Nonlinear Dyn (2015) 80:1209–1220
DOI 10.1007/s11071-015-1937-0

ORIGINAL PAPER

Design and analysis of a chaotic maps-based three-party authenticated key agreement protocol

Xiong Li · Jianwei Niu · Saru Kumari · Muhammad Khurram Khan · Junguo Liao · Wei Liang

Received: 27 May 2014 / Accepted: 23 January 2015 / Published online: 3 February 2015

Abstract An authenticated key agreement protocol is a protocol for information security over insecure networks. Due to the excellent properties of chaotic system, chaos-related cryptography has received a certain development, and recently, researchers have presented some three-party authenticated key agreement protocols based on the chaotic maps. Unfortunately, most of the chaotic maps-based key agreement protocols use a password to achieve the key agreement, and this leads to some security loopholes. First, the server has to store a sensitive password table, and it would be dangerous if the server was compromised or the password table was leaked. Besides, the low-entropy passwords are vulnerable to some password-related attacks, such as insider attack and password guessing attacks. In this paper, we design a communication- and computation-efficient chaotic maps-based three-party authenticated key agreement protocol without password and clock synchronization, and formally analyze the security using Burrows–Abadi–Needham logic. In addition to the formal analysis, we also prove that the presented protocol is free from most of the common attacks, and compare the performance and functionality with other related protocols. The result of the analysis and comparisons demonstrate that our protocol is more efficient and practical for real applications.

Keywords Chaotic maps · Authentication · Key agreement · Information security

X. Li (✉) · J. Liao · W. Liang
School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China
e-mail: snnulizhihui@163.com; lixiongzhq@163.com

X. Li · J. Niu (✉)
State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, Beihang University, Beijing 100191, China
e-mail: niujianwei@buaa.edu.cn

S. Kumari
Department of Mathematics, Agra College, Dr. B. R. A. University, Agra 282002, Uttar Pradesh, India

M. K. Khan
Center of Excellence in Information Assurance, King Saud University, Riyadh 11653, Saudi Arabia

1 Introduction

A chaotic system is characterized by pseudo-randomness and sensitivity to initial conditions that an arbitrarily small change or perturbation of the current trajectory may lead to significantly different future behavior [1]. These characteristics of the chaotic system meet the diffusion and confusion principles for the design of cryptography. Besides, chaotic system has been proved to be suitable for the design of secret key cryptosystems, such as symmetric encryption algorithms [2,3], and hash functions [4].

As a kind of important cryptographic protocols, authenticated key agreement protocols find applications in providing confidential communications over

the open networks. Since the same secret key is used for both encryption and decryption in symmetric cryptography, the generation and sharing of the common key is a challenge. An authenticated key agreement protocol allows the communication entities to agree upon a shared session key, and then, the following security communications can be protected using the agreed session key. So far, many public key cryptography-based authenticated key agreement protocols [5–10] have been presented by researchers. Recently, due to the excellent properties and better performance of the Chebyshev chaotic maps, many chaotic maps-based key agreement protocols [11–29] and user authentication schemes [30–32] have been proposed by researchers. Xiao et al. [11] designed a key agreement protocol using the semi-group property of Chebyshev chaotic maps. Unfortunately, Han [12] pointed out that Xiao et al.'s protocol was not secure since an adversary can impede the user and the server to agree on a session key without gaining any private information from the communications of the two parties. Meanwhile, Xiang et al. [13] demonstrated that Xiao et al.'s scheme was easy to suffer from stolen verifier attack and off-line password guessing attack. Later, Xiao et al. [14] designed a new key agreement protocol which combined the chaotic maps with the timestamp, where the timestamp was used to keep the freshness of messages. Han and Chang [15] presented two chaotic maps-based key agreement protocol, where one was based on clock synchronization, the another had not clock synchronization. Tseng et al. [16] presented a key agreement protocol with user anonymity based on the merits of chaotic maps. However, Niu and Wang [17] have shown that Tseng et al.'s protocol cannot achieve the property of user anonymity, and it also suffered from man-in-the-middle attack. Hence, Niu and Wang proposed an enhanced protocol [17]. Later, Yoon [18] pointed out that Niu and Wang's protocol suffered from computational efficiency problems, and cannot resist the denial-of-service (DoS) attack. Further, Yoon proposed an enhanced protocol. Xue and Hong [19] also pointed out the weaknesses of Niu and Wang's scheme, and proposed an improved protocol. Unfortunately, Tan [20] showed that Xue and Hong's scheme cannot provide strong user anonymity and vulnerable to man-in-the-middle attack. Very recently, Lee et al. [21] proposed an extended key agreement protocol using chaotic maps. However, He et al. [22] found out that Lee et al.'s scheme was vulnerable to insider attack, DoS attack, and cannot provide user anonymity.

All the above key agreement protocols using chaotic maps are designed for client–server environment, and all of these protocols had some drawbacks. Most of these protocols were based on clock synchronization, but it was a hard problem in itself to ensure the clocks to be synchronized in the full network; in order to achieve key agreement between any two clients, any two parties have to share a secret key in advance, and it was also difficult to be realized in a large-scale network. Wang and Zhao [23] presented a chaotic maps-based three-party key agreement protocol. However, Yoon and Jeon [24] stated that Wang and Zhao's protocol needed clock synchronization and suffered from an illegal message modification attack, and they proposed an enhanced protocol. Based on the review of these protocol, we find that both Wang–Zhao protocol [23] and Yoon-Jeon protocol [24] were not the three-party key agreement protocol. In their scheme, the Trent was assumed as a reliable third party in the network, and each user had to pre-share a secret key with the Trent, therefore, the key management of the Trent becomes a serious problem. Lai et al. [25] presented a password-based three-party key agreement protocol using the enhanced Chebyshev polynomials. Unfortunately, their protocol cannot resist off-line password guessing and the privileged insider attack [26]. Thus, Zhao [26] proposed an enhanced chaotic maps-based three-party key agreement protocol with the property of user anonymity. Very recently, Xie et al. [27] proposed a chaotic map-based three-party password-authenticated key agreement protocol. All the above three protocols use the password to achieve the key agreement, and then, the server had to store a fragile password table in the storage. Chen et al. [10] pointed out three flaws which should be improved in three-party authenticated key agreement protocols. He suggested that the password table should be removed else it would deduce some problems in case the password table was leaked or the server was compromised. Besides, the low-entropy passwords easily suffer from password-related attacks, such as insider attack and password guessing attacks. In this paper, we will propose a computation- and communication-efficient three-party authenticated key agreement protocol using chaotic maps, where the server does not need to store a password table, and it is free from the password-related attacks.

