Hindawi Publishing Corporation
Applied Computational Intelligence and Soft Computing
Volume 2016, Article ID 3916942, 10 pages
http://dx.doi.org/10.1155/2016/3916942

Research Article

An Efficient Chaotic Map-Based Authentication Scheme with Mutual Anonymity

Yousheng Zhou,[1,2] Junfeng Zhou,[1] Feng Wang,[3] and Feng Guo[4]

[1]*College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[2]*College of Computer Science, Chongqing University, Chongqing 400044, China*
[3]*College of Mathematical Sciences, Dezhou University, Dezhou 253023, China*
[4]*School of Electronic Engineering, Dublin City University, Dublin 9, Ireland*

Correspondence should be addressed to Yousheng Zhou; zhyousheng@gmail.com

Received 19 December 2015; Revised 7 March 2016; Accepted 17 March 2016

Academic Editor: Christian W. Dawson

A chaotic map-based mutual authentication scheme with strong anonymity is proposed in this paper, in which the real identity of the user is encrypted with a shared key between the user and the trusted server. Only the trusted server can determine the real identity of a user during the authentication, and any other entities including other users of the system get nothing about the user's real identity. In addition, the shared key of encryption can be easily computed by the user and trusted server using the Chebyshev map without additional burdensome key management. Once the partnered two users are authenticated by the trusted server, they can easily proceed with the agreement of the session key. Formal security analysis demonstrates that the proposed scheme is secure under the random oracle model.

1. Introduction

Due to its characteristic of sensibility of initial conditions and the chaotic parameter, a chaos system shows aperiodicity and pseudorandomness, and it has been widely used in many cryptographic constructions, such as chaotic system based hash functions [1–3], chaotic system based encryption [4–8], and chaotic based block cipher [9], and so forth.

Authentication and key agreement are the fundamental blocks used to achieve authenticity and confidentiality in cryptographic system. Much efforts on chaotic maps based authentication and key establishment have been made in recent years. In 2009, Han and Chang [10] proposed a chaotic map-based key agreement protocol, which removes the constraint of synchronization. However, Yoon and Yoo [11] pointed out that Han and Chang's [10] scheme cannot counter replay attack. Later, Tseng et al. [12] presented a chaotic map-based key agreement protocol for smart card-oriented application, which is vulnerable to internal attack and lacks perfect forward security as pointed out by Niu and Wang [13]. Though Niu and Wang [13] improved Tseng et al.'s [12] scheme and proposed a new one, it is expensive and cannot resist DoS attack. In addition, other researchers investigated the improvement for key agreement of smart card [14, 15]. Wang and Zhao [16] first proposed trusted third party (TTP) based key agreement scheme using the Chebyshev chaotic maps, which is improved by Yoon and Jeon [17] for its vulnerability to tampering attack. In 2012, Lai et al. [18] developed a novel TTP based key agreement protocol using the extended Chebyshev map, but their scheme cannot counter internal attack and off-line key guessing attack [19]. Later, Lee et al. [20] presented a mutual anonymous authentication scheme with the extended Chebyshev map, but it can incur the man-in-the-middle attack. Tan [21] proposed a novel authentication and key agreement protocol with smart card, which can achieve user anonymity; however, the cost consumption is expensive. To cut the heavy computation cost due to the smart card, Gong et al. [22] proposed an improved chaotic map-based key management scheme without a smart card. However, Wang and Luan [23] pointed out that Gong et al.'s scheme exists key management issues and potential security problems and then proposed a new secure key agreement protocol. In addition, some chaotic maps based schemes

[24–28] have been investigated for solving various security problems.

Although a lot of works on chaotic maps based authentication have been made, most of them cannot provide mutual authentication and are vulnerable to external attack. Only few schemes address this issue using encryption; however, the confidentiality of these schemes is not perfect, since internal users of the system can know the real identities of others during the execution of the authentication process. As the popularity of wireless communication enabled devices, the private information of users, such as identity and locations, can be easily illegally intercepted and then exploited to trace individuals by potential attackers [29]. The privacy of the user has attracted increasing attention from both industry and academia nowadays. To the best of our knowledge, a scheme can that addresses this privacy requirement does not exist. Motived by this, a mutual chaotic map-based authentication scheme with mutual anonymity is proposed in this paper, which has the following properties.

(1) Mutual Strong Anonymity. When user, Alice, in the system interacts with another user, Bob, to fulfill the authentication process, no entity except the trusted server can learn some information about the real identity of Alice and Bob. Furthermore, Alice and Bob cannot determine the opposite side as well; that is, Alice does not know Bob's real identity and vice versa.

(2) Untraceability. Any internal user cannot connect any two authentication sessions; that is, to say, even if a system user Alice has established a session with the same user Bob who was once authenticated, Alice still cannot determine that the opposite side is Bob using the historic session. In addition, any external entities cannot determine whether users in one session are the similar to users in another session using the intercepted messages.

The rest of the paper is organized as follows; some related basics and definitions are introduced in Section 2. The concrete construction of the proposed scheme is illustrated in Section 3. Analysis and comparison are presented in Section 4. At last, the paper is concluded.

2. Preliminaries

This section introduces the common user requirements, the security requirements for mutual authentication, some basics about the Chebyshev chaotic map and its advantage, and the security definitions.

2.1. Requirements

2.1.1. User Requirements. Given that the authentication scheme to be constructed should be easy to use, the following user requirements need to be satisfied.

(1) Independency. The system should enable users to choose their seeds to produce the shared encryption/decryption keys independently, which means the user can encrypt the transferred messages with a distinctive key in a new authentication session without additional agreement with the trusted server in advance.

(2) Round-Optimization. When a user wants to authenticate another entity, the number of the interactive rounds should be minimized as much as possible, which is helpful to save computation and communication cost, meanwhile users' experiences will be enhanced as well.

(3) Anonymity. From the user perspective, his real identity needs protection and it should not be exposed to other entities except the trusted server.

2.1.2. Security Requirements. Since the objective of our proposed protocol is to provide a reliable and robust authentication mechanism to counter all possible outside and inside attacks, based on previous studies [21–25, 32, 33], we give the following critical requirements to provide secure authentication.

(1) Mutual Authentication. After the involved partnered two users finish the process of authentication, they should be convinced that the opposite user is an authentic one, not a forged one.

(2) Efficiency. Since the process of mutual authentication is on-line and the trusted server is required to support all authentication processes, the communication and computation costs should be as low as possible.

