and it is being researched to be a standard such as RSA or ECC. However, this is a new method, so there are so many papers applying it into their schemes. At first, authentication schemes use polynomial on real field to make a security foundation, but Bergamo [11] proposed a solution to break its security. In 2013, Hao et al. [12] proposed a scheme in telecare medicine information system using polynomial in real field, but Lee et al. [13] discovered that this scheme is vulnerable to violation of the contributory of key agreements. And Lee proposed a different improved scheme. However, we see that his scheme is still vulnerable to what Hao's scheme did. Also, there are some papers [14, 15] facing the same problem which Lee and Hao did. To enhance security for *Chebyshev* polynomial, Zhang [16] extended the polynomial's semigroup property to the interval (-∞, +∞). Since then, *Chebyshev* polynomial can be placed in modular prime number field and receives more consideration of security analysis [17]. In 2016, Irshad et al. [18] proposed an authentication scheme in multiserver with *Chebyshev*. This scheme is designed with three actors suitable for global mobility network (Glomonet). However, a partial of information about registration centre's master key (K_y) can be leaked. In their scheme, they have $PID_i \oplus K_y = (q_i \parallel ID \parallel PW)$. Clearly, the value and length of *ID* and *PW* is known, and any users easily guess by inspecting $PID_i = (x \parallel \textbf{\textit{easily\_guess}}) \oplus (q_i \parallel \textbf{\textit{ID}} \parallel \textbf{\textit{PW}})$. Although all information of K_y is not leaked, this is dangerous because user can collect many *PID* to find the "x" value. In 2017, Wang and Xu [19] proposed a reference model to solve the offline dictionary attacks. Their model is truly useful for designing many schemes with different approaches, such as RSA, ECC, or *Chebyshev*. It can be said that *Chebyshev* polynomial is a new approach which is being developed by many researchers and can be replaced for ECC or RSA in the future.

The rest of our paper is organized as follows. In Section 2, we present some background about *Chebyshev* polynomial. In Section 3, we review some previous typical schemes and analyse them on security aspect. Then in Section 4 we propose improved scheme in client-server environment using *Chebyshev* polynomial in modular prime number field. In Section 5, we analyse our proposed scheme on two aspects, namely, security and efficiency. Finally, the conclusion is presented in Section 6.

2. Preliminaries

This section describes some features of *Chebyshev* polynomial in real and modular prime number fields [20]. Also, we give some different proofs compared with [21, 22]. Following are *chaotic maps* and two hard problems.

2.1. Chebyshev Chaotic Maps. Let $n \in \mathbb{N}$ and $x \in [-1, 1]$; we define *Chebyshev* polynomial $T_n(x)$: $[-1, 1] \longrightarrow [-1, 1]$ as $T_n(x) = \cos(n \times \arccos(x))$. Its semigroup property is as follows:

$$\begin{aligned} T_a(T_b(x)) &= \cos(a \times \arccos(\cos(b \times \arccos(x)))) \\ &= \cos(a \times b \times \arccos(x)) \\ &= \cos(b \times a \times \arccos(x)) \\ &= \cos(b \times \arccos(\cos(a \times \arccos(x)))) \\ &= T_b(T_a(x)) \end{aligned} \quad (1)$$

In 2008, Zhang [16] extended (1) to the interval (-∞, +∞). Therefore, we have a different formula of *Chebyshev* polynomial as follows:

$$\begin{aligned} T_0(x) &= 1 \bmod p \\ T_1(x) &= x \bmod p \\ T_n(x) &= 2 \times x \times T_{n-1}(x) - T_{n-2}(x) \bmod p \end{aligned} \quad (2)$$

where $p \in \mathbb{P}$, $x \in [0, p-1]$ and $n \in \mathbb{N}$. We see that (2) can be changed to

$$T_n(x) \bmod p = \frac{\lambda_1^n + \lambda_2^n}{2} \bmod p \quad (3)$$

2.2. The Hard Problems. In addition to four important properties, we have two computational problems on *chaotic maps* we apply in proposed user authentication scheme.

(i) The first problem is *chaotic maps* based *discrete logarithm* (CMDLP): Given $y \in [0, p-1]$, $p \in \mathbb{P}$, and x, it is hard to find r value such that $T_r(x) = y \bmod p$. We call this *discrete logarithm* problem on *chaotic maps*.

(ii) The second problem is *chaotic maps* based *discrete logarithm* (CMDHP): Given $x \in [0, p-1]$, $p \in \mathbb{P}$, $T_a(x) \bmod p$, and $T_b(x) \bmod p$, it is hard to find

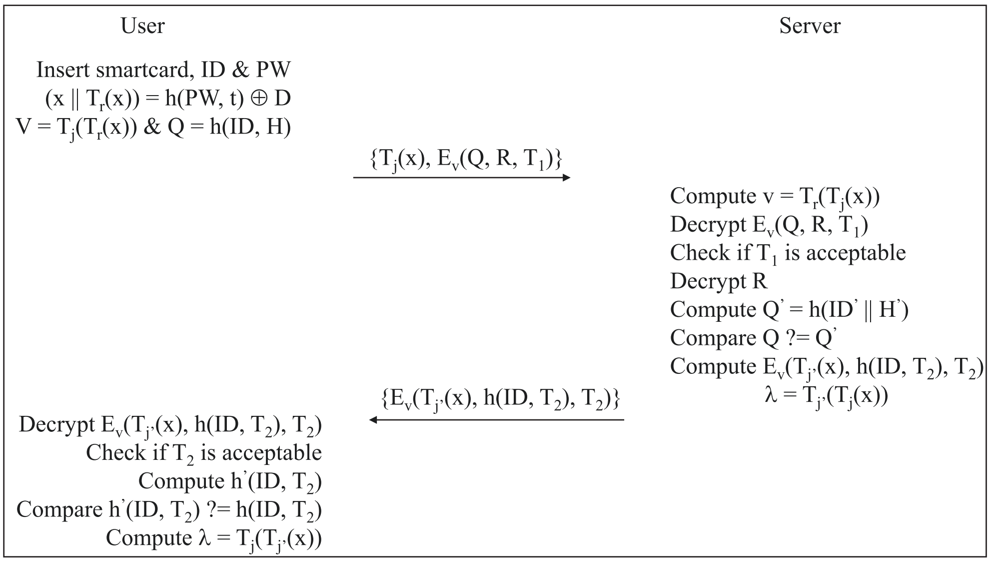

Figure 1: Han-Yu Lin's authentication scheme.

$T_{ab}(x) \bmod p$. We call this *Diffie-Hellman* problem *chaotic maps*.

3. Cryptanalysis of Typical Related Works

In this section, we review some typical related works applying *Chebyshev chaotic map* in user authentication schemes. Also, we analyse on their security.

3.1. Han-Yu Lin's Scheme. Lin's scheme [23] includes four phases: system initialization, user registration, authentication and password change phases.

(1) *Initialization phase*. The server S chooses all necessary parameters (r, x, $T_r(x)$, $h(.)$, $E_k(.)$). Especially (x, $T_r(x)$) is written into user's smartcard.