The remaining sections of the paper are arranged as follows. Some preliminaries about the Chebyshev chaotic maps and the related intractable problems are introduced in Sect. 2. Section 3 proposes a computation- and communication-efficient chaotic maps-based three-party authenticated key agreement protocol without password. The analysis of the proposed protocol and its comparison with other related protocols are described in Sects. 4 and 5, respectively. At last, we summarize the whole paper in Sect. 6.

2 Chebyshev chaotic maps

In order to facilitate understanding the protocol proposed in this paper, we first introduce some preliminaries about the Chebyshev chaotic maps, the chaotic maps-based discrete logarithm problem (DLP) and Diffie–Hellman problem (DHP).

Definition 1 Let n be an integer, and x be a variable belonging to the interval $[-1, 1]$. The Chebyshev polynomial $T_n(x) : [-1, 1] \rightarrow [-1, 1]$ is defined as $T_n(x) = \cos(n \cdot \arccos(x))$.

According to the Definition 1, the Chebyshev polynomial satisfies the following recursive relationship: $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), n \geq 2$, where the initial conditions $T_0(x) = 1$ and $T_1(x) = x$, and the first few Chebyshev polynomials are:

$$T_2(x) = 2x^2 - 1,$$
$$T_3(x) = 4x^3 - 3x,$$
$$T_4(x) = 8x^4 - 8x^2 + 1,$$
$$T_5(x) = 16x^5 - 20x^3 + 5x.$$

Definition 2 The chaotic property of Chebyshev polynomials. When $n > 1$, the Chebyshev polynomial map $T_n(x) : [-1, 1] \rightarrow [-1, 1]$ of degree n is a chaotic map with invariant density $f^*(x) = 1/(\pi\sqrt{1-x^2})$ for positive Lyapunov exponent $\lambda = \ln n > 0$.

Definition 3 Semi-group property of Chebyshev polynomials.

$$\begin{aligned} T_r(T_s(x)) &= \cos(r\cos^{-1}(\cos(s\cos^{-1}(x)))) \\ &= \cos(rs\cos^{-1}(x)) \\ &= T_{sr}(x) \\ &= T_s(T_r(x)) \end{aligned}$$

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

Zhang [33] further demonstrated that the semi-group property holds for Chebyshev polynomials on the interval $(-\infty, +\infty)$, which can enhance the property as follows: $T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x)) \bmod p$, where $n \geq 2, x \in (-\infty, +\infty)$ and p is a large prime number.

Therefore, $T_r(T_s(x)) \equiv T_{rs}(x) \equiv T_{sr}(x) \equiv T_s(T_r(x)) \bmod p$.

Definition 4 The discrete logarithm problem (DLP): DLP is to find an integer r such that $T_r(x) = y$ under the condition of knowing the parameters x and y.

Definition 5 The Diffie–Hellman problem (DHP): DHP is to compute the value $T_{rs}(x)$ under the condition of knowing the parameters x, $T_r(x)$ and $T_s(x)$.

It is generally believed that these two problems are intractable, i.e., there are no polynomial-time algorithms to solve these problems with non-negligible probability, and it provides the possibility to design information security protocols using these two intractable problems.

3 The proposed protocol

In this section, we propose a new three-party authenticated key agreement protocol without password using the excellent properties of the chaotic maps. We describe the notations used in our protocol in Table 1. There are three parties in the proposed protocol, i.e., user A, user B and server S, where S is supposed to be a trusted third party responsible for system initialization. Besides, the server S helps users to achieve key agreement. The proposed protocol includes two phases:

Table 1 The notations used in this paper

Notation	Description
A, B	The user A and B
S	The trusted third party
ID_A, ID_B	The identities of user A and B, respectively
x	The master secret key of server S
X	The seed of Chebyshev chaotic maps
SK	The session key shared between A and B
$E_k(\cdot)$	A secure symmetric encryption algorithm under the control of the secret key k
$D_k(\cdot)$	A secure symmetric decryption algorithm under the control of the secret key k
$h(\cdot)$	A secure one-way hash function
$\|$	The message concatenation operation

initialization phase and authenticated key agreement phase.

3.1 The initialization phase

In this phase, the server S initializes the system by choosing the system parameters, and the users should register himself/herself to the server S. We suppose that the channel between the server S and the user is secure in the initialization phase.

First, the server S chooses a random key x as the master secret key and generates a public seed X of Chebyshev polynomial, where the binary length of x must be long enough to resist guessing attacks, such as 1,024 bits. Then, the server chooses a secure chaotic maps-based hash function $h(\cdot)$ [34], and the secure symmetric encryption/decryption algorithm $E_k(\cdot)/D_k(\cdot)$.

When the user A (B) wants to agree on a session key with other users with the help of the server S, he/she needs to register himself/herself to the server S at first. The user A chooses the identity ID_A, and submits ID_A to the server S and request for registration. The server computes $h(ID_A\|x)$ and transmits it to the user A through the secure channel, then the user becomes a registered and valid user of the system. A stores $h(ID_A\|x)$ as a secret and does not reveal it to any other party. Similarly, the other users such as B can register himself/herself to the server S using identity ID_B, and he/she will receive a secret information $h(ID_B\|x)$.

3.2 Authentication and key agreement phase

When the users A and B have registered as the legal user of the system, they can agree on a session key SK with the aid of the trusted server S by performing the authentication and key agreement phase. We describe this phase in detail as follows, and we also illustrate it in Fig. 1.

Step 1 $A \rightarrow B : \{ID_A, C_1\}$

The user A generates a nonce a and computes

$K_A = T_a(X)$,

$H_A = h(ID_A\|ID_B\|K_A)$,

$C_1 = E_{h(ID_A\|x)}(ID_A\|ID_B\|K_A\|H_A)$.

Then, the user A transmits $\{ID_A, C_1\}$ to the user B.

Step 2 $B \rightarrow S : \{ID_A, C_1, ID_B, C_2\}$

When receiving the message $\{ID_A, C_1\}$ from the user A, the user B generates a nonce b and computes

$K_B = T_b(X)$,

$H_B = h(ID_B\|ID_A\|K_B)$,

$C_2 = E_{h(ID_B\|x)}(ID_B\|ID_A\|K_B\|H_B)$.