(3) Integrity. This means the involved entities can verify the integrity of received messages, which aims to detect possible damage to those messages.

(4) Confidentiality. After the authentication process, a session key should be produced for both partnered users to provide a secure communication, and it ensures forward secrecy as well.

Next, a brief introduction of the Chebyshev map and some related preliminaries [25, 31, 33] are given.

2.2. The Chebyshev Chaotic Maps

2.2.1. Definitions of Chebyshev Chaotic Maps

Definition 1. Let n be an integer, $x \in [-1, 1]$, and an n-order Chebyshev polynomial map $T_n(x) : [-1, 1] \to [-1, 1]$ is defined as follows:

$$T_n(x) = \cos(n * \arccos(x)). \quad (1)$$

According to the definition, the Chebyshev polynomial map can also be defined recursively as follows:

$$T_n(x) = 2 * x * T_{n-1}(x) - T_{n-2}(x), \quad (2)$$

where $T_0(x) = 1$ and $T_2(x) = x$, $n \geq 2$.

The Chebyshev polynomial map has the following two properties.

(1) Semigroup property is as follows:

$$T_r(T_s(x)) = \cos\left(r * \cos^{-1}\left(s * \cos^{-1}(x)\right)\right) = \cos\left(r * s * \cos^{-1}(x)\right) = T_{sr}(x) = T_s(T_r(x)), \quad (3)$$

where r, s are two integers, $x \in [-1, 1]$.

(2) Chaos property is as follows. When n is bigger than 1, an n-degree Chebyshev polynomial map $T_n(x) : [-1, 1] \to [-1, 1]$ has the constant measure $f^*(x) = 1/(\pi\sqrt{1-x^2})$ and positive Lyapunov exponent $\lambda = \ln n > 0$.

According to the periodicity of $y = \cos(x)$, there exist multiple x associated with the same y to make the equation hold. To improve the security of classic Chebyshev polynomial map, Zhang [33] gave a proof that the Chebyshev polynomial map still keeps the semigroup property over the interval $(-\infty, \infty)$, which is called *the extended Chebyshev chaotic maps* with the following definition:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \bmod P, \quad (4)$$

where $n \geq 2$, $x \in [-1, 1]$, and P is a big prime number. It can be easily found the following equation holds as well:

$$T_r(T_s(x)) = T_{sr}(x) = T_s(T_r(x)) \bmod P. \quad (5)$$

Definition 2 (discrete logarithm problem (DLP)). Given any two big integers x, y, find an integer s to make the equation $T_s(x) \equiv y$ hold.

Definition 3 (decisional Diffie-Hellman problem (DDH)). Given $T_r(x)$, $T_s(x)$, and $T_u(x)$, where r, s, and u are unknown, determine whether equation $T_{sr}(x) = T_u(x) \bmod P$ holds or not.

2.2.2. The Advantages of Using Chebyshev Chaotic Maps. As a chaotic system characterizes excellent properties of diffusion and confusion, it is widely used to design various cryptographic schemes. Our design aims to provide a secure efficient mutual authentication with strong anonymity, and this means encryption will be integrated to keep the confidentiality of the identities. However, the traditional public key cryptography schemes are not desirable to achieve it since the management of encryption key in these schemes produces heavy computational burden. Inspired by the excellent semigroup property, the extended Chebyshev chaotic map over the finite field is used to develop our protocol since the discrete logarithm problem and Diffie-Hellman problem are assumed to be intractable within polynomial time [21]. However, there are no hardness assumptions of the discrete logarithm problems or the Diffie-Hellman problems about the Chebyshev chaotic maps over the interval $[-1, 1]$ [34], so that it is still challenging to design a secure chaotic map-based key agreement protocol over the interval $[-1, 1]$. Meanwhile, with the Chebyshev chaotic map, our proposed based scheme enables the users and trusted server efficiently to generate the shared encryption key and agree session key without additional key management. Though there are some other types of chaos systems, only the extended Chebyshev chaotic map has the semigroup property and satisfies the requirements stated above. In addition, the Chebyshev map has good chaotic properties with mixture and ergodicity, and the chaotic sequences generated by the Chebyshev map have good statistical distribution characteristics as the mean is 0 [35]. Wang et al. [7, 8] pointed out that low dimension chaotic maps have degradation of dynamics in finite precision computations in computers; however, this issue can be addressed using appropriate implementation; for example, Liu et al. [36] proposed an analogue-digital mixed method to solve the dynamical degradation of digital chaotic system. Given the previous advantages, the extended Chebyshev chaotic map is used to construct mutual authentication with strong anonymity in this paper.

2.3. Security Definitions. Based on the attack model in literatures [37, 38], the security model of the proposed chaotic map based mutual authentication and key agreement with strong anonymity (CMASA) is defined in this section. In the model, the capability of the adversary $\mathcal{A}$ is defined by the following interactive game which consists of oracle queries and security assumptions.

$\mathcal{A}$ can join the game through issuing series of oracle queries to any participant from the entity set $\prod_U^i$ including the trusted server. During the interactive activities, $\mathcal{A}$ is assigned with some attacking capabilities to the authentication protocol. The communication channel is under the full control of $\mathcal{A}$, which means $\mathcal{A}$ can intercept, block, inject, delete, and modify any message transferred via this channel. The queries that $\mathcal{A}$ can issue are as follows.

Excute$(\prod_U^i)$. This query is designed to assign $\mathcal{A}$ with passive attacking capability. After the execution of this query, all the transferred messages produced by the honest parities will be output according to the definition of P.

Send$(\prod_U^i, M)$. This query is designed to simulate the situation that $\mathcal{A}$ has controlled the whole communication process. $\mathcal{A}$ can issue *Send* query on M to $\prod_U^i$, and the corresponding entity from $\prod_U^i$ will compute the results according to P and respond to $\mathcal{A}$.

Reveal$(\prod_U^i)$. This query is used to simulate the known key attacking. If it is a valid session, all the computed shared session keys by $\prod_U^i$ will be responded to and null will be responded to otherwise.

Corrupt$(\prod_U^i)$. This query is used to simulate that $\mathcal{A}$ corrupts entities from $\prod_U^i$. $\mathcal{A}$ can obtain the permanent password and real identification of $\prod_U^i$ with this query.