(2) *Registration phase*. The user U chooses identity ID, password PW, and random value t, then computes $H = h(PW, t)$, and sends $\{ID, H\}$ to S through a secure channel. Once receiving U's messages, S checks ID's validity and uses master key s to compute $R = E_s(ID, H)$, $D = H \oplus (x \parallel T_r(x))$. Finally, S sends $\{R, h(.), E_k(.), D\}$ to U through a secure channel. U receives S's incoming smartcard and inserts t into it.

(3) *Authentication phase*. When U authenticates with S, U provides (ID, PW) and smartcard into the terminal. Below are some steps for authentication (see Figure 1):

 (a) Smartcard chooses j and computes $(x \parallel T_r(x)) = h(PW, t) \oplus D$, $v = T_j(T_r(x))$ and $Q = h(ID, H)$.

 (b) Next, U sends $(T_j(x), E_v(Q, R, T_1))$ to S, where T_1 is receiving time-stamp.

 (c) Once receiving U's messages, S computes $v = T_r(T_j(x))$ and decrypts $E_v(Q, R, T_1)$ and then checks T_1.

 (d) Next, S decrypts R with s to recover (ID', H') and computes $Q' = h(ID', H')$. If $Q = Q'$ then S successfully authenticates with U. Otherwise, S terminates the session.

 (e) Then, S chooses j' and sends $E_v(T_{j'}(x), h(ID, T_2), T_2)$ to U, where T_2 is time-stamp when S sends the message to U.

 (f) Once receiving S's message, U decrypts and checks T_2's validity. At the same time, U computes $h'(ID, T_2)$. U checks the validity of $h'(ID, T_2)$?= $h(ID, T_2)$. If this condition holds, U successfully authenticates with S; otherwise U terminates the session.

 (g) After successfully authentication phase, both U and S compute session key $\lambda = T_j(T_{j'}(x))$ for later usage.

(4) *Password change phase*. U provides smartcard, old PW and new PW^*. Then, smartcard randomly chooses i and computes $H' = h'(PW, t)$, $(x \parallel T_r(x)) = H' \oplus D$, $\eta = T_i(T_r(x))$ and $H^* = h(PW^*, t)$ and then sends $\{T_i(x), E_\eta(H', H^*, R)\}$ to S. Once receiving U's messages, S computes $\eta = T_s(T_i(x))$ and decrypts $E\eta(H', H^*, R)$ with η and s. Finally, S compares H' ?= H. If this holds, S returns $R^* = E_s(ID, H^*)$ to smartcard, then it updates $R = R^*$.

3.2. Security Analysis on Han-Yu Lin's Scheme. In this subsection, we also review some limitations existing in this scheme.

(i) In this phase, $(x \parallel T_r(x))$ is the same in all users' smartcard. Therefore, malicious user can exploit this to launch an offline password-guessing attack if another user's smartcard is lost. Suppose malicious user extracts $\{R, h(.), E_k(.), D, t\}$ of another user. Then, malicious user computes H by performing $H = D \oplus (x \parallel T_r(x))$, where $(x \parallel T_r(x))$ belongs to malicious

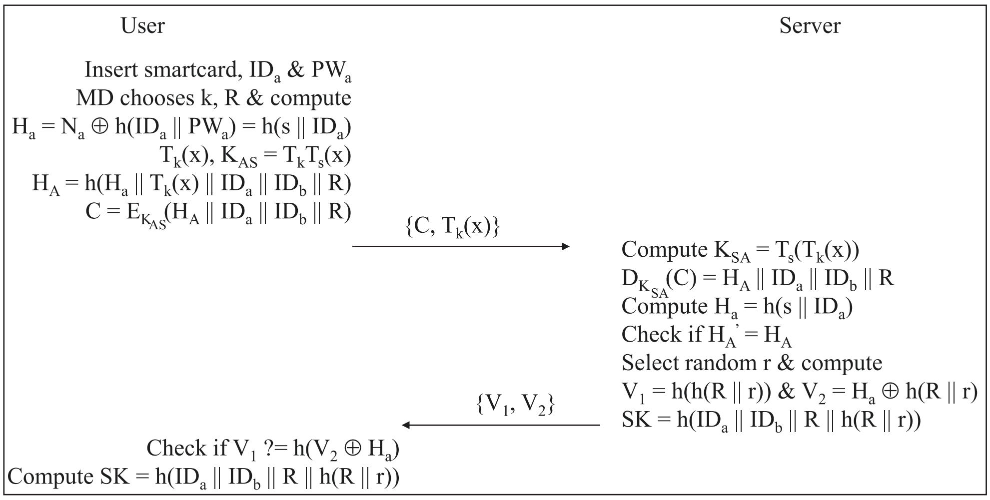

Figure 2: Hongfeng Zhu's authentication scheme.

user's smartcard. With H, malicious user builds H ?= $h(PW, t)$ and Han-Yu Lin's scheme is vulnerable to this kind of attack.

(ii) Also, Han-Yu Lin's scheme is vulnerable to contributory property of key agreement. In this scheme, S can determine common session key without U's random value. Below are some steps S can perform:

(1) Find $j\prime$ such that $T_{j'}(x) = T_j(x)$, where $j\prime = (\cos^{-1}(T_j(x)) + k2\pi)/\cos^{-1}(x) \mid k \in \mathbb{Z}$.

(2) Then, S chooses session key $\lambda\prime$ and computes $v = (\cos^{-1}(\lambda') + k2\pi)/(j' \times \cos^{-1}(x)) \mid k \in \mathbb{Z}$.

(3) S has $\lambda' = T_v(T_{j'}(x))$ and transmits $T_v(x)$ to U. When U receives $T_v(x)$, U computes $T_j(T_v(x)) = T_v(T_j(x)) = T_v(T_{j'}(x)) = \lambda\prime$. So, Lin's scheme is vulnerable to this property.

3.3. Hongfeng Zhu's Scheme. Zhu's scheme [24] includes four phases: registration, login, authentication, and password change phases.

(1) *Registration phase*. In this phase, the user U chooses password PW_a, then randomly chooses value t, and computes $W_a = h(PW_a \parallel t)$. Next, U sends (ID_a, W_a) to the server S through a secure channel. After receiving U's (ID_a, W_a), S computes $H_a = h(s \parallel ID_a)$, $n_a = h(W_a \parallel ID_a) \oplus H_a$, and sends $\{n_a, x, T_s(x)\}$ to U through a secure channel. Once receiving S's messages, U computes $N_a = h(ID_a \parallel PW_a) \oplus n_a \oplus h(W_a \parallel ID_a) = h(ID_a \parallel PW_a) \oplus H_a$. Finally, U saves $\{N_a, x, T_s(x)\}$ into U's device.