Then, the user B submits $\{ID_A, C_1, ID_B, C_2\}$ to the server S.

Step 3 $S \rightarrow B : \{C_3, C_4\}$

Upon receiving the message $\{ID_A, C_1, ID_B, C_2\}$ from the user B, the server S checks whether the ID_A and ID_B are valid identities of the registered users in the system. If so, S computes $D_{h(ID_A\|x)}(C_1) = (ID_A, ID_B, K_A, H_A)$, $D_{h(ID_B\|x)}(C_2) = (ID_B, ID_A, K_B, H_B)$, and the server S can check the validity of the identities ID_A and ID_B again. Then, the server S computes $H'_A = h(ID_A\|ID_B\|K_A)$, $H'_B = h(ID_B\|ID_A\|K_B)$, and checks whether $H'_A = H_A$ and $H'_B = H_B$. If they are both valid, S computes $H_{SAB} = h(ID_A\|ID_B\|K_A\|K_B)$, $C_3 = E_{h(ID_B\|x)}(ID_B\|ID_A\|K_B\|K_A\|H_{SAB})$, $C_4 = E_{h(ID_A\|x)}(ID_A\|ID_B\|K_A\|K_B\|H_{SAB})$. At last, the server S submits the message $\{C_3, C_4\}$ to the user B.

Step 4 $B \rightarrow A : \{C_4, H_{BA}\}$

When gets the message $\{C_3, C_4\}$ submitted from server S, the user B calculates $D_{h(ID_B\|x)}(C_3) = (ID_B, ID_A, K_B, K_A, H_{SAB})$, $H'_{SAB} = h(ID_A\|ID_B\|K_A\|K_B)$, and checks whether H'_{SAB} equals to the decrypted H_{SAB}. If the equation is hold, the validity of the server S is verified by the user B. Then, the user B computes $SK = T_b(K_A) = T_{ab}(X)$, $H_{BA} = h(ID_B\|ID_A\|SK\|K_A)$, and submits $\{C_4, H_{BA}\}$ to the user A.

Step 5 $A \rightarrow B : \{H_{AB}\}$

After receiving the message $\{C_4, H_{BA}\}$ from the user B, the user A computes $D_{h(ID_A\|x)}(C_4) = (ID_A, ID_B, K_A, K_B, H_{SAB})$, $H''_{SAB} = h(ID_A\|ID_B\|K_A\|K_B)$, and checks whether H''_{SAB} equals to the decrypted H_{SAB}. If so, the validity of the server S is affirmed by the user A. Then, the user A computes $SK' = T_a(K_B) = T_{ab}(X)$, $H'_{BA} = h(ID_B\|ID_A\|SK'\|K_A)$, and checks whether $H'_{BA} = H_{BA}$. If the equation holds, the validity of the user B is confirmed by the user A. At last, the user A computes $H_{AB} = h(ID_A\|ID_B\|SK'\|K_B)$, and submits $\{H_{AB}\}$ to the user B.

Step 6 When obtaining the message $\{H_{AB}\}$ from A, the user B computes $H'_{AB} = h(ID_A\|ID_B\|SK\|K_B)$, and checks whether H'_{AB} equals to H_{AB}. If they are equal,

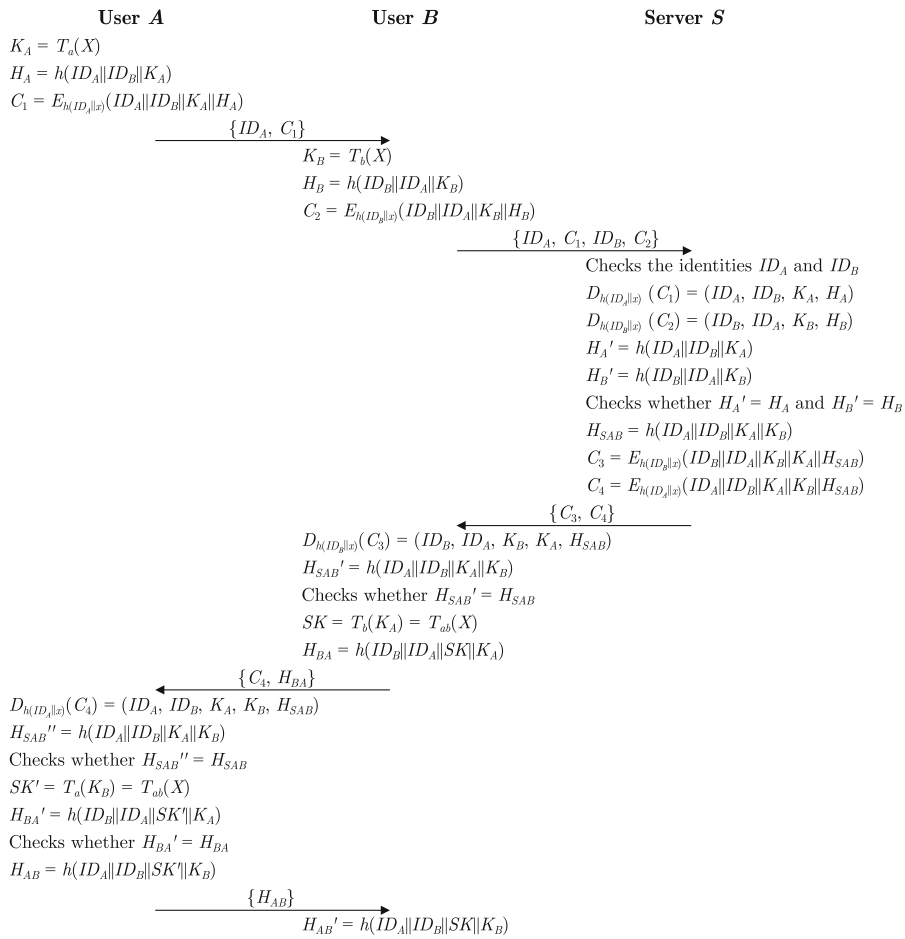

Fig. 1 The proposed protocol

the validity of the user A is authenticated by the user B. At last, a session key $SK = T_b(K_A) = T_bT_a(X) = T_{ab}(X) = T_aT_b(X) = T_a(K_B) = SK'$ is established between the user A and the user B.

4 Security analysis

We analyze the security properties of the presented protocol in this section. First, we use Burrows–Abadi–Needham logic (i.e., BAN logic) [35] to prove that the session key SK is correctly generated between the user A and the user B in this protocol. Next, we demonstrate the proposed three-party authenticated key agreement protocol without password using chaotic maps can resist various kinds of attacks and can achieve many ideal functions.