SymEnc$(\{E, D\}, k, \{M, C\})$. This query is designed to assign $\mathcal{A}$ with the capability of accessing the encryption oracle. In order to respond to $\mathcal{A}$ correctly, a list L_e is needed to setup and maintenance. Upon receiving the query *SymEnc*(E, k, M), first check if there exists some entry

$\{K, M, C\}$ in L_e. If yes, return C of the corresponding entry; otherwise, a random value C' will be returned. Meanwhile, a new tuple $\{K, M, C'\}$ will be added into L_e. Equivalently, for the decryption query $SymEnc(D, k, C)$, first check if there exists some entry $\{K, M, C\}$ in L_e, if yes, return M of the corresponding entry, and a random value M' will be returned, otherwise. Meanwhile, the new tuple $\{K, M', C\}$ will be added to L_e.

$Hash(M)$. This query is utilized to simulate hashing for $\mathcal{A}$. To respond to $\mathcal{A}$ effectively, a list L_h will be set up. Upon receiving the query on m from $\mathcal{A}$, firstly check if there already exists some entry $\{m, h\}$ in L_h. If yes, return the value h of the existing entry to $\mathcal{A}$. Otherwise, generate a random value h' as the response and add $\{m, h'\}$ to L_h at the same time.

$Test(\prod_U^i)$. This query is used to measure the semantic security of the session key K_S. If the entity of this session key U has already computed K_S with his partnered peer, return K_S to $\mathcal{A}$. Otherwise, null will be responded to. $\mathcal{A}$ can also issue a single *Test* query to $\prod_U^i$, and $\prod_U^i$ will make an unbiased toss $b \in (0, 1)$ to demine the response. If $b = 1$, return K_S to $\mathcal{A}$. Otherwise, return a random value.

Definition 4 (security of the session key (ASK-Secure)). In an adversary involved interactive game, the adversary $\mathcal{A}$ can arbitrarily issue Test query, where the response is the real session key or a random value. If $\mathcal{A}$ issued a Test query to an unauthorized entity, $\mathcal{A}$ would be responded with $\perp$. If $\mathcal{A}$ issued a Test query to a dishonest entity or the entity whose peer is dishonest, the corresponding real session key will be responded to. Otherwise, a random c from an unbiased coin toss is used to determine that the response is the real session key or a random value. $\mathcal{A}$ would guess the uncovered c through analyzing the response. Let the event $E = \{\mathcal{A}$ wins this game$\}$, and let $\mathrm{Adv}_P^{\mathrm{ASK}}(\mathcal{A})$ be the advantage that $\mathcal{A}$ wins the distinguishability of P. If $\mathrm{Adv}_P^{\mathrm{ASK}}(\mathcal{A})$ is negligible, then P is called ASK-Secure [37].

Definition 5 (security of symmetric encryption (OT-Secure)). One-time security of symmetric encryption (OT-Secure) [39] means that the indistinguishability of symmetric encryption under the passive attack can also be called find-guess security. Let $\pi = (E, D)$ be a symmetric encryption scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary of π, and then consider the following interactive game between π and $\mathcal{A}$.

(1) Choose $a \leftarrow$ KSP.

(2) Input 1^k to run $\mathcal{A}_1$. $\mathcal{A}_1$ outputs two distinctive messages $m_0, m_2 \in$ MSP and the state s.

(3) Choose $b \leftarrow R^{(0,1)}$ randomly and compute $c \leftarrow E_a(m_b)$.

(4) Input c, s and run $\mathcal{A}_2$, and then $\mathcal{A}_2$ outputs $b' \in [0, 1]$.

The advantage of $\mathcal{A}$ represents how far it will guess the right b with the possibility bigger than 1/2; that is $2\Pr[b = b'] - 1$. During the whole process of the game, $\mathcal{A}$ is passive; in other words, it cannot access any encryption or decryption oracle.

3. Concrete Construction

The detailed construction of the proposed scheme is presented in this section. For convenience, the descriptions of all symbols to be used are listed in Description of Symbols.

Suppose there exist three entities in our scheme, two system users A, B, who need to authenticate each other, and a trusted third party Tread. During the authentication, Tread will authenticate A and B using their submitted messages. If Tread identifies that A or B has been revoked, the authentication process will be terminated. The whole process of authentication consists of two stages, that is, registration and authentication including key establishment.

At the beginning of registration, A, B generate their passwords, respectively. They precompute passwords using a hash function and then submit them to Tread together with identifications and other related information. Upon receiving the registration queries from A and B, Tread will check the validity of the submitted information. If yes, the registering is successful and Tread would securely store the needed information locally. The authenticating can be launched by A or B, and then the process will be conducted through the following interactive steps.

3.1. Registration. A user can register using the following steps.

(1) Tread chooses two random numbers x, s and a big prime number P, then computes $T_s = T_s(x) \bmod P$, and publishes (x, T_s, P).

(2) User u chooses his PW_u and computes $H(H(PW_u) \parallel ID_u)$, and then sends $\{ID_u, H(PW_u), H(H(PW_u) \parallel ID_u)\}$ to Tread.

(3) Tread checks the validity of ID_u and $H(PW_u)$ using $H(H(PW_u) \parallel ID_u)$. If yes, it stores $\{ID_u, H(PW_u)\}$. Otherwise, user u fails to register in the system.

3.2. Mutual Authentication and Key Establishment. Users A and B can finish the authentication and establishment by following the steps shown in Figure 1.

(1) $A \rightarrow B$. A first chooses two numbers x_A, r_A randomly, and then computes $T_A = T_{r_A}(x)$, $N_A = x_A \oplus H(PW_A)$, and $K_{TA} = T_{r_A}(T_s) \bmod P$, where N_A denotes the temporary identification of A and K_{TA} denotes the shared session key between A and Tread. After that, A encrypts ID_A, x_A, and $H(N_A \parallel t_A \parallel T_A)$ using K_{TA}; that is $C_1 = E_{K_{TA}}(ID_A \parallel x_A \parallel H(N_A \parallel t_A \parallel T_A))$, where t_A is the timestamp of A. Next, A sends $M_1 = \{C_1, N_A, t_A, T_A\}$ to B.