(2) *Login-authentication phases* (see Figure 2). In this phase, U inputs (ID_a, PW_a) and U's device randomly chooses two values k, R to compute $H_a = N_a \oplus h(ID_a \parallel PW_a) = h(s \parallel ID_a)$, $T_k(x)$, $K_{AS} = T_k T_s(x)$, $H_A = h(H_a \parallel T_k(x) \parallel ID_a \parallel ID_b \parallel R)$, $C = E_{K_{AS}}(H_A \parallel ID_a \parallel ID_b \parallel R)$. Then, U sends $\{C, T_k(x)\}$ to S. After receiving U's messages, S computes $K_{SA} = T_s T_k(x)$ by using $T_k(x)$ and master key s. Then, S decrypts C to recover $H_A \parallel ID_a \parallel ID_b \parallel R$. S computes $H_a = h(s \parallel ID_a)$ and $H_A{}' = h(H_a \parallel T_k(x) \parallel ID_a \parallel ID_b \parallel R)$. Finally, S checks $H_A{}'$?= H_A. If this holds, S randomly chooses r and computes $V_1 = h(h(R \parallel r))$, $V_2 = H_a \oplus h(R \parallel r)$, $SK = h(ID_a \parallel ID_b \parallel R \parallel h(R \parallel r))$ and sends $\{V_1, V_2\}$ to U. Otherwise, S terminates the session. Once receiving S's $\{V_1, V_2\}$, U's device checks V_1 ?= $h(V_2 \oplus H_a)$. If this does not hold, U's device terminates the session, otherwise it computes $SK = h(ID_a \parallel ID_b \parallel R \parallel h(R \parallel r))$.

(3) *Password change phase*. In this phase, U provides old PW_a, new $PW_a{}'$, and ID. Then, the device chooses random value $k\prime$ and computes $H_a = N_a \oplus h(ID_a \parallel PW_a) = h(s \parallel ID_a)$, $N_a{}' = N_a \oplus h\ (ID_a \parallel PW_a) \oplus h(ID_a \parallel PW_a{}')$, $T_k(x)\prime$, $K_{AS}{}' = T_k{}' T_s(x)$, $H_A{}' = h(H_a \parallel T_k(x)' \parallel ID_a \parallel ID_b \parallel h(N_a{}'))$ and $C\prime = E_{K'_{AS}}(H_A{}' \parallel ID_a \parallel ID_b \parallel h(N_a{}'))$. The device sends $\{C\prime, T_k(x)\prime\}$ to S. S computes $K_{SA}{}' = T_s T_k{}'(x)$ to decrypt $D_{K'_{AS}}(C) = H_A{}' \parallel ID_a \parallel ID_b \parallel h(N_a{}')$. Next, S computes $H_a = h(s \parallel ID_a)$ and $H'_{A'} = h(H_a \parallel T_k(x)\prime \parallel ID_a \parallel ID_b \parallel h(N_a{}'))$. S checks $H'_{A'}$?= $H_{A'}$. If this does not hold, S rejects. Otherwise, password-update is accepted and device computes $V_3 = h(K_{SA}{}' \parallel response)$, where *response* = "*update*" or "*refuse*". Finally, S returns V_3 to U. Once U's device receives $\{V_3, response\}$, it checks V_3 ?= $h(K_{AS}{}' \parallel response)$. If this holds, it updates (N_a with $N_a{}'$) or rejects if response is *refused*.

3.4. Security Analysis on Hongfeng Zhu's Scheme. Next, we also review some limitations existing in this scheme.

(i) Hongfeng Zhu's scheme does not provide PFS. If important information such as master key s is leaked, U's session key will be easily computed with previous exchanged messages. Suppose an adversary captures $\{C, T_k(x)\}$ and $\{V_1, V_2\}$ at another session between U and S. With s, the adversary computes

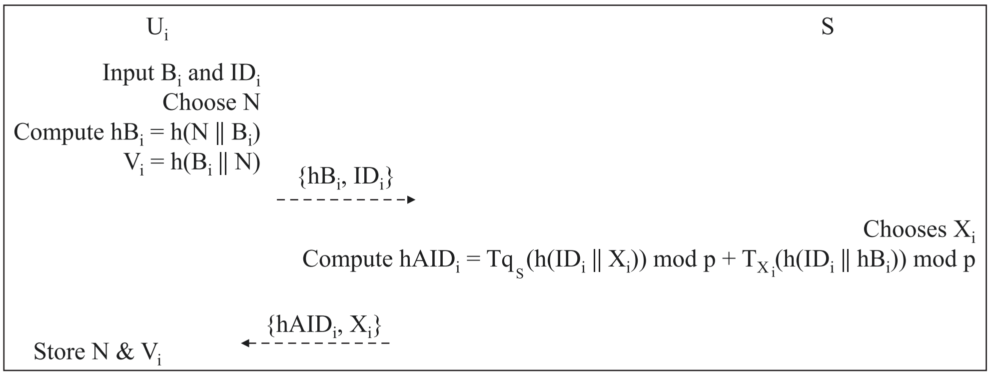

Figure 3: Proposed scheme's registration phase.

$K_{SA} = T_s T_k(x)$ and decrypts $D_{K_{SA}}(C) = H_A \parallel ID_a \parallel ID_b \parallel R$. Next, he/she computes $H_a = h(s \parallel ID_a)$ and extracts $h(R \parallel r) = V_2 \oplus H_a$. Finally, he/she computes $SK = h(ID_a \parallel ID_b \parallel R \parallel h(R \parallel r))$. Clearly, this scheme does not satisfy PFS.

(ii) This scheme does not store password-confirmation message at U's device, so password-update phase must connect to S. However, adding a value in U's device helps login phase be more secure if the device is stolen. For example, we add $L = h(ID_a \parallel PW_a \parallel H_a)$ into the device. When logging, U inputs ID_a and PW_a. U's device computes $H_a = N_a \oplus h(ID_a \parallel PW_a)$ and then checks $L \mathrel{?=} h(ID_a \parallel PW_a \parallel H_a)$. If this holds, U is real device's owner. With L, this scheme can resist offline password-guessing attack if device's information is leaked because L contains authentication key $H_a = h(s \parallel ID_a)$. In password change phase, U needs correcting old PW to pass $L = h(ID_a \parallel PW_a \parallel H_a)$. If this holds, U's device will accept new PW' provided by U. Next, U's device recomputes $N_a' = N_a \oplus h(ID_a \parallel PW_a) \oplus h(ID_a \parallel PW_a')$ and $L' = h(ID_a \parallel PW_a' \parallel H_a)$. Clearly, password change phase is more efficient than previous old phase.

4. Proposed Scheme

Our proposed scheme using *Chebyshev* polynomial includes five phases: initialization, registration, authentication, and biometrics update phases. Below are some notations used in our scheme:

(i) U_i: the ith user

(ii) ID_i: the ith user's identification

(iii) B_i: the ith user's biometrics

(iv) S: the server

(v) q_S: the server's master key

(vi) $h(.)$: hash function

(vii) sk: common session key

(viii) SC: the smartcard

(ix) $\oplus$, $\parallel$: the XOR and concatenation operations

(x) $T(.)$: *Chebyshev* polynomial operation

4.1. Initialization Phase. In this phase, we choose a huge prime number *k-bit* p. Then S chooses $q_S \in [1, p-1]$ and h: $\{0,1\}^* \longrightarrow \{0,1\}^k$. Finally, S publishes $\{p, T(.), h(.)\}$

4.2. Registration Phase. U_i provides B_i and ID_i. Also, U_i randomly chooses N. Then, U_i computes $hB_i = h(N \parallel B_i)$ and $V_i = h(B_i \parallel N)$. Finally, U_i sends $\{hB_i, ID_i\}$ to S through a secure channel. On receiving the U_i's information, S checks ID_i's validity. Then, S randomly chooses X_i. S computes $hAID_i = T_{q_S}(h(ID_i \parallel X_i)) \bmod p + T_{X_i}(h(ID_i \parallel hB_i)) \bmod p$, then S sends $\{hAID_i, X_i\}$ to U_i through a secure channel. U_i stores the information sent from S into a SC (see Figure 3).