4.1 Authentication proof based on BAN logic

The BAN logic [35] is a well-known formal analysis mean for the security analysis of cryptographic protocols, and it plays a very important role in formal security analysis of the authentication and session key agreement protocols. We first show some notations and rules about BAN logic as follows.

$P \mid\equiv X$: P believes X, i.e., P believes X is true.
$P \lhd X$: P sees X, i.e., someone sent a message which contains X to P, and P can read X.
$P \mid\sim X$: P once said X, i.e., entity P once sent a message which contains X in some time.
$P \Rightarrow X$: P controls X, i.e., X is subject to the jurisdiction of entity P, and P is trusted for X.

$\#(X)$: X is fresh, i.e., no entity sent a message containing X at any time before the current round of protocol.
$P \overset{K}{\longleftrightarrow} Q$: P and Q can communicate with each other using the common shared key K, where K is called good if any other entity cannot get K except P, Q and the entity who is trusted by P or Q.
(X, Y) : X or Y is a part of message (X, Y).
$\{X, Y\}_K$: X and Y are encrypted under the control of the key K.
$(X, Y)_K$: X and Y are hashed under the control of the key K.

Rule 1 Message-meaning rule: $\frac{P|\equiv P \overset{K}{\longleftrightarrow} Q, P \lhd \{X\}_K}{P|\equiv Q|\sim X}$, if P believes he/she shares the key K with Q, and P sees the message $\{X\}_K$, P trusts that Q once said X.
Rule 2 Nonce-verification rule: $\frac{P|\equiv \#(X), P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}$, if P believes X is fresh and Q once said X, P believes Q believes X.
Rule 3 Jurisdiction rule: $\frac{P|\equiv Q \Rightarrow X, P|\equiv Q|\equiv X}{P|\equiv X}$, if P believes that Q had jurisdiction right to X and believes Q believes X, P believes X.
Rule 4 Freshness-conjuncatenation rule: $\frac{P|\equiv \#(X)}{P|\equiv \#(X,Y)}$, if X is a part of message (X, Y) and X is fresh, message (X, Y) is also fresh.
Rule 5 Belief rule: $\frac{P|\equiv Q|\equiv (X,Y)}{P|\equiv Q|\equiv (X)}$, if P believes Q believes the message set (X, Y), P also believes Q believes the message X.

Although the BAN logic has its own limitations, it is being widely used in analyzing the correctness of authentication protocol with key agreement. Protocol correctness means that both of the communication parties confirm they are sharing a fresh session key with each other after the protocol execution, i.e., the session key agreement protocol should achieve the following goals:

Goal 1: $A \mid\equiv (A \overset{SK}{\longleftrightarrow} B)$.
Goal 2: $A \mid\equiv B \mid\equiv (A \overset{SK}{\longleftrightarrow} B)$.
Goal 3: $B \mid\equiv (A \overset{SK}{\longleftrightarrow} B)$.
Goal 4: $B \mid\equiv A \mid\equiv (A \overset{SK}{\longleftrightarrow} B)$.

At first, we idealize the communication messages of the proposed protocol to facilitate the analysis, and the idealized messages are listed as below:

Message 1: $A \rightarrow S : \{ID_B, K_A, H_A\}_{h(ID_A \| x)}$.
Message 2: $B \rightarrow S : \{ID_A, K_B, H_B\}_{h(ID_B \| x)}$.
Message 3: $S \rightarrow A : \{ID_A, ID_B, K_A, A \overset{K_B}{\longleftrightarrow} B, H_{SAB}\}_{h(ID_A \| x)}$.
Message 4: $S \rightarrow B : \{ID_B, ID_A, K_B, A \overset{K_A}{\longleftrightarrow} B, H_{SAB}\}_{h(ID_B \| x)}$.
Message 5: $B \rightarrow A : (ID_B, ID_A, A \overset{SK}{\longleftrightarrow} B, K_A)_{SK}$.
Message 6: $A \rightarrow B : (ID_A, ID_B, A \overset{SK}{\longleftrightarrow} B, K_B)_{SK}$.

Second, we make some initial state assumptions based on the proposed protocol as follows:

$A_1 : A \mid\equiv \#(a)$.
$A_2 : B \mid\equiv \#(b)$.
$A_3 : A \mid\equiv \#(K_A)$.
$A_4 : B \mid\equiv \#(K_B)$.
$A_5 : A \mid\equiv A \overset{h(ID_A \| x)}{\longleftrightarrow} S$.
$A_6 : S \mid\equiv A \overset{h(ID_A \| x)}{\longleftrightarrow} S$.
$A_7 : B \mid\equiv B \overset{h(ID_B \| x)}{\longleftrightarrow} S$.
$A_8 : S \mid\equiv B \overset{h(ID_B \| x)}{\longleftrightarrow} S$.
$A_9 : A \mid\equiv S \Rightarrow (A \overset{K_B}{\longleftrightarrow} B)$.
$A_{10} : B \mid\equiv S \Rightarrow (A \overset{K_A}{\longleftrightarrow} B)$.

A_1 and A_2 mean that A and B generate fresh random numbers a and b, respectively. Therefore, they assure their freshness, respectively. Since $K_A = T_a(X)$ and $K_B = T_b(X)$, A_3 and A_4 are valid according to A_1 and A_2. A_5 and A_6 are valid because the secret key $h(ID_A \| x)$ is possessed by user A and can be computed by server S, and A_7 and A_8 are valid with the same reason. Because server S is a trusted party, once A decrypts C_4 and gets K_B, then A believes S controls K_B, and A_9 is valid. A_{10} is valid because S is trusted and B can get K_A by decrypting C_3.

Next, based on the rules of the BAN logic, we prove the proposed protocol can achieve the intended goals using the initial assumptions, and the detailed descriptions are as below:

According to the message 3, we can get

$S_1 : A \lhd \{ID_A, ID_B, K_A, A \overset{K_B}{\longleftrightarrow} B, H_{SAB}\}_{h(ID_A \| x)}$.

According to S_1 and A_5, we apply the message-meaning rule and can obtain

$S_2 : A \models S \mid\sim (ID_A, ID_B, K_A, A \overset{K_B}{\longleftrightarrow} B, H_{SAB})$.