(2) $B \rightarrow Tread$. Upon receiving M_1 from A, B first checks if $|t_B - t_A| < \Delta T$ holds or not, where t_B is the timestamp of B. If yes, it stores N_A temporarily. Then, it chooses x_B, r_B randomly and computes $T_B = T_{r_B}(x)$, $N_B = x_B \oplus H(PW_B)$, and $K_{TB} = T_{r_B}(T_s) \bmod P$, where N_B denotes the temporary identification of B and K_{TB} denotes the shared session key between B and Tread. After that, B encrypts ID_B, x_B, and

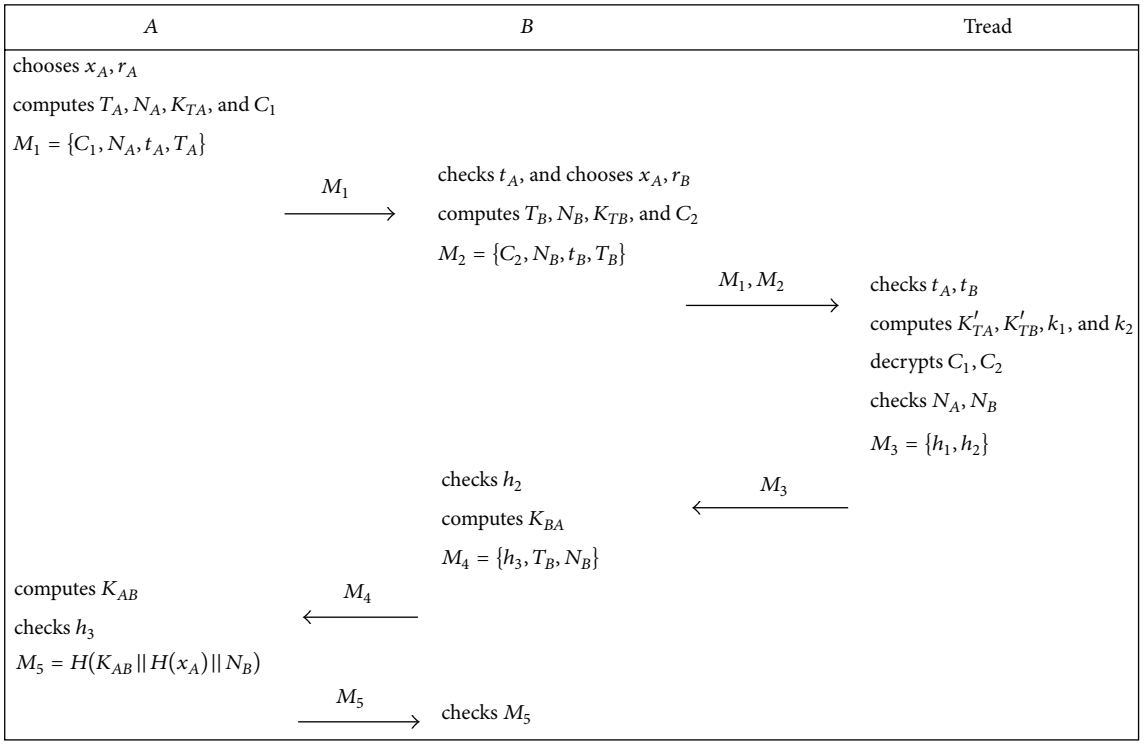

Figure 1: Process of authentication and key establishing.

$H(N_B \| t_B \| T_B)$ using K_{TB}, that is $E_{K_{TB}}(ID_B \| x_B \| H(N_B \| t_B \| T_B))$, where t_B is the timestamp of B. Next, B sends $M_1, M_2 = \{C_2, N_B, t_B, T_B\}$ to Tread.

(3) Tread $\rightarrow$ B. Upon receiving M_1, M_2 from B, Tread checks if $|T - t_A| < \Delta T, |T - t_B| < \Delta T$ holds or not, where T denotes the timestamp of Tread and ΔT denotes the permissible time interval threshold. If so, Tread will compute the shared keys $K'_{T_A} = T_s(T_A) \bmod P$, $K'_{T_B} = T_s(T_B) \bmod P$, $k_1 = H(N_A \| t_A \| T_A)$, and $k_2 = H(N_B \| t_B \| T_B)$, then it decrypts T_A and T_B using K'_{T_A} and K'_{T_B}. After that, Tread checks if $H(N_A \| t_A \| T_A) = k_1$, $H(N_B \| t_B \| T_B) = k_2$ hold or not. If yes, Tread validates N_A and N_B as follows.

Step 1. Search for $H(PW_A)$ and $H(PW_B)$ in the database.

Step 2. Compute $N_A \oplus H(PW_A)$ and $N_B \oplus H(PW_B)$. And then check if both $N_A \oplus H(PW_A) = x_A$ and $N_B \oplus H(PW_B) = x_B$ hold or not. If yes, go to Step 3; else it terminates.

Step 3. Compute $h_1 = H(x_A)$ and $h_2 = H(H(x_A) \| H(x_B))$, and then sends $M_3 = \{h_1, h_2\}$ to B.

(4) B $\rightarrow$ A. Upon receiving M_3 from A, B first computes $h'_2 = H(H(x_A) \| H(x_B))$ and then checks if $h'_2 = h_2$. If yes, B checks the temporary identification N_A of A. After that, B computes $K_{BA} = T_B(T_A) \bmod P$, $h_3 = H(K_{BA} \| H(x_A) \| N_B)$, where K_{BA} is the session key between B and A, and then sends $M_4 = \{h_3, T_B, N_B\}$ to A.

(5) A $\rightarrow$ B. Upon receiving M_4 from B, A first computes $K_{AB} = T_A(T_B) \bmod P$, $h'_3 = H(K_{AB} \| H(x_A) \| N_B)$, where K_{AB} is the session key between A and B and then checks if $h'_3 = h_3$. If yes, A confirms the temporary identification N_B of B and establishes the session key K_{AB}. Then A computes the hash value $M_5 = H(K_{AB} \| H(x_A) \| N_B)$ and sends it to B.

At last, B checks M_5 from A as follows: firstly, it computes $M'_5 = H(K_{BA} \| N_B)$ and then it checks if $M'_5 = M_5$. If it holds, the authentication is done.

4. Analysis

4.1. Security. The proof of the security consists of multiple interactive games, and it is based on the difference lemma [37], which is briefly reviewed as follows.

Lemma 6 (difference lemma). *Let E_a, E_b, and E_c be the events following some distribution. If $E_a \wedge \neg E_c \Leftrightarrow E_b \wedge \neg E_c$, then the following equation holds:*

$$\left|\Pr[E_a] - \Pr[E_b]\right| \leq \Pr[E_c]. \tag{6}$$

The proof of this lemma can be found in [37].