4.3. Authentication Phase. U_i provides B_i and ID_i at the terminal. Then SC checks if $V_i \mathrel{?=} h(B_i \parallel N)$. If this holds, SC computes $hB_i = h(N \parallel B_i)$, $AID_i = hAID_i - T_{X_i}(h(ID_i \parallel hB_i)) \bmod p$. SC chooses $r_i \in [1, p-1]$ and computes $R' = T_{r_i}(h(ID_i \parallel X_i)) \bmod p$, $R_i = T_{r_i}(AID_i) \bmod p$, $M_i = h(R_i, AID_i)$, and $CID = ID_i \oplus h(R_i)$. SC sends $\{X_i, CID, M_i, R'\}$ to S through a common channel (see Figure 4).

When receiving U_i's login message, S computes R_i' and AID_i^*, where $R_i' = T_{q_S}(R') \bmod p$, $ID_i = CID \oplus h(R_i')$ and $AID_i^* = T_{q_S}(h(ID_i \parallel X_i)) \bmod p$. Then S checks if $M_i \mathrel{?=} h(R_i', AID_i^*)$. If this does not hold, S terminates the session. Otherwise, S chooses $r_S \in [1, p-1]$ and computes $R_S = T_{r_S}(AID_i^*) \bmod p$, $S' = R_S + R_i'$, $sk = h(T_{r_S}(R_i'))$ and $M_S = h(R_S, AID_i^*)$. S sends $\{S', M_S\}$ to U_i through a common channel.

When receiving $\{S', M_S\}$, U_i's SC computes $R_S' = S' - R_i$ and checks if $M_S \mathrel{?=} h(R_S', AID_i)$. If this holds, SC computes $sk = h(T_{r_i}(R_S'))$, $M_{US} = h(R_S', T_{r_i}(R_S'))$. Then, SC sends $\{M_{US}\}$ to S through a common channel.

S checks if $M_{US} \mathrel{?=} h(R_S, T_{r_S}(R_i'))$. If this holds, S accepts U_i. U_i and S use sk to encrypt the information after authentication phase.

4.4. Biometrics Update Phase. When U_i changes his/her biometrics, U_i's SC checks if $V_i \mathrel{?=} h(B_i \parallel N)$. If this holds, SC computes $V_{i_{new}} = h(B_{i_{new}} \parallel N)$ and $hAID_{i_{new}} = T_{X_i}(h(ID_i \parallel hB_{i_{new}})) \bmod p + hAID_i - T_{X_i}(h(ID_i \parallel hB_i)) \bmod p$. Finally, SC replaces V_i and $hAID_i$ with $V_{i_{new}}$ and $hAID_{i_{new}}$.

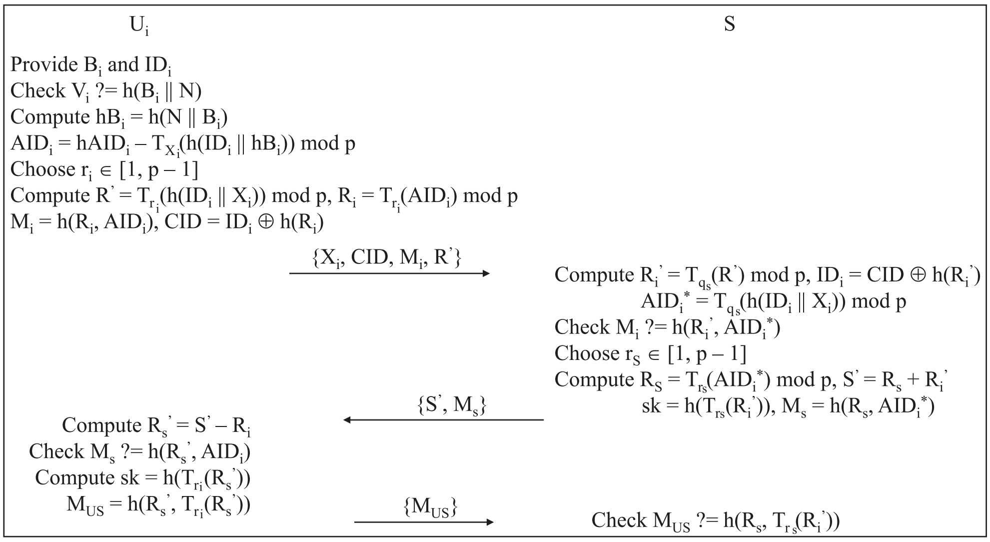

FIGURE 4: Proposed scheme's authentication phase.

TABLE 1: The assumptions in BAN-logic.

Assumptions
A_1: $U \mid\equiv (U \overset{ID}{\longleftrightarrow} S)$ – U believes U can share ID with S
A_2: $U \mid\equiv (U \overset{K}{\longleftrightarrow} S)$ – U believes U can share K with S
A_3: $U \mid\equiv (S \Longrightarrow (U \overset{SK}{\longleftrightarrow} S))$ – U believes S controls the sharing of sk between U and S
A_4: $S \mid\equiv (U \Longrightarrow (U \overset{ID}{\longleftrightarrow} S))$ – S believes U controlsthe sharing of ID between U and S
A_5: $S \mid\equiv (U \Longrightarrow (U \overset{SK}{\longleftrightarrow} S))$ –S believes U controls the sharing of sk between U and S
A_6: $S \mid\equiv (S \overset{K}{\longleftrightarrow} U)$ – S believes S can share K with U
A_7: $U \mid\equiv \#(r_S \otimes K)$ – U believes the challenge messages from S is fresh
A_8: $S \mid\equiv \#(r_U \otimes K)$ – S believes the challenge messages from U is fresh

5. Security and Efficiency Analyses

In this section, we analyse our scheme on security and efficiency aspects. Also, our scheme's design is correctly proved with BAN-*logic* [25], while its security is presented in each concrete attack case.

5.1. Correctness Analysis. Before getting into details about security, we will prove our scheme's correctness with BAN-logic. We inherit some objectives from [26] because we see that they are reasonable ones, which authentication scheme must achieve to successfully share partner's identities and session keys. For simplicity, we let K denote user's long-term key shared by server at registration phase, sk denote session key, and $\otimes$ denote *Chebyshev* operation. Firstly, our scheme must satisfy some assumptions as shown in Table 1 (this is a must in this model)

These assumptions represent the first necessary believes of user and server. For example, when the users register with server, it is mean that they believe they can share identity with server (A_1). Next, we will normalize all messages exchanged between user and server.