According to A_3, we employ the freshness-conjuncatenation rule and can obtain

$S_3 : A \mid\equiv \#(ID_A, ID_B, K_A, A \overset{K_B}{\longleftrightarrow} B, H_{SAB})$.

According to S_2 and S_3, we apply nonce-verification rule to get

$$S_4 : A \models S \mid\equiv (ID_A, ID_B, K_A, A \overset{K_B}{\longleftrightarrow} B, H_{SAB}).$$

According to S_4, we apply belief rule to obtain

$$S_5 : A \mid\equiv S \mid\equiv (A \overset{K_B}{\longleftrightarrow} B).$$

According to A_9 and S_5, we apply jurisdiction rule to obtain

$$S_6 : A \mid\equiv (A \overset{K_B}{\longleftrightarrow} B).$$

According to $SK = T_a(K_B) = T_{ab}(X)$, we can get

$$S_7 : A \mid\equiv (A \overset{SK}{\longleftrightarrow} B). \quad \text{(Goal 1)}$$

According to message 4, we can get

$$S_8 : B \lhd \{ID_B, ID_A, K_B, A \overset{K_A}{\longleftrightarrow} B, H_{SAB}\}_{h(ID_B \| x)}.$$

According to S_8 and A_7, we employ the message-meaning rule and can obtain

$$S_9 : B \mid\equiv S \mid\sim (ID_B, ID_A, K_B, A \overset{K_A}{\longleftrightarrow} B, H_{SAB}).$$

According to A_4, we apply freshness-conjuncatenation rule to obtain

$$S_{10} : B \mid\equiv \#(ID_B, ID_A, K_B, A \overset{K_A}{\longleftrightarrow} B, H_{SAB}).$$

According to S_9 and S_{10}, we apply nonce-verification rule to get

$$S_{11} : B \models S \mid\equiv (ID_B, ID_A, K_B, A \overset{K_A}{\longleftrightarrow} B, H_{SAB}).$$

According to S_{11}, we apply belief rule to obtain

$$S_{12} : B \mid\equiv S \mid\equiv (A \overset{K_A}{\longleftrightarrow} B).$$

According to A_{10} and S_{12}, we apply jurisdiction rule to obtain

$$S_{13} : B \mid\equiv (A \overset{K_A}{\longleftrightarrow} B).$$

According to $SK = T_b(K_A) = T_{ab}(X)$, we can get

$$S_{14} : B \mid\equiv (A \overset{SK}{\longleftrightarrow} B). \quad \text{(Goal 3)}$$

According to message 5, we can get

$$S_{15} : A \lhd (ID_B, ID_A, A \overset{SK}{\longleftrightarrow} B, K_A)_{SK}.$$

According to S_{15} and S_7, we employ the message-meaning rule to obtain

$$S_{16} : A \mid\equiv B \sim (ID_B, ID_A, A \overset{SK}{\longleftrightarrow} B, K_A).$$

According to A_3, we apply freshness-conjuncatenation rule to obtain

$$S_{17} : A \mid\equiv \#(ID_B, ID_A, A \overset{SK}{\longleftrightarrow} B, K_A).$$

According to S_{16} and S_{17}, we employ the nonce-verification rule to obtain

$$S_{18} : A \mid\equiv B \mid\equiv (ID_B, ID_A, A \overset{SK}{\longleftrightarrow} B, K_A).$$

According to S_{18}, we apply belief rule to obtain

$$S_{19} : A \mid\equiv B \mid\equiv (A \overset{SK}{\longleftrightarrow} B). \quad \text{(Goal 2)}$$

According to message 6, we can get

$$S_{20} : B \lhd (ID_A, ID_B, A \overset{SK}{\longleftrightarrow} B, K_B)_{SK}.$$

According to S_{20} and S_{14}, we employ the message-meaning rule to get

$$S_{21} : B \mid\equiv A \sim (ID_A, ID_B, A \overset{SK}{\longleftrightarrow} B, K_B).$$

According to A_4, we apply freshness-conjuncatenation rule to obtain

$$S_{22} : B \mid\equiv \#(ID_A, ID_B, A \overset{SK}{\longleftrightarrow} B, K_B).$$

According to S_{21} and S_{22}, we employ the nonce-verification rule to get

$$S_{23} : B \mid\equiv A \mid\equiv (ID_A, ID_B, A \overset{SK}{\longleftrightarrow} B, K_B).$$

According to S_{23}, we apply belief rule to obtain

$$S_{24} : B \mid\equiv A \mid\equiv (A \overset{SK}{\longleftrightarrow} B) \quad \text{(Goal 4)}$$

According to S_7, S_{19}, S_{14}, S_{24}, we can clearly see that the presented protocol achieves Goal 1, Goal 2, Goal 3, Goal 4, and both of the user A and user B believe that they share a common session key $SK = T_{ab}(X)$ with each other. In the following subsections, we analyze other security aspects of the proposed protocol and prove that the proposed protocol can meet many kinds of functional features and can withstand various kinds of attacks.

4.2 Perfect forward secrecy

The perfect forward secrecy is one of most important security property of a key agreement protocol that the attacker still cannot calculate the previously established session keys even if he/she gets the server S's master secret key x. In the presented protocol, the session key $SK = T_{ab}(X)$ between the user A and user

B is only related to the random numbers a and b chosen by these two users, respectively. The attacker can eavesdrop C_1, C_2 from the public channel, if the master secret key x is obtained by the attacker, he/she can get $K_A = T_a(X)$ and $K_B = T_b(X)$ by decrypting C_1 and C_2, respectively. However, due to the intractable nature of the chaotic maps DLP and DHP, the attacker cannot get $SK = T_{ab}(X)$. Therefore, the presented protocol achieves the property of perfect forward secrecy.

4.3 Known-key security

A key agreement protocol satisfy the property of known-key security if the compromise of one session key does not cause the compromise of other session keys. In the presented protocol, the session key $SK = T_{ab}(X)$ only relies on the random numbers a and b. Since these two random numbers of one session are independent of those of other session's, one session key is independent of those of other session's. Therefore, even if one session key $SK = T_{ab}(X)$ was compromised by an attacker, he/she cannot get the other session key $SK = T_{a'b'}(X)$ without knowing a' and b', and the proposed protocol can provide known-key security.