4.1.1. Security of Session Key. The security of session key for our proposed scheme is given by Theorem 7.

Theorem 7. *Let Adv_{Γ}^{OT} be the advantage that an OT adversary breaks the symmetric encryption within time t_1, and let ADV_G^{DDH} be the advantage that DDH adversary breaks DDH*

with time t_2. Then, the advantage that $\mathscr{A}$ breaks a ASK-secure mutual authentication scheme is

$$\begin{aligned} &Adv_{msaa}^{ask}\left(t', q_0, q_1, q_2, q_3\right) \\ &\leq \frac{\sum_{i=1}^{3} q_i^2}{2^{\lambda-1}} + \frac{2q_1 + 2q_2}{l} + 2Adv_{\Gamma}^{OT}\left(t_1, q_0, q_1, q_2, q_3\right) \\ &\quad + 2ADV_{G}^{DDH}\left(t_2, q_0, q_1, q_2, q_3\right), \end{aligned} \tag{7}$$

where $t' \leq t_1 + t_2 + (q_1 + q_2)\tau_1 + q_3(\tau_2 + \tau_3)$, q_0 is the times of Send queries, q_1, q_2 are the times of SymEnc queries of A to T and B to T respectively, q_3 is the times of Hash queries, l is the size of ID_u space, λ is the security parameter, and τ_1, τ_2, and τ_3 are the running time of single symmetric cryptographic operation, chaotic map operation, and Hash operation respectively.

Proof. To illustrate the proof, six interactive games G_i ($0 \leq i \leq 5$) are introduced. In every game G_i, $\mathscr{A}$ can arbitrarily issue any oracle queries defined in Section 2.3. When every game G_i is done, the possibility of event $E_i = \{\mathscr{A}$ wins $G_i\}$ can be captured.

Game G_0. This game depicts the attacking from $\mathscr{A}$ on MSAA in reality. According to the definition, the advantage should be as follows:

$$\mathrm{Adv}_{\mathrm{msaa}}^{\mathrm{OT}}(\mathscr{A}) = \left|2\Pr[E]_0 - 1\right|. \tag{8}$$

Game G_1. This game can simulate all oracle queries; the only difference is that guessing attack on real identification will be simulated as well. Since ID_A, ID_B will be encrypted by OT-Secure symmetric encryption, every value of $E_{K_{T_A}}(ID_A \parallel x_A \parallel H(N_A \parallel T_A \parallel T_{r_A}(x)))$, $E_{K_{T_B}}(ID_B \parallel x_B \parallel H(N_B \parallel T_B \parallel T_{r_B}(x)))$ should be distinctive. Therefore, $\mathscr{A}$ has no other auxiliary information to validate its guess on the real identification; that is to say, the success possibility is $(q_1 + q_2)/N$. According to the difference lemma [37], we can have

$$\left|\Pr[E_0] - \Pr[E_1]\right| \leq \frac{(q_1 + q_2)}{l}. \tag{9}$$

Game G_2. This game is the same as previous games except for the additional simulation of breaking symmetric encryption using *SymEnc*. According to the difference lemma, we can have

$$\left|\Pr[E_1] - \Pr[E_2]\right| \leq \frac{q_1 + q_2}{2^{\lambda}} + \mathrm{Adv}_{\Gamma}^{\mathrm{OT}}(\mathscr{A}). \tag{10}$$

Game G_3. This game is same as the previous games, except for the additional simulation of collusion attack to Hash. Game G_3 is indistinguishable against G_2 except for the possible collision in L_H. According to the Birthday Paradox and difference lemma, we can have

$$\left|\Pr[E_2] - \Pr[E_3]\right| \leq \frac{q_3^2}{2^{\lambda}}. \tag{11}$$

Game G_4. This game is same as the previous games except for the modification on the response of $T_a(x)$ mod P and $T_b(x)$ mod P on the *Send* query. Assume $(X, Y, Z) = (T_a(x), T_b(x), T_{ab}(x))$ is a random extended chaotic CDH triple. The simulator S will serve all oracle queries from all honest entities using (X, Y, Z). To do so, S firstly sets passwords for A and B and then responds as follows: it computes the chaotic maps $\{(a_0, T_{a_0}(x)), (b_0, T_{b_0}(x)), T_{a_0}(x), T_{b_0}(x)$, and $z_0 = T_{a_0 b_0}(x)\}$ and stores them in the list, where a_0, b_0 is random. For the Test query, it returns the stored z_0 as the response. In terms of the definition, the response for Test query is valid. Meanwhile, the random variable set in G_3 will be replaced by another identical distributed random variable set in G_4. Hence, the possibility that $\mathscr{A}$ wins G_4 and G_3 is the same, then we have

$$\Pr[E_3] = \Pr[E_4]. \tag{12}$$

Game G_5. This game simulates $\mathscr{A}$ breaking DDH. All the queries are same as the previous queries except that the response (X, Y, Z) is not a CDH triple, but a random triple $(T_u(x), T_v(x), T_w(x))$.

Assume that $\mathscr{A}_{\mathrm{DDH}}$ is a challenger who attempts to break the distinguishability of DDH over G, then $\mathscr{A}_{\mathrm{ask}}$ is an adversary who is capable of breaking the security of session key. $\mathscr{A}_{\mathrm{DDH}}$ responds to $c \in (0, 1)$ from the unbiased toss as follows. If $c = 1$, it returns the real session key to $\mathscr{A}_{\mathrm{ask}}$; else it returns a random number to $\mathscr{A}_{\mathrm{ask}}$. After that, $\mathscr{A}_{\mathrm{ask}}$ outputs its guess, c'. If $c' = c$, $\mathscr{A}_{\mathrm{ask}}$ wins this game. $\mathscr{A}_{\mathrm{DDH}}$ can respond for querying *Excute*, *Corrupt*, *SymEnc*, and *Test*; the process is the same as previous games except for the query on (X, Y, Z) as inputs. If $\mathscr{A}_{\mathrm{ask}}$ outputs c, $\mathscr{A}_{\mathrm{DDH}}$ outputs 1; otherwise, it outputs 0. If (X, Y, Z) is exactly a real CDH triple, then $\mathscr{A}_{\mathrm{DDH}}$ runs $\mathscr{A}_{\mathrm{ask}}$ in G_4, so we have $\Pr[\mathscr{A}_{\mathrm{DDH}}$ outputs $1] = \Pr[E_4]$. If (X, Y, Z) is a random triple, $\mathscr{A}_{\mathrm{DDH}}$ runs $\mathscr{A}_{\mathrm{ask}}$ in G_5, so we have $\Pr[\mathscr{A}_{\mathrm{DDH}}$ outputs $0] = \Pr[E_5]$. Thus,

$$\left|\Pr[E_4] - \Pr[E_5]\right| \leq \mathrm{Adv}_{G}^{\mathrm{DDH}}\left(\mathscr{A}_{\mathrm{DDH}}\right). \tag{13}$$