(i) From the message $\{CID\}$ we have $<U \overset{ID}{\longleftrightarrow} S,\ U \overset{K}{\longleftrightarrow} S,\ r_U \otimes K>$

(ii) From the message $\{M_i\}$ we have $< r_U \otimes K,\ U \overset{K}{\longleftrightarrow} S>$

(iii) From the third messages $\{M_{US}\}$ we have $< r_S \otimes K,\ U \overset{K}{\longleftrightarrow} S>$

(iv) From the fourth message $\{M_{US}\}$ we have $<U \overset{K}{\longleftrightarrow} S,\ U \overset{sk}{\longleftrightarrow} S>$

The normalization is an arrangement of information exchanged between user and server. For example, *CID* contains identity, challenge information $r_U \otimes K$, and long-term key K. Normalization helps to highlight the important data in the messages. Next, we will demonstrate how our scheme satisfies seven lemmas that we reorganized from [32].

Lemma 1. *If the server believes authentication key (long-term key) is successfully shared with user and the user's messages encrypted with this key are fresh, the server will believe that the user believes his/her identity is successfully shared with server.*

$$\frac{S \mid\equiv \left(S \overset{K}{\longleftrightarrow} U\right), S \mid\equiv \#\left(r_U \otimes K\right)}{S \mid\equiv \left(U \mid\equiv \left(U \overset{ID}{\longleftrightarrow} S\right)\right)} \qquad (4)$$

Proof. With A_6 and *CID*, we apply *message-meaning* rule to have

$$\frac{S \mid\equiv S \overset{K}{\longleftrightarrow} U, S \triangleleft < U \overset{ID}{\longleftrightarrow} S, U \overset{K}{\longleftrightarrow} S, r_U \otimes K >}{S \mid\equiv U \mid\sim < U \overset{ID}{\longleftrightarrow} S, U \overset{K}{\longleftrightarrow} S, r_U \otimes K >} \quad (5)$$

With A_8, we apply *freshness* rule to have

$$\frac{S \mid\equiv \#\left(r_U \otimes K\right)}{S \mid\equiv \#\left(< U \overset{ID}{\longleftrightarrow} S, U \overset{K}{\longleftrightarrow} S, r_U \otimes K >\right)} \quad (6)$$

With (5) and (6), we apply *nonce-verification* rule to have

$$\frac{S \mid\equiv U \mid\sim < U \overset{ID}{\longleftrightarrow} S, U \overset{K}{\longleftrightarrow} S, r_U \otimes K >, S \mid\equiv \#\left(< U \overset{ID}{\longleftrightarrow} S, U \overset{K}{\longleftrightarrow} S, r_U \otimes K >\right)}{S \mid\equiv U \mid\equiv < U \overset{ID}{\longleftrightarrow} S, U \overset{K}{\longleftrightarrow} S, r_U \otimes K >} \quad (7)$$

With (7), we apply *believe* rule:

$$\frac{S \mid\equiv U \mid\equiv < U \overset{ID}{\longleftrightarrow} S, U \overset{K}{\longleftrightarrow} S, r_U \otimes K >}{S \mid\equiv U \mid\equiv U \overset{ID}{\longleftrightarrow} S} \quad (8)$$

So, with A_6 and A_8 we successfully demonstrate how our scheme satisfies *Lemma 1*. □

Lemma 2. *If the server believes the user also believes his/her identity is successfully shared with each other and user totally controls this identity's sharing, the server also believes user's identity is successfully shared with each other.*

$$\frac{S \mid\equiv \left(U \mid\equiv \left(U \overset{ID}{\longleftrightarrow} S\right)\right), S \mid\equiv \left(U \Longrightarrow \left(U \overset{ID}{\longleftrightarrow} S\right)\right)}{S \mid\equiv \left(U \overset{ID}{\longleftrightarrow} S\right)} \quad (9)$$

Proof. With *Lemma 1* and A_4, we apply *jurisdiction* rule to have

$$\frac{S \mid\equiv \left(U \mid\equiv \left(U \overset{ID}{\longleftrightarrow} S\right)\right), S \mid\equiv \left(U \Longrightarrow \left(U \overset{ID}{\longleftrightarrow} S\right)\right)}{S \mid\equiv \left(U \overset{ID}{\longleftrightarrow} S\right)} \quad (10)$$

So, with *Lemma 1* and A_4, we successfully demonstrate how our scheme satisfies *Lemma 2*. □

Lemma 3. *If the user believes authentication key is successfully shared with server and the server's messages encrypted with this key are fresh, the user will believe the server also believes user's identity is successfully shared with each other.*

$$\frac{U \mid\equiv \left(U \overset{K}{\longleftrightarrow} S\right), U \mid\equiv \#\left(r_S \otimes K\right)}{U \mid\equiv \left(S \mid\equiv \left(U \overset{ID}{\longleftrightarrow} S\right)\right)} \quad (11)$$

Proof. With A_2 and M_S, we apply *jurisdiction* rule to have

$$\frac{U \mid\equiv U \overset{K}{\longleftrightarrow} S, U \triangleleft < r_S \otimes K, U \overset{K}{\longleftrightarrow} S >}{U \mid\equiv S \mid\sim < r_S \otimes K, U \overset{K}{\longleftrightarrow} S >} \quad (12)$$

With (12) and A_7, we apply *freshness* rule to have

$$\frac{U \mid\equiv \#\left(r_S \otimes K\right), U \mid\equiv S \mid\sim < r_S \otimes K, U \overset{K}{\longleftrightarrow} S >}{U \mid\equiv \# < r_S \otimes K, U \overset{K}{\longleftrightarrow} S >} \quad (13)$$

With (12) and (13), we apply *nonce-verification* rule to have

$$\frac{U \mid\equiv S \mid\sim < r_S \otimes K, U \overset{K}{\longleftrightarrow} S >, U \mid\equiv \# < r_S \otimes K, U \overset{K}{\longleftrightarrow} S >}{U \mid\equiv S \mid\equiv < r_S \otimes K, U \overset{K}{\longleftrightarrow} S >} \quad (14)$$

With (14), we apply *believe* rule to have

$$\frac{U \mid\equiv S \mid\equiv < r_S \otimes K, U \overset{K}{\longleftrightarrow} S >}{U \mid\equiv \left(S \mid\equiv \left(U \overset{ID}{\longleftrightarrow} S\right)\right)} \quad (15)$$

□

So, with A_2 and A_7 we successfully demonstrate how our scheme satisfies *Lemma 3*. In short, with three lemmas we can say that both server and user believe and successfully share their identities with each other. Next, we need to prove the similar thing for session key.