4.4 Key-compromise impersonation attack

If the attacker cannot impersonate as the user C to communicate with other users under the condition that he/she gets the secret key $h(ID_A\|x)$ of the user A, we say the key agreement protocol can resist key-compromise impersonation attack. In our presented protocol, when the secret key $h(ID_A\|x)$ is leaked to an attacker, he/she can imitate the user A to communicate with other users definitely. In order to impersonate as the user C to communicate with other users, the attacker has to get $h(ID_C\|x)$ to generate and reply the valid communication messages; however, the secret key $h(ID_C\|x)$ is independent in and has not any relation to $h(ID_A\|x)$. Therefore, the attacker cannot impersonate as the user C to communicate with other users when he/she obtains the secret key $h(ID_A\|x)$ of the user A, and the proposed protocol can resist this kind of attack.

4.5 Unknown key-share attack and mutual authentication

For a key agreement protocol, the unknown key-share attack is that a party A thinking he/she has established a session key with party B, but the party B mistakenly thinking the session key is instead agreed with a party $C \neq A$. In our presented protocol, the user A and user B can establish a session key only if the mutual authentication among the user A, user B and the server S is achieved, which ensures that the proposed protocol avoids of unknown key-share attack.

4.5.1 Mutual authentication between the user A and the server S

In step 3 of our presented protocol, when receiving the message $\{ID_A, C_1, ID_B, C_2\}$ from the user B, the server S first checks whether the ID_A is a valid registered user identity. If so, S computes $D_{h(ID_A\|x)}(C_1) = (ID_A, ID_B, K_A, H_A)$ and the server S can check the validity of the identity ID_A again. Next, the server S calculates $H'_A = h(ID_A\|ID_B\|K_A)$ and checks $H'_A \stackrel{?}{=} H_A$. If they are equal, the validity of the user A is authenticated by the server S. In contrast, in step 5 of the proposed protocol, after receiving the message $\{C_4, H_{BA}\}$ from the user B, user A computes $D_{h(ID_A\|x)}(C_4) = (ID_A, ID_B, K_A, K_B, H_{SAB})$, $H''_{SAB} = h(ID_A\|ID_B\|K_A\|K_B)$, and checks $H''_{SAB} \stackrel{?}{=} H_{SAB}$. If the equation holds, the validity of the server S is verified by the user A. Because the exchanged messages between A and S are encrypted or decrypted by the user A's secret key $h(ID_A\|x)$, any other entity cannot forge valid messages without knowing the secret $h(ID_A\|x)$. Therefore, the user A and the server S can authenticate each other in the presented protocol.

4.5.2 Mutual authentication between the user B and the server S

In step 3 of the proposed protocol, when receiving the message $\{ID_A, C_1, ID_B, C_2\}$ from the user B, the server S first checks whether the ID_B is a valid registered user identity. If so, S computes $D_{h(ID_B\|x)}(C_2) = (ID_B, ID_A, K_B, H_B)$ and the server S can check the validity of the identity ID_B again. Then, the server S computes $H'_B = h(ID_B\|ID_A\|K_B)$, and checks $H'_B \stackrel{?}{=} H_B$. If the equation holds, the validity of the user B is verified by the server S. In contrast, in step 4 of our presented protocol, when receiving the message $\{C_3, C_4\}$ from the server S, the user B computes $D_{h(ID_B\|x)}(C_3) = (ID_B, ID_A, K_B, K_A, H_{SAB})$, $H'_{SAB} = h(ID_A\|ID_B\|K_A\|K_B)$, and checks $H'_{SAB} \stackrel{?}{=} H_{SAB}$. The

user B can affirm that the server S is valid if the equation is hold. Because the messages exchanged between the user B and the server S are encrypted or decrypted by user B's secret key $h(ID_B\|x)$, any other entity cannot forge valid messages without knowing the secret $h(ID_B\|x)$. Therefore, the user B and the server S can authenticate each other in the presented protocol.

4.5.3 Mutual authentication between the user A and the user B

In step 5 of our presented protocol, when gets the message $\{C_4, H_{BA}\}$ from user B, user A computes $D_{h(ID_A\|x)}(C_4) = (ID_A, ID_B, K_A, K_B, H_{SAB}), SK' = T_a(K_B) = T_{ab}(X), H'_{BA} = h(ID_B\|ID_A\|SK'\|K_A)$, and checks whether $H'_{BA} = H_{BA}$. If the equation holds, the validity of the user B is confirmed by the user A. Then, the user A computes $H_{AB} = h(ID_A\|ID_B\|SK'\|K_B)$ and submits $\{H_{AB}\}$ to the user B. When receiving the message $\{H_{AB}\}$ from A in step 6, the user B computes $H'_{AB} = h(ID_A\|ID_B\|SK\|K_B)$, and checks whether $H'_{AB} = H_{AB}$. If they are equal, the validity of the user A is authenticated by the user B, and the mutual authentication between the user A and the user B is achieved.

Therefore, our presented protocol is free from the unknown key-share attack, and the user A and the user B believe a session key is established between them.

4.6 The proposed protocol provides key control

For a key agreement protocol, the property of key control means that neither of the entities can predetermine the value of the session key. In our presented protocol, the session key between user A and user B is $SK = T_{ab}(X)$, which contains both user A's contribution a and user B's contribution b, and both of them cannot predetermine the value of the session key. Besides, even though the server S can get information $ID_A, ID_B, K_A = T_a(X), K_B = T_b(X)$, he/she still cannot get $T_{ab}(X)$ because of the difficulty of chaotic maps DLP and DHP, and hence, the server S cannot get the session key shared between the users.

4.7 Replay attack

The attacker can eavesdrop the communication messages from the public channel, and then may want to impersonate the users or the server by replaying the eavesdropped messages. In our presented protocol, in order to impersonate as the user A, an attacker can replay user A's message $\{ID_A, C_1\}$ to user B in step 1, and he/she will receive the reply message $\{C_4, H_{BA}\}$ in step 4; however, the attacker cannot decrypt C_4 because he/she does not have the user A's secret $h(ID_A\|x)$, so the attacker cannot generate the valid reply message $\{H_{AB}\}$. Therefore, the attacker cannot replay user A's message successfully. By the same token, without knowing the user B's secret information $h(ID_B\|x)$, the attacker cannot successfully replay user B's message $\{ID_B, C_2\}$ because he/she cannot decrypt the message C_3 replied by the server S. Besides, in our protocol, the user A and user B generate the random numbers a and b, respectively, and the random numbers are different for each session; therefore, the attacker cannot impersonate the server S by replaying the message $\{C_3, C_4\}$.