Since the session key Z_0 is random, the information about c does not leak, so we have

$$\Pr[E_5] = \frac{1}{2}. \tag{14}$$

According to formulas (8)–(14), the advantage can be evaluated as follows:

$$\begin{aligned} \mathrm{Adv}_{\mathrm{msaa}}^{\mathrm{ask}}\left(\mathscr{A}_{\mathrm{ask}}\right) \leq{}& \frac{\sum_{i=1}^{3} q_i^2}{2^{\lambda-1}} + \frac{2q_1 + 2q_2}{l} \\ &+ 2\mathrm{Adv}_{\Gamma}^{\mathrm{OT}}\left(t_1, q_0, q_1, q_2, q_3\right) \\ &+ 2\mathrm{Adv}_{G}^{\mathrm{DDH}}\left(t_2, q_0, q_1, q_2, q_3\right). \end{aligned} \tag{15}$$

□

4.1.2. Strong Anonymity for Client. To prevent the exposure of real identification during the message exchange, one practical solution is employing pseudonym. In the proposed scheme, if the adversary $\mathscr{A}$ attempts to obtain the real identification of a system user, the first possible step is to obtain the key to decrypt the cihpertext C_i even if $\mathscr{A}$ can intercept all the

Table 1: Comparison of security.

Scheme	Lee et al.'s [20]	Xie et al.'s [26]	Farash and Attari's [30]	Li et al.'s [31]	Lee et al.'s [25]	Niu and Wang's [32]	Ours
Forward secrecy	√	√	√	√	√	√	√
Backward secrecy	√	×	×	×	×	√	√
Resistance to the replay attack	√	√	√	√	√	√	√
Resistance to the MIM attack	×	√	√	√	√	√	√
Mutual anonymity	√	×	×	×	×	×	√
Strong anonymity	×	×	×	×	×	×	√

transferred messages. Though $\mathscr{A}$ possesses $T_{r_A}(x)$, $T_{r_B}(x)$, and T_s, he or she still faces the difficulty of solving DL problem if $\mathscr{A}$ tries to deduce the secret value from T_{AT} or T_{BT}. Since $\mathscr{A}$ cannot decrypt C_i, he or she cannot get to know the real identification, and then the privacy of users is preserved. For the entities who get involved in the authentication, they only get the temporary identification, which is generated by the XOR operation on the real identification and random number, so that they cannot know the real identification of the partnered peer. Even if he or she stores N_i for off-line analysis in future, N_i^* in the next session is generated by another distinctive random number, so N_i and N_i^* are indistinguishable for the PPT adversary $\mathscr{A}$. Furthermore, a system user entity even cannot determine whether the current partnered peer is the same with those in historic sessions or not. Thus, our proposed scheme achieves the strong anonymity successfully.

4.1.3. Resistance to Man-in-the-Middle Attack. Suppose there exists an active attacker $\mathscr{A}$ over the communication channel, who attempts to intercept and tamper the messages transferred via this channel to conduct the man-in-the-middle attack. If $\mathscr{A}$ tries to carry out the attack by tampering C_1, C_2, he or she will face the difficulty of solving DL problem. If $\mathscr{A}$ attempts to tamper or forge C_3, C_4, and C_5, he or she will face the difficulty of breaking the secure one-way hash function according to the definition of the protocol. Above all, the proposed protocol is secure enough to prevent the man-in-the-middle attack.

4.1.4. Resistance to Replay Attack. According to the construction of the presented protocol, all the transferred messages of A, B, and Tread use timestamps t_A, t_B to provide freshness. Furthermore, the system users have independently chosen (x_A, r_A) and (x_B, r_B) randomly to ensure freshness at the beginning of every authentication session. So, the proposed scheme can counter replay attack effectively.

4.1.5. Forward Secrecy. In our scheme, the forward secrecy means the previous used session key cannot be deduced even if adversary $\mathscr{A}$ is given the current session key and the password of the user. Actually, the establishment of the session key K_{AB} (or K_{BA}) between A and B is based on x_A and x_B chosen by themselves independently, and $\mathscr{A}$ cannot get anything about K_{AB} (or K_{BA}) because the randomness of x_A and x_B, and the success possibility will not increase even if PW_A and PW_B are given to the adversary.

4.1.6. Backward Secrecy. The backward secrecy of our scheme refers to the fact that even if the adversary $\mathscr{A}$ has obtained a client's password, all historic session keys, and current session key, could not finish authentication and key agreement. However, all the messages are transferred in anonymous way; thus, $\mathscr{A}$ cannot generate a valid message without knowing the real user identification according to the protocol, even if he or she is given the password PW. So, our scheme achieves the backward secrecy.

The overall comparison of security between our proposed scheme and the existing similar schemes is listed in Table 1.

All the schemes listed in Table 1 have employed random number in the construction, so they all can achieve the forward secrecy. Since only our scheme and work in [20] uncover the real identification, both schemes can ensure the backward secrecy. Subsequently, all the schemes in [25, 26, 30, 31] cannot provide mutual anonymity for the same reason. Although the scheme in [20] can uncover the real identification for the outside attacker, the authenticated peers can know the real identification of each other, so that it lacks the strong anonymity, and the scheme in [32] fails to protect the identity of the server because the identity of the user is transferred in plaintext during authentication, so it cannot provide strong anonymity. For the use of timestamp and random number, all the schemes in Table 1 can counter the replay attack. However, in the scheme of [20], the attacker can choose a random number x' and compute $R_A' = T_{x'}(r)$, and then he or she can finish the authentication successfully by blocking and injecting operation; thus, it is vulnerable to the man-in-the-middle attack.

4.2. Comparison of Performance. The overall performance comparison is listed in Table 2.

As the authentication is a sort of synchronized process, the total computational cost of the client and server in a whole authentication and key agreement should be investigated. Since the cost of XOR operation and module addition are much cheaper, these two operations are not included in comparison, and only symmetric encryption/decryption

Table 2: Comparison on performance.