Lemma 4. *If the user believes that authentication key is successfully shared with server and server's messages encrypted with this key are fresh, the user will believe the server also believes session key is successfully shared with each other.*

$$\frac{U \mid\equiv \left(U \overset{K}{\longleftrightarrow} S\right), U \mid\equiv \#\left(r_S \otimes K\right)}{U \mid\equiv \left(S \mid\equiv \left(S \overset{SK}{\longleftrightarrow} U\right)\right)} \quad (16)$$

Proof. With A_2 and M_{US}, we apply *message-meaning* rule to have

$$\frac{U \mid\equiv U \overset{K}{\longleftrightarrow} S, U \triangleleft < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >}{U \mid\equiv S \mid\sim < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >} \quad (17)$$

With A_7 and M_{US}, we apply *freshness* rule to have

$$\frac{U \mid\equiv \#\left(r_S \otimes K\right), U \triangleleft < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >}{U \mid\equiv \# < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >} \quad (18)$$

With (17) and (18), we apply *believe* rule to have

$$\frac{U \mid\equiv S \mid\sim < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >, U \mid\equiv \# < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >}{U \mid\equiv S \mid\equiv < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >} \quad (19)$$

With (19), we apply *believe* rule to have

$$\frac{U \mid\equiv S \mid\equiv < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >}{U \mid\equiv S \mid\equiv S \overset{sk}{\longleftrightarrow} U} \tag{20}$$

So, with A_2 and A_7 we successfully demonstrate how our scheme satisfies *Lemma 4*. □

Lemma 5. *If the user believes the server totally controls session key's sharing and the server also believes session key is successfully shared with user, the user will believe this session key's sharing.*

$$\frac{U \mid\equiv \left(S \Longrightarrow \left(U \overset{SK}{\longleftrightarrow} S\right)\right), U \mid\equiv \left(S \mid\equiv \left(S \overset{SK}{\longleftrightarrow} U\right)\right)}{U \mid\equiv \left(U \overset{SK}{\longleftrightarrow} S\right)} \tag{21}$$

Proof. With A_3 and *Lemma 4*, we apply *jurisdiction* rule to have

$$\frac{U \mid\equiv S \Longrightarrow U \overset{sk}{\longleftrightarrow} S, U \mid\equiv S \mid\equiv S \overset{sk}{\longleftrightarrow} U}{U \mid\equiv U \overset{sk}{\longleftrightarrow} S} \tag{22}$$

So, with A_3 and *Lemma 4*, we successfully demonstrate how our scheme satisfies *Lemma 5*. □

Lemma 6. *If the server believes authentication key is successfully shared with user and the user's messages encrypted with this key are fresh, the server will believe the user also believes this session key's sharing.*

$$\frac{S \mid\equiv \left(S \overset{K}{\longleftrightarrow} U\right), S \mid\equiv \#\left(r_U \otimes K\right)}{S \mid\equiv \left(U \mid\equiv \left(U \overset{SK}{\longleftrightarrow} S\right)\right)} \tag{23}$$

Proof. With A_6 and M_{US}, we apply *message-meaning* rule to have

$$\frac{S \mid\equiv S \overset{K}{\longleftrightarrow} U, S \triangleleft < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >}{S \mid\equiv U \mid\sim < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >} \tag{24}$$

With A_8 and M_{US}, we apply *freshness* rule to have

$$\frac{S \mid\equiv \#\left(r_U \otimes AID\right), S \triangleleft < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >}{S \mid\equiv \# < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >} \tag{25}$$

With (24) and (25), we apply *nonce-verification* to have

$$\frac{S \mid\equiv U \mid\sim < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >, S \mid\equiv \# < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >}{S \mid\equiv U \mid\equiv < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >} \tag{26}$$

With A_6 and (26), we apply *believe* rule to have

$$\frac{S \mid\equiv S \overset{K}{\longleftrightarrow} U, S \mid\equiv U \mid\equiv < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >}{S \mid\equiv U \mid\equiv U \overset{sk}{\longleftrightarrow} S} \tag{27}$$

So, with A_6 and A_8, we successfully demonstrate how our scheme satisfies *Lemma 6*. □

Lemma 7. *If the server believes the user totally controls the session key's sharing, the server will believe the session key is successfully shared with user.*

$$\frac{S \mid\equiv \left(U \Longrightarrow \left(U \overset{SK}{\longleftrightarrow} S\right)\right)}{S \mid\equiv \left(S \overset{sk}{\longleftrightarrow} U\right)} \tag{28}$$

Proof. With (26) and A_5, we apply *message-meaning* rule to have

$$\frac{S \mid\equiv U \Longrightarrow U \overset{sk}{\longleftrightarrow} S, S \mid\equiv U \mid\equiv < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >}{S \mid\equiv < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >} \tag{29}$$

With (29), we apply *believe* rule to have

$$\frac{S \mid\equiv < U \overset{K}{\longleftrightarrow} S, U \overset{sk}{\longleftrightarrow} S >}{S \mid\equiv S \overset{sk}{\longleftrightarrow} U} \tag{30}$$

□

So, with A_5 we completely demonstrate how our scheme satisfies *Lemma 7*. Finally, we can say that both server and user believe the common session key in our scheme.

5.2. Security Analysis. Before getting into details about some kinds of attacks, we will use random oracle model to prove the security for the session key in *Chebyshev* polynomial case (see [27, 28] for more details). At first, we need to remind the model's circumstance. Assuming another actor B has $\Omega = \{T_{\mathrm{p}}(x), T_{\mathrm{q}}(x)\}$) and B needs to compute $T_{\mathrm{p}\times\mathrm{q}}(x)$, B has some *oracles* ***Client*** and***Server*** with all their instances at different times. B also has an algorithm A being able to break our scheme to compute the session key with given probability ε. B will use A to find the session key and then compute $T_{\mathrm{p}\times\mathrm{q}}(x)$ to solve CMDHP. To achieve this, B must "inject" Ω's parameters into the messages when A interacts with the oracles' instances, and B also simulates an appropriate environment suitable for A to operate. Note that our scheme uses hash function and it is considered as an oracle. Next, we claim our theorem about the session key's security.

Theorem 8. *Let A be an adversary breaking our scheme in the meaning of AKESecurity in time t_{A}, using q_{Send} Send queries and q_{Hash} Hash queries. We have*

$$\begin{aligned} &Adv_{P}^{AKE}\left(A, t_{\mathrm{A}}, q_{Send}, q_{Hash}\right) \\ &\quad \leq q_{Hash} \times q_{Send} \\ &\quad\quad \times Adv_{E(\mathbb{F}_q)}^{ECDHP(CMDHP)}\left(B, t_{\mathrm{B}}, q_{Send}, q_{Hash}\right) \end{aligned} \tag{31}$$

where B is an adversary breaking CMDHP in t_{B}. The meaning of theorem is that A's successful probability breaking our scheme in the meaning of AKESecurity is less than B's successful probability breaking CMDHP. According to CMDHP, B's success probability is extremely low, and so is A's successful probability breaking our scheme. Therefore, we can claim that our scheme has secure session key in the meaning of AKESecurity.

Table 2: The security feature comparison among the schemes.

Kinds of attack	Lin[23]	Zhu[24]	Ours
Password-guessing	✘	✔	✔
Replay	∅[1]	✔	✔
User anonymity	✔	∅	✔
Impersonation	✔	✔	✔
Man-in-the-middle	∅	∅	✔
Parallel	∅	∅	✔
Two-factor	✘	✔	✔
Perfect secrecy session key	∅	✘	✔

[1]The authors do not consider.

Proof. Assume that we have actor B. This time, B needs to create some instances of oracles***Client*** and***Server*** with Ω's parameter. B also simulates an appropriate environment where A can operate.