4.8 Man-in-the-middle attack

As shown in Sect. 4.4, the mutual authentication is achieved between any two parties of the user A, user B and server S in the authentication and key agreement phase. Furthermore, the messages C_1, C_2, C_3, C_4 contain the identities of user A and user B. Therefore, our presented protocol is free from man-in-the-middle attack.

4.9 Impersonation attack

In the proposed protocol, the messages exchanged between the server and the user are encrypted under the control of the user's secret key. The impersonation attack on the user A and user B will not succeed because the attacker does not have the secret information $h(ID_A\|x)$ and $h(ID_B\|x)$. Besides, without the master secret key x of the server S, the attacker has no way to impersonate as the server S to communicate with the users.

4.10 Password-related attacks

In a key agreement protocol with password, the server has to store the password for each user, and the password table becomes an attractive target. It will lead to huge security problem if the server is compromised

Table 2 Performance comparisons

	User A	User B	Server S	Total	Time cost (ms)
Lai et al. [25]	$3T_C + 5T_H$	$3T_C + 5T_H$	$2T_C + 2T_E + 6T_H$	$8T_C + 2T_E + 16T_H$	261.7
Zhao et al. [26]	$3T_C + 1T_E + 6T_H$	$3T_C + 1T_E + 5T_H$	$2T_C + 2T_E + 8T_H$	$8T_C + 4T_E + 19T_H$	263.2
Xie et al. [27]	$3T_C + 2T_E + 5T_H$	$3T_C + 2T_E + 5T_H$	$2T_C + 4T_E + 4T_H$	$8T_C + 8T_E + 14T_H$	264
Our protocol	$2T_C + 2T_E + 4T_H$	$2T_C + 2T_E + 4T_H$	$4T_E + 3T_H$	$4T_C + 8T_E + 11T_H$	134.6

Table 3 Performance comparisons

	Lai et al. [25]	Zhao et al. [26]	Xie et al. [27]	Our protocol
Mutual authentication among three parties	Yes	Yes	Yes	Yes
No need of password table	No	No	No	Yes
No need of clock synchronization	No	No	Yes	Yes
Perfect forward secrecy	Yes	Yes	Yes	Yes
Known-key security	Yes	Yes	Yes	Yes
Resist insider attack	No	Yes	No	N/A
Resist password guessing attack	No	Yes	No	N/A
Resist key-compromise impersonation attack	Yes	Yes	Yes	Yes
Resist unknown key-share attack	Yes	Yes	Yes	Yes
Resist replay attack	Yes	Yes	Yes	Yes
Resist man-in-the-middle attack	Yes	Yes	Yes	Yes
Resist impersonation attack	Yes	Yes	Yes	Yes
Resist message modification attack	Yes	Yes	Yes	Yes
Communication rounds	7	7	5	5

or the password table is leaked. Besides, in password-based key agreement protocols, the low-entropy passwords easily suffer from password-related attacks, such as online/off-line password guessing attack, and insider attack. However, our protocol is not based on the password, the server S just needs to maintain a master secret key x, and the users just need to hold a secret information, and then the users can negotiate a session key under the assistance of the server S. Therefore, our presented protocol is free from any password-related attacks.

5 Comparisons in performance and functionality aspects

In this part, we evaluate the performance of our presented protocol, and compare it with other related protocols for performance and functionality aspects. In order to facilitate the analysis of the performance, we define some notations as below:

- T_C: The time consumption for a Chebyshev polynomial computation;
- T_E: The time consumption for a symmetric encryption/decryption;
- T_H: The time consumption for a one-way hash function.

Generally, T_C is far greater than T_E and T_H. According to the reference [19], under the environment of 3.2 GHz CPU and 3.0G RAM, T_C, T_E and T_H are 32.2, 0.45 and 0.2 ms, respectively.

Table 2 demonstrates that our presented protocol has high computational efficiency as compared with Lai et al. [25], Zhao et al. [26] and Xie et al.'s [27] protocols, where the computational cost for executing our protocol once is only half of the time needed for other related protocol due to our protocol needs less Chebyshev polynomial computation.

The functionality comparison among our presented protocol and other similar chaotic maps-based key agreement protocols are listed in Table 3. From Table 3,

we can find that our presented protocol needs less rounds of communication than Lai et al. [25] and Zhao et al.'s [26] protocols. Compared with other similar protocols [25–27], there are no password table stored on the server in our presented protocol, and then our protocol is free from password-related attacks, such as password guessing attack and insider attack. Besides, our protocol does not need the clock synchronization. In a word, the proposed protocol is a secure, communication- and computation-efficient chaotic maps-based authenticated key agreement protocol.

6 Conclusions

In this paper, we design a communication- and computation-efficient three-party authenticated key agreement protocol without password based on chaotic-maps, and analyze and compare it with related protocols in the aspects of performance and security. The proposed protocol does not need clock synchronization and can be deployed in most of the environments. In our presented protocol, the server does not need to manage a password table, which not only reduces the security burden on the server side, but also make the proposed protocol free from password-related attacks. Besides, the server does not need to pre-share the secret key with the users, and there is no key management in server side. Furthermore, our presented protocol can achieve known-key security and perfect forward secrecy, and can resist most of the known attacks. At the same time, our presented protocol is the most communication and computation efficient as compared to other related protocols due to the fact that it needs fewest communication rounds and Chebyshev polynomial computation. Therefore, the proposed protocol meets most of the requirements of a three-party authenticated key agreement protocol and is more suitable for real applications.

Acknowledgments This work was supported by the National Natural Science Foundation of China under Grant Nos. 61300220 & 61170296 & 61202462, the Research Fund of the State Key Laboratory of Software Development Environment, BUAA under Grant No. SKLSDE-2014KF-02, the China Postdoctoral Science Foundation Funded Project under Grant No. 2014M550590, the Scientific Research Fund of Hunan Provincial Education Department (Nos. 13C324 & 14A047), and the National Nature Science Foundation of Hunan province under Grant No. 13JJ3091.