Scheme	Client	Server
Lee et al. [20]	$6T_H + 6T_C$	$4T_H + 2T_C$
Xie et al.'s [26]	$6T_H + 6T_C$	$6T_H + 4T_C$
Farash and Attari's [30]	$8T_H + 6T_C$	$4T_H + 4T_C$
Li et al.'s [31]	$4T_H + 8T_C$	$5T_C$
Lee et al.'s [25]	$9T_H + 2T_E + 2T_D + 7T_C$	$4T_H + 2T_E + 2T_D + 2T_C$
Niu and Wang's [32]	$4T_H + 2T_E + 2T_D + 4T_C$	$2T_E + 2T_D$
Ours	$7T_H + 2T_E + 6T_C$	$3T_H + 2T_D + 2T_C$

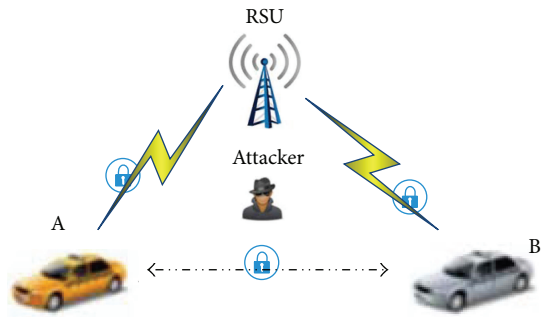

Figure 2: Authentication in VANETs.

operation, chaotic map operation, and hash operations are evaluated. Although no much advantage of performance is won in the proposed scheme, its critical privacy preserving feature deserves it.

4.3. Application Prospects. Our proposed scheme can be applied to privacy-sensitive situations, such as VANETs [29]. Consider an authentication scenario in VANETs as shown in Figure 2. Since the communication via wireless channel, the system is susceptible to attack from outside and inside adversaries. When the driver of vehicle A detects that another vehicle B nearby is sharing some resources, he becomes interested in using the application installed in his vehicle, he then issues a request of accessing the data. On one hand, for security, A is not allowed to access B's data directly, and B would firstly verify whether A is an authentic entity. However, A and B are unwilling to reveal their real identities to each other. Then, A and B have to proceed with a mutual anonymous authentication protocol. Meanwhile, A and B also want to keep themselves anonymous even if they authenticate each other again in the future, since few drivers like to expose their trace to other untrusted entities.

On the other hand, the real identities of A and B including the transferred messages should be kept confidential to any external entities, and any external attacker is not allowed to distinguish users from two different sessions using all intercepted messages. However, our proposed protocol can achieve all goals stated previously. Since the road side unit (RSU) is supposed to be trusted, then it can be regarded as the trusted server in our protocol, and then vehicle A and B can fulfill authentication via RSU by following the steps as defined in Section 3.

5. Conclusions

Of all the existing chaotic map based authentication schemes, most of them neglect the anonymity of the user. Since the privacy preservation in cryptographic systems has become a great concern nowadays, it is necessary to take the appropriate measures to address this problem. Thus, an extended Chebyshev chaotic map-based mutual authentication scheme with strong anonymity is investigated in this paper, in which the outside attacker, even the authenticated peers, cannot determine the real identity of others. The strong anonymity feature of the proposed scheme is suitable for the privacy sensible applications, such as mobile social networks, vehicle ad hoc networks.

Description of Symbols

ID_i:	Identification of user i
N_i:	Temporary identification of user i
$T_n(x)$:	The Chebyshev polynomial with degree n
T_s:	$T_s(x)$
T_A, T_B:	$T_{r_A}(x), T_{r_B}(x)$
x:	The initial value of chaotic map
s:	Private key of the trusted server
P:	A big prime number
x_i, r_i:	Random numbers chosen by users
K_{TA}, K_{TB}:	Session key shared between A, B, and Tread
$E(\cdot)/D(\cdot)$:	Symmetric encryption/decryption algorithm
t_A, t_B:	Timestamp
ΔT:	Threshold of interval
$H(\cdot)$:	A secure one-way hash function
$\oplus$:	XOR operation
PW_i:	Password of user i
T_H:	Running time for hash operation
T_E:	Running time for encryption operation
T_D:	Running time for decryption operation
T_C:	Running time for chaotic map operation.

Competing Interests

No potential conflict of interests was reported by the authors.

Acknowledgments

Our work was jointly supported by the National Social Science Foundation of China (no. 14CTQ026), the National Natural Science Foundation of China (no. 61272400 and no. 61472464), the Chongqing Research Program of Application Foundation and Advanced Technology (no. cstc2014jcyjA-40028 and no. cstc2013jcyjA40017), and the Natural Science Foundation of Shandong Province, China (no. ZR2015FL024).

References

[1] Y. Wang, X. Liao, D. Xiao, and K.-W. Wong, "One-way hash function construction based on 2D coupled map lattices," *Information Sciences*, vol. 178, no. 5, pp. 1391–1406, 2008.

[2] A. Akhavan, A. Samsudin, and A. Akhshani, "A novel parallel hash function based on 3D chaotic map," *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, article 126, pp. 1–12, 2013.

[3] H.-P. Ren and Y. Zhuang, "One-way hash function construction based on Chen-type hyper-chaotic system and key-stream," *Journal on Communications*, vol. 30, no. 10, pp. 100–113, 2009.

[4] T. Chen, L. Ge, J. Cai, and S. Ma, "TinyTCSec: a novel and lightweight data link encryption scheme for wireless sensor networks," *Chinese Journal of Sensors and Actuators*, vol. 24, no. 2, pp. 275–281, 2011.

[5] S. Chen, X. X. Zhong, and Z. Z. Wu, "Chaos block cipher for wireless sensor network," *Science in China, Series F: Information Sciences*, vol. 51, no. 8, pp. 1055–1063, 2008.

[6] T.-M. Chen and L. Ge, "Chaos-based encryption and message authentication algorithm for wireless sensor network," *Journal on Communications*, vol. 34, no. 5, pp. 113–120, 2013.

[7] Y.-Q. Zhang and X.-Y. Wang, "A symmetric image encryption algorithm based on mixed linear-nonlinear coupled map lattice," *Information Sciences*, vol. 273, pp. 329–351, 2014.