When A sends *Send*("*Start*") to $U_x^i \in$***Client***, U_x^i replies m_1 to A (note that $m_1 = \{$*identity encrypted*, *challenge information*, *confirmation*$\}$ depends on concrete scheme). Maybe, A sends *Send*("*Start*") to some simulated oracle, for example, U_B^v. This time, B needs to inject Ω's parameters into m_1 with concrete scheme's rules. Finally, B sends m_1 to A. When A sends $Send(m_1)$ to $S_y^j \in$***Server***, S_y^j replies m_2 to A (note that $m_2 = \{$*confirmation*, *challenge information*$\}$ depends on concrete scheme). Maybe, A sends $Send(m_1)$ to simulated oracle S_B^v. This time, B also needs to inject Ω's parameters into m_2 and sends m_2 to A. When A sends $Send(m_2)$ to $S_y^j \in$***Server***, S_y^j replies $m_3 = \{$*final confirmation*$\}$ to A. Maybe, A sends $Send(m_2)$ to simulated oracle S_B^v. B randomly chooses $sk \xleftarrow{\$} \{0,1\}^n$, computes *final confirmation message* with this random sk, and sends m_3 to A.

Sometimes, A sends wrong *Send* queries to the instances, so there are some oracles with **"Accept"** state and some oracles with "Reject" state. However, when A finishes sending q_{Send} *Send* queries, all instances A interacts with must have **"Terminated"** state which is *true*.

When A sends *Corrupt* and *Reveal* queries to these instances, their state will determine what A obtains, such as long-term key or session key. When A sends *Corrupt* and *Reveal* queries to the oracles B simulates, B will generate a random string representing session key for them. When A sends *hash* queries, B will let hash's *oracle* interact with A.

Finally, B activates A to sends a unique *Test* query to simulated oracle or indicated oracle, and B expects A to correctly guess bit b of this instance. In other words, B wants A to correct guess this instance's session key with A's successful probability. We see that if B is success, B needs three following consecutive factors:

(i) B needs A to correctly find sk of simulated oracle, and A's successful probability is $\varepsilon = Adv_P^{AKE}(A, t_A)$.

(ii) Furthermore, when A correctly finds $sk = h(T_{p\times q}(x)\ldots)$, A had found "$T_{p\times q}(x)$" satisfying with this sk. Clearly, if A sends q_{Hash} queries to hash's oracle, there is at least one-time A succeeds. So, $\gamma \geq 1/q_{Hash}$. (Note that: the session key sk will be always computed with hash function.)

(iii) Next, when A correctly finds $T_{p\times q}(x)$, A must correct guess q or s. Clearly, if A sends q_{Send} queries to the oracle, there is at least one-time A succeeds. So, $\mu \geq 1/q_{Send}$.

Finally, we have $Adv_{E(\mathbb{F}_q)}^{ECDHP(CMDHP)}(B, t_B, q_{\text{Send}}, q_{\text{Hash}}) = \varepsilon \times \gamma \times \mu \geq \varepsilon \times 1/(q_{Hash} * q_{Send}) \Longrightarrow q_{\text{Send}} \times q_{\text{Hash}} \times Adv_{E(\mathbb{F}_q)}^{ECDHP(CMDHP)}(B, t_B, q_{\text{Send}}, q_{\text{Hash}}) \geq \varepsilon = Adv_P^{AKE}(A, t_A, q_{\text{Send}}, q_{\text{Hash}})$. □

In this subsection, we analyse our scheme on security aspect (see Table 2).

(1) *Password-guessing attack*. If the smartcard's information is leaked, and the adversary can exploit to perform password-guessing attack. Therefore, the adversary has $V_i = h(B_i \parallel N)$ in the smartcard. Differently from password, B_i is the user's biometrics and it cannot be predicted. In short, our scheme easily resists this kind of attack.

(2) *Replay attack*. In this kind of attack, the adversary can replay the login message to impersonate the user. In our scheme, the adversary can replay $\{X_i, CID, M_i, R\prime\}$ to the server. Then, the server replies $\{S\prime, M_S\}$ to the adversary. At this time, the adversary cannot compute M_{US} because $R_i = T_{r_i}(AID_i) \bmod p$ is impossible to know. Therefore, our scheme can resist this kind of attack.

(3) *User anonymity*. In this kind of attack, the adversary eavesdrops $\{X_i, CID, M_i, R\prime\}$, $\{S\prime, M_S\}$ and $\{M_{US}\}$ of another user. The user's identity is encrypted with R_i, which includes the secret AID_i. Therefore, the adversary cannot trace who is authenticating and our scheme provides user anonymity.

(4) *Impersonation attack*. In this kind of attack, the adversary can impersonate either user or server. In our scheme, the adversary eavesdrops $\{X_i, CID, M_i, R\prime\}$ and $\{S\prime, M_S\}$. However, he must send M_{US} to cheat the server and this is impossible because r_i and AID_i are secret. Moreover, if he wants to impersonate the server, he needs to compute M_S and this is impossible because AID_i is secret. Therefore, our scheme can resist this kind of attack

Table 3: The efficiency comparison among the schemes.

The phases	Lin[23]	Zhu[24]	Ours
Registration	1×h+1×T+1×e/d	5×h	4×h+2×T
Authentication	4×h+5×T+5×e/d	9×h+3×T+2×e/d	14×h+8×T

(5) *Man-in-the-middle attack*. In this kind of attack, the adversary can cheat both user and server simultaneously. However, he must compute random values r_i and AID_i. From this information, he can derive R_i to cheat the server and derive M_{US} to cheat the user. Clearly, this information is secret, and the adversary cannot steal them.

(6) *Parallel attack*. In this kind of attack, the adversary uses another session's messages to exploit the others. In our scheme, this is impossible because each session has different random values. For example, another session has the unique values r_i and r_S, so all sessions have no relationship with each other.

(7) *Two-factor attack*. In this kind of attack, the adversary can steal the user's biometrics, and then use this information to compute authentication key. We see that the smartcard includes $\{N, V_i, hAID_i, X_i\}$, so if there is no B_i, the adversary cannot compute AID_i. Of course, if the smartcard is well-protected, the adversary has no way to compute AID_i. Our scheme can resist this kind of attack.

(8) *Perfect secrecy*. In this kind of attack, the adversary has all secret keys of the users and the server. Of course, the service must be stopped at this time. However, we need to prevent the adversary from knowing past-transactions, and this means that all session keys must be secret. In our scheme, the session key is constructed from r_i, r_S, and AID_i. Clearly, if the adversary knows $T_{r_i}(AID_i)$ and $T_{r_S}(AID_i)$, he cannot compute $T_{r_i}(T_{r_S}(AID_i))$ because of facing with CMDHP.

5.3. Efficiency Analysis. To compare efficiency between our scheme and previous ones, we let "h" be the hash operation, "e/d" be the encryption/decryption, and "T" be computational operation of polynomial. At registration phase, our scheme uses $4 \times h$ and $2 \times T$. Lin's scheme uses $1 \times h$, $1 \times T$, and $1\times e/d$. Zhu's scheme uses $5 \times h$. At authentication phase, our scheme uses $14 \times h$ and $8 \times T$. Lin's scheme uses $4 \times h$, $5 \times T$, and $5 \times e/d$. Zhu's scheme uses $9 \times h$, $3 \times T$, and $2 \times e/d$. Our scheme's computational cost is more than previous ones due to security enhancement (see Table 3).