References

1. Liu, B., Peng, J.: Nonlinear dynamics. High Education Press, Beijing (2004)
2. Wang, X.Y., Wang, X.J., Zhao, J.F., Zhang, Z.F.: Chaotic encryption algorithm based on alternant of stream cipher and block cipher. Nonlinear Dyn. **63**(4), 587–597 (2011)
3. Sheu, L.J.: A speech encryption using fractional chaotic systems. Nonlinear Dyn. **65**(1–2), 103–108 (2011)
4. Wang, Y., Wong, K.W., Liao, X.F., Xiang, T.: A block cipher with dynamic S-boxes based on tent map. Commun. Nonlinear Sci. Numer. Simul. **14**(7), 3089–3099 (2009)
5. Xiong, H., Chen, Z., Li, F.G.: New identity-based three-party authenticated key agreement protocol with provable security. J. Netw. Comput. Appl. **36**(2), 927–932 (2013)
6. He, D.B., Padhye, S., Chen, J.H.: An efficient certificateless two-party authenticated key agreement protocol. Comput. Math. Appl. **64**(6), 1914–1926 (2012)
7. Hölbl, M., Welzer, T., Brumen, B.: An improved two-party identity-based authenticated key agreement protocol using pairings. J. Comput. Syst. Sci. **78**(1), 142–150 (2012)
8. Lv, X.X., Li, H., Wang, B.C.: Group key agreement for secure group communication in dynamic peer systems. J. Parallel Distrib. Comput. **72**(10), 1195–1200 (2012)
9. He, D.B., Chen, J.H., Hu, J.: An ID-based client authentication with key agreement protocol for mobile client–server environment on ECC with provable security. Inf. Fusion **13**(3), 223–230 (2012)
10. Chen, T.H., Lee, W.B., Chen, H.B.: A round-and computation-efficient three-party authenticated key exchange protocol. J. Syst. Softw. **81**(9), 1581–1590 (2008)
11. Xiao, D., Liao, X.F., Deng, S.J.: A novel key agreement protocol based on chaotic maps. Inf. Sci. **177**(4), 1136–1142 (2007)
12. Han, S.: Security of a key agreement protocol based on chaotic maps. Chaos Solitons Fractals **38**(3), 764–768 (2008)
13. Xiang, T., Wong, K.W., Liao, X.F.: On the security of a novel key agreement protocol based on chaotic maps. Chaos Solitons Fractals **40**(2), 672–675 (2009)
14. Xiao, D., Liao, X.F., Deng, S.J.: Using time-stamp to improve the security of a chaotic maps-based key agreement protocol. Inf. Sci. **178**(6), 1598–1602 (2008)
15. Han, S., Chang, E.: Chaotic map based key agreement with/out clock synchronization. Chaos Solitons Fractals **39**(3), 1283–1289 (2009)
16. Tseng, H.R., Jan, R.H., Yang, W.: A chaotic maps-based key agreement protocol that preserves user anonymity. In: IEEE International Conference on Communications, 2009, ICC'09, pp. 1–6. Dresden, Germany (2009)
17. Niu, Y.J., Wang, X.Y.: An anonymous key agreement protocol based on chaotic maps. Commun. Nonlinear Sci. Numer. Simul. **16**(4), 1986–1992 (2011)
18. Yoon, E.J.: Efficiency and security problems of anonymous key agreement protocol based on chaotic maps. Commun. Nonlinear Sci. Numer. Simul. **17**(7), 2735–2740 (2012)
19. Xue, K.P., Hong, P.L.: Security improvement on an anonymous key agreement protocol based on chaotic maps. Commun. Nonlinear Sci. Numer. Simul. **17**(7), 2969–2977 (2012)

20. Tan, Z.W.: A chaotic maps-based authenticated key agreement protocol with strong anonymity. Nonlinear Dyn. **72**(1–2), 1–10 (2013)
21. Lee, C.C., Chen, C.L., Wu, C.Y., Huang, S.Y.: An extended chaotic maps-based key agreement protocol with user anonymity. Nonlinear Dyn. **69**(1–2), 79–87 (2012)
22. He, D.B., Chen, Y.T., Chen, J.H.: Cryptanalysis and improvement of an extended chaotic maps-based key agreement protocol. Nonlinear Dyn. **69**(3), 1149–1157 (2012)
23. Wang, X.Y., Zhao, J.F.: An improved key agreement protocol based on chaos. Commun. Nonlinear Sci. Numer. Simul. **15**(12), 4052–4057 (2010)
24. Yoon, E.J., Jeon, I.S.: An efficient and secure Diffie–Hellman key agreement protocol based on Chebyshev chaotic map. Commun. Nonlinear Sci. Numer. Simul. **16**(6), 2383–2389 (2011)
25. Lai, H., Xiao, J., Li, L., et al.: Applying semigroup property of enhanced Chebyshev polynomials to anonymous authentication protocol. Math. Probl. Eng. Article ID 454823, 17 pages(2012). doi:10.1155/2012/454823
26. Zhao, F.J., Gong, P., Li, S., Li, M.G., Li, P.: Cryptanalysis and improvement of a three-party key agreement protocol using enhanced Chebyshev polynomials. Nonlinear Dyn. **74**(1–2), 419–427 (2013)
27. Xie, Q., Zhao, J.M., Yu, X.Y.: Chaotic maps-based three-party password-authenticated key agreement scheme. Nonlinear Dyn. **74**(4), 1021–1027 (2013)
28. Farash, M.S., Attari, M.A.: An efficient and provably secure three-party password-based authenticated key exchange protocol based on Chebyshev chaotic maps. Nonlinear Dyn. (2014, in press). doi:10.1007/s11071-014-1304-6
29. Lee, C.C., Li, C.T., Hsu, C.W.: A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps. Nonlinear Dyn. **73**(1), 125–132 (2013)
30. Li, C.T., Lee, C.C., Weng, C.Y.: An extended chaotic maps based user authentication and privacy preserving scheme against DoS attacks in pervasive and ubiquitous computing environments. Nonlinear Dyn. **74**(4), 1133–1143 (2013)
31. Lee, C.C., Lou, D.C., Li, C.T., Hsu, C.W.: An extended chaotic-maps-based protocol with key agreement for multi-server environments. Nonlinear Dyn. **76**(1), 853–866 (2014)
32. Lee, C.C., Hsu, C.W.: A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps. Nonlinear Dyn. **71**(1), 201–211 (2013)
33. Zhang, L.H.: Cryptanalysis of the public key encryption based on multiple chaotic systems. Chaos Solitons Fractals **37**(3), 669–674 (2008)
34. Xiao, D., Shih, F.Y., Liao, X.F.: A chaos-based hash function with both modification detection and localization capabilities. Commun. Nonlinear Sci. Numer. Simul. **15**(9), 2254–2261 (2010)
35. Burrows, M., Abadi, M., Needham, R.M.: A logic of authentication. Proc. R. Soc. Lond. A Math. Phys. Sci. **1989**(426), 233–271 (1871)