[8] X. Wang, D. Luan, and X. Bao, "Cryptanalysis of an image encryption algorithm using Chebyshev generator," *Digital Signal Processing*, vol. 25, no. 1, pp. 244–247, 2014.

[9] L. Xuan and J.-N. Yan, "The "one-group-one-cipher" cryptograph of block-cipher based on chaotic," *Journal on Communications A*, vol. 30, no. 11, pp. 105–110, 2009.

[10] S. Han and E. Chang, "Chaotic map based key agreement without clock synchronization," *Chaos, Solitons & Fractals*, vol. 39, no. 3, pp. 1283–1289, 2009.

[11] E.-J. Yoon and K.-Y. Yoo, "Replay attacks on Han et al.'s chaotic map based key agreement protocol using nonce," in *Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques*, D.-S. Huang, D. C. Wunsch II, D. S. Levine, and K.-H. Jo, Eds., vol. 15 of *Communications in Computer and Information Science*, pp. 533–540, Springer, Berlin, Germany, 2008.

[12] H.-R. Tseng, R.-H. Jan, and Y. Wuu, "A chaotic maps-based key agreement protocol that preserves user anonymity," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–6, IEEE, Dresden, Germany, June 2009.

[13] Y. Niu and X. Wang, "An anonymous key agreement protocol based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 4, pp. 1986–1992, 2011.

[14] K. Xue and P. Hong, "Security improvement of an anonymous key agreement protocol based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 7, pp. 2969–2977, 2012.

[15] C.-C. Lee, C.-L. Chen, C.-Y. Wu, and S.-Y. Huang, "An extended chaotic maps-based key agreement protocol with user anonymity," *Nonlinear Dynamics*, vol. 69, no. 1-2, pp. 79–87, 2012.

[16] X. Wang and J. Zhao, "An improved key agreement protocol based on chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 12, pp. 4052–4057, 2010.

[17] E.-J. Yoon and I.-S. Jeon, "An efficient and secure Diffie–Hellman key agreement protocol based on Chebyshev chaotic map," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 6, pp. 2383–2389, 2011.

[18] H. Lai, J. Xiao, L. Li, and Y. Yang, "Applying semigroup property of enhanced Chebyshev polynomials to anonymous authentication protocol," *Mathematical Problems in Engineering*, vol. 2012, Article ID 454823, 17 pages, 2012.

[19] F. Zhao, P. Gong, S. Li, M. Li, and P. Li, "Cryptanalysis and improvement of a three-party key agreement protocol using enhanced Chebyshev polynomials," *Nonlinear Dynamics*, vol. 74, no. 1-2, pp. 419–427, 2013.

[20] C.-C. Lee, C.-T. Li, and C.-W. Hsu, "A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps," *Nonlinear Dynamics*, vol. 73, no. 1-2, pp. 125–132, 2013.

[21] Z. Tan, "A chaotic maps-based authenticated key agreement protocol with strong anonymity," *Nonlinear Dynamics*, vol. 72, no. 1-2, pp. 311–320, 2013.

[22] P. Gong, P. Li, and W. Shi, "A secure chaotic maps-based key agreement protocol without using smart cards," *Nonlinear Dynamics*, vol. 70, no. 4, pp. 2401–2406, 2012.

[23] X.-Y. Wang and D.-P. Luan, "A secure key agreement protocol based on chaotic maps," *Chinese Physics B*, vol. 22, no. 11, Article ID 110503, 2013.

[24] P. Zhen, G. Zhao, L. Min, and X. Li, "Optimized key agreement protocol based on chaotic maps," *Journal of Communications*, vol. 9, no. 5, pp. 398–403, 2014.

[25] C.-C. Lee, C.-T. Li, S.-T. Chiu, and Y.-M. Lai, "A new three-party-authenticated key agreement scheme based on chaotic maps without password table," *Nonlinear Dynamics*, vol. 79, no. 4, pp. 2485–2495, 2015.

[26] Q. Xie, B. Hu, and T. Wu, "Improvement of a chaotic maps-based three-party password-authenticated key exchange protocol without using server's public key and smart card," *Nonlinear Dynamics*, vol. 79, no. 4, pp. 2345–2358, 2015.

[27] J. Shu, "An authenticated key agreement protocol based on extended chaotic maps," *Acta Physica Sinica*, vol. 63, no. 5, pp. 500–507, 2014.

[28] M. S. Farash, M. A. Attari, and S. Kumari, "Cryptanalysis and improvement of a three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps," *International Journal of Communication Systems*, 2014.

[29] F. Qu, Z. Wu, F.-Y. Wang, and W. Cho, "A security and privacy review of VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 2985–2996, 2015.

[30] M. S. Farash and M. A. Attari, "An efficient and provably secure three-party password-based authenticated key exchange protocol based on Chebyshev chaotic maps," *Nonlinear Dynamics*, vol. 77, no. 1, pp. 399–411, 2014.

[31] C.-T. Li, C.-W. Lee, and J.-J. Shen, "A secure three-party authenticated key exchange protocol based on extended chaotic maps in cloud storage service," in *Proceedings of the International Conference on Information Networking (ICOIN '15)*, pp. 31–36, IEEE, January 2015.

[32] Y. Niu and X. Wang, "An anonymous key agreement protocol based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 4, pp. 1986–1992, 2011.

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

[34] K. Y. Cheong and T. Koshiba, "More on security of public-key cryptosystems based on Chebyshev polynomials," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 9, pp. 795–799, 2007.

[35] L. Cao, Y. Luo, S. Qiu, and J. Liu, "A perturbation method to the tent map based on Lyapunov exponent and its application," *Chinese Physics B*, vol. 24, no. 10, Article ID 100501, 2015.

[36] L. F. Liu, H. P. Hu, and Y. S. Deng, "An analogue–digital mixed method for solving the dynamical degradation of digital chaotic systems," *IMA Journal of Mathematical Control and Information*, vol. 32, no. 4, pp. 703–716, 2015.

[37] V. Shoup, "Sequences of games: a tool for taming complexity in security proofs," *IACR Cryptology ePrint Archive*, 2004.

[38] J. Xu, W.-T. Zhu, and D.-G. Feng, "An improved smart card based password authentication scheme with provable security," *Computer Standards and Interfaces*, vol. 31, no. 4, pp. 723–728, 2009.

[39] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," *Journal of Cryptology*, vol. 26, no. 1, pp. 80–101, 2013.