Also, we let t_h, t_T, and $t_{e/d}$ denote running-time corresponding to each operation, for example, h, T, and e/d ($t_h \ll t_{e/d} < t_T$). To relatively measure the running-time of three operations, we conduct an experiment using *Java Cryptography Architecture* with *Bouncy Castle* library in *Android* mobile device, core 4 CPU 1.2 GHz, and we have $t_h \approx 0.00004$ms, $t_{e/d} \approx 0.09385$ms, and $t_T \approx 80$ ms (see Figure 5).

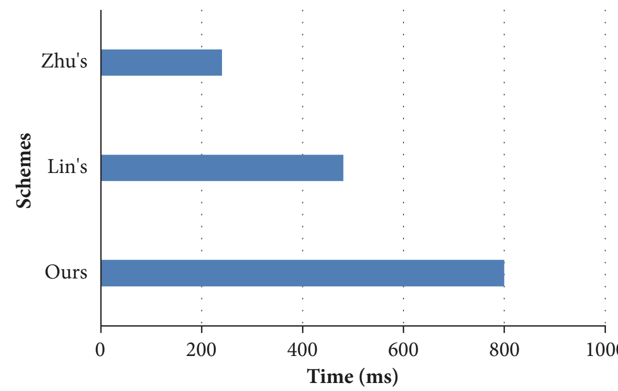

Figure 5: The running-time of three schemes.

6. Conclusion

This paper proposes a Chebyshev polynomials-based scheme in client-server environment. Although, our scheme takes more time than previous ones, it is advanced and resists some popular kinds of attack. Soon, we improve some techniques to reduce time-cost for computing Chebyshev polynomials.

Conflicts of Interest

We declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under Grant no. B2015-18-01.

References

[1] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[2] 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.

[3] E. Yoon and K. Yoo, "Improving the dynamic Id-based remote mutual authentication scheme," in *Proceedings of the OTM Workshops*, vol. 4277 of *LNCS*, pp. 499–507, 2006.

[4] I.-E. Liao, C.-C. Lee, and M.-S. Hwang, "Security enhancement for a dynamic ID-based remote user authentication scheme," in *Proceedings of the International Conference on Next Generation Web Services Practices, NWeSP 2005*, pp. 437–440, Republic of Korea, August 2005.

[5] 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 & Security*, vol. 28, no. 3-4, pp. 138–143, 2009.

[6] 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," *The Journal of Systems and Software*, vol. 84, no. 11, pp. 1892–1898, 2011.

[7] B. Huang, M. K. Khan, L. Wu, F. T. B. Muhaya, and D. He, "An Efficient Remote User Authentication with Key Agreement Scheme Using Elliptic Curve Cryptography," *Wireless Personal Communications*, vol. 85, no. 1, pp. 225–240, 2015.

[8] S. A. Chaudhry, H. Naqvi, K. Mahmood, H. F. Ahmad, and M. K. Khan, "An Improved Remote User Authentication Scheme Using Elliptic Curve Cryptography," *Wireless Personal Communications*, vol. 96, no. 4, pp. 5355–5373, 2017.

[9] S. A. Chaudhry, H. Naqvi, M. S. Farash, T. Shon, and M. Sher, "An improved and robust biometrics-based three factor authentication scheme for multiserver environments," *The Journal of Supercomputing*, vol. 74, no. 8, pp. 3504–3520, 2015.

[10] D. Hankerson, S. Vanstone, and A. J. Menezes, *Guide to Elliptic Curve Cryptography*, LNCS, Springer, New York, NY, USA, 2004.

[11] P. Bergamo, P. D'Arco, A. De Santis, and L. Kocarev, "Security of public-key cryptosystems based on Chebyshev polynomials," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 7, pp. 1382–1393, 2005.

[12] X. Hao, J. Wang, Q. Yang, X. Yan, and P. Li, "A chaotic map-based authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 37, article 9919, 2013.

[13] T.-F. Lee, "An efficient chaotic maps-based authentication and key agreement scheme using smartcards for telecare medicine information systems," *Journal of Medical Systems*, vol. 37, no. 6, pp. 9985–9994, 2013.

[14] H. Lin, "Chaotic map based mobile dynamic ID authenticated key agreement scheme," *Wireless Personal Communications*, vol. 78, no. 2, pp. 1487–1494, 2014.

[15] 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, pp. 1433–1440, 2013.

[16] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos, Solitons & Fractals*, vol. 37, no. 3, pp. 669–674, 2008.

[17] X. Liao, F. Chen, and K.-W. Wong, "On the security of public-key algorithms based on Chebyshev polynomials over the finite field ZN," *IEEE Transactions on Computer*, vol. 59, no. 10, pp. 1392–1401, 2010.

[18] A. Irshad, H. F. Ahmad, B. A. Alzahrani, M. Sher, and S. A. Chaudhry, "An efficient and anonymous chaotic map based authenticated key agreement for multi-server architecture," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 12, pp. 5572–5595, 2016.

[19] C. Wang and G. Xu, "Cryptanalysis of Three Password-Based Remote User Authentication Schemes with Non-Tamper-Resistant Smart Card," *Security and Communication Networks*, vol. 2017, Article ID 1619741, 14 pages, 2017.

[20] L. Kocarev, "Chaos-based cryptography: a brief overview," *IEEE Circuits and Systems Magazine*, vol. 1, no. 3, pp. 6–21, 2001.

[21] L. Kocarev and S. Lian, *Chaos-Based Cryptography: Theory, Algorithms and Applications*, vol. 354, Springer, Berlin, Germany, 2011.

[22] Z.-H. Li, Y.-D. Cui, and H.-M. Xu, "Fast algorithms of public key cryptosystem based on Chebyshev polynomials over finite field," *Journal of China Universities of Posts and Telecommunications*, vol. 18, no. 2, pp. 86–93, 2011.

[23] H.-Y. Lin, "Improved chaotic maps-based password-authenticated key agreement using smart cards," *Communications in Nonlinear Science and Numerical Simulation*, vol. 20, no. 2, pp. 482–488, 2015.

[24] H. Zhu, "Cryptanalysis and Improvement of a Mobile Dynamic ID Authenticated Key Agreement Scheme Based on Chaotic Maps," *Wireless Personal Communications*, vol. 85, no. 4, pp. 2141–2156, 2015.

[25] M. Burrows, M. Abadi, and R. Needham, "Logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[26] J.-L. Tsai, T.-C. Wu, and K.-Y. Tsai, "New dynamic ID authentication scheme using smart cards," *International Journal of Communication Systems*, vol. 23, no. 12, pp. 1449–1462, 2010.

[27] C.-C. Chang and C.-Y. Lee, "A secure single sign-on mechanism for distributed computer networks," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 629–637, 2012.

[28] M. Bellare and P. Rogaway, "The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs," in *Advances in Cryptology*, 2006.

Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
Advances in Multimedia
Advances in Civil Engineering
Hindawi
Submit your manuscripts at
www.hindawi.com
Journal of Control Science and Engineering
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration