Nonlinear Dyn (2015) 79:2485–2495
DOI 10.1007/s11071-014-1827-x

ORIGINAL PAPER

A new three-party-authenticated key agreement scheme based on chaotic maps without password table

Cheng-Chi Lee · Chun-Ta Li · Shih-Ting Chiu · Yan-Ming Lai

Received: 14 May 2014 / Accepted: 26 November 2014 / Published online: 3 December 2014

Abstract Three-party-authenticated key agreement allows two users to establish a common session key through a trusted server via an insecure communication channel. Early authenticated key agreement schemes were mostly based on either pairing operations, hash operations, or modular exponentiation operations. In 2011, Wang and Zhao took a new path and built their three-party-authenticated key agreement scheme on the basis of chaotic maps. By applying Chebyshev chaotic maps, Wang and Zhao succeeded in lifting their scheme up to a higher level of efficiency and security. In this paper, we shall propose a new three-party-authenticated key agreement scheme based on chaotic maps that can do without passwords. Keeping no password table, our new scheme is completely resistant to password guessing attacks. Besides that, our scheme also offers thorough privacy protection to the users, so the user forgery attack can cause no damage. Compared with the schemes currently available including Wang and Zhao's work, our new scheme obviously provides better security.

Keywords Chaotic maps · Three-party · Authenticated key agreement · Anonymity · Password table

1 Introduction

Password authentication schemes are among the many different designs used to verify users' identities on the Internet. Through a password authentication mechanism, the user can set up a password and pass the authentication. In 1981, Lamport [11] proposed a password authentication scheme in 1981. Since then, many researchers have followed the lead of Lamport's early work and extended it into different forms to be applied in different settings. The password-authenticated key agreement scheme makes one of the major varieties.

A password-authenticated key agreement scheme allows two parties to use their passwords to establish a session key so that they can authenticate each other via an insecure channel. Most of the password-authenticated key agreement schemes developed so far are mainly based on bilinear pairing, hash functions, or modular exponentiation operations. Tseng et al., how-

C.-C. Lee · S.-T. Chiu
Department of Library and Information Science, Fu Jen Catholic University, 510 Jhongjheng Rd., Xinzhuang District, New Taipei City 24205, Taiwan, ROC
e-mail: cclee@mail.fju.edu.tw

C.-T. Li (✉)
Department of Information Management, Tainan University of Technology, 529 Zhongzheng Road, Tainan 710, Taiwan, ROC
e-mail: th0040@mail.tut.edu.tw

Y.-M. Lai
Graduate Institute of Networking and Multimedia, National Taiwan University, #1 Roosevelt Rd. Sec. 4, Taipei 106, Taiwan, ROC

ever, decided to take an alternative route and build a password-authenticated key agreement scheme with user anonymity on the basis of a chaotic cryptosystem [4]. Unfortunately, their scheme was later proved to have some security weaknesses such as having a user anonymity problem, failing to provide perfect forward secrecy, and being vulnerable to insider attacks. In 2011, Niu and Wang pointed out some weaknesses in Tseng et al.'s design and proposed a new anonymity scheme with the trusted third party [3] to fix the problems. After that, in 2012, Xue and Hong came up with the idea of an anonymity scheme without the trusted third party [8]. Then, Yoon pointed out that Xue and Hong's scheme was not strong enough to resist the DoS attack [20]. In the same year, Guo and Chang proposed the first password-authenticated key agreement scheme based on chaotic maps with smart card [9].

Besides the advancement of the algorithm, the two-party design has also been developed into a three-party design to make the scheme more adaptable [2,3,5,7,10,12–18,22]. In 2011, Wang and Zhao proposed a three-party-authenticated key agreement scheme based on chaotic maps [5]. Then, in 2012, Lai et al. [10] proposed another three-party key agreement scheme that is based on Chebyshev chaotic maps. In 2013, Zhao et al. pointed out that Lai et al.'s scheme had some security flaws and was not strong enough against insider attacks and offline password guessing attacks [22]. Meanwhile, Xie et al. also proposed a three-party password-authenticated key agreement protocol based on Chebyshev chaotic maps which allows two parties to establish a secure session key over an insecure communication channel [7]. In the same year, Lee et al. [13] proposed their new three-party password-based authenticated key exchange protocol with user anonymity. Then, in 2014, Farash and Attari proposed a new protocol and claimed that their design could outperform other schemes in terms of communication, computation, and security [2].

Due to the great advancements made in the development of the chaotic map-based cryptosystem, many new key agreement protocols have been created that can satisfy the requirements of different applications [9,12,15,24–28]. For example, Lee et al. exploited extended chaotic maps and developed a biometric-based remote user authentication scheme with key agreement in 2013 [15].

1.1 Contributions

Generally speaking, major security issues that password-authenticated key agreement schemes have include protection against password guessing attacks and solution to the password table maintenance problem. In a password-authenticated key agreement scheme, a password is given to a user to authenticate that particular user. If a legitimate user's password is somehow known to a malicious user that legal user's account is then exposed to danger. Although Xie et al. claimed that their scheme could do without a timestamp and could resist all the possible attacks identified, there are, unfortunately, still some attacks we have found can cause damage to Xie et al.'s scheme. The security flaw will be further specified in detail later in Sect. 3.2. Here, we shall discuss what achievements we have made in the new scheme we are going to present in this paper as follows:

1. In our design, two users can use their identities to establish a common session key to authenticate each other via an insecure channel with the trusted server's help.
2. Our new scheme guarantees user anonymity. No information about the identities of the participants can be learned by anyone except for the participants themselves in the course of the communication session.
3. The messages sent to the server that contain the two users' identities are encrypted by the two users. With the messages properly decrypted, the server can determine whether or not the messages came from the real communication participants and thus rule out the possibility of the user forgery attack.
4. Using no password, our new scheme saves the trouble of keeping a password table and stays clear of the password guessing attack.

1.2 Organization

The organization of our paper is as follows. In Sect. 2, we will review Chebyshev chaotic maps and discuss some important properties. Then, in Sect. 3, we will review Xie et al.'s three-party password-authenticated key agreement scheme and show the weaknesses we have identified. In Sect. 4, we will present our new three-party-authenticated key agreement scheme based on chaotic maps without password table. Then, the new scheme will be analyzed in Sect. 5. Finally, our conclusion will be presented in Sect. 6.

2 Chebyshev chaotic maps

The concept of Chebyshev polynomial was proposed by Mason and Handscomb in 2003 [19]. The Chebyshev polynomial $T_n(x)$ is a polynomial in x of degree n. Let n be an integer, and let x be a variable taking value over the interval $[-1, 1]$. Then, the Chebyshev polynomial $T_n(x) : [-1, 1] \to [-1, 1]$ can be defined as follows:

$$T_n(x) = \cos(n \cdot \arccos(x))$$

The recurrence relation of the Chebyshev polynomial is defined as

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \text{ where } n \geq 2,$$
$$T_0(x) = 1, T_1(x) = 1$$

Below are some examples of the Chebyshev polynomial:

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

The Chebyshev polynomial exhibits two important properties: the semi-group property and the chaotic property.

- Semi-group property

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

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

- Chaotic property

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

2.1 Extended chaotic maps

Zhang [21] extended the range of the semigroup property and proved that the semi-group property holds for Chebyshev polynomials defined over the interval $(-\infty, +\infty)$. In other words, now we come to:

$$T_n(x) \equiv (2xT_{n-1}(x) - T_{n-2}(x)) \mod p,$$

where $n \geq 2$, $x \in (-\infty, +\infty)$, and p is a large prime.

As a result, now we have:

$$T_r(T_s(x)) \equiv T_{sr}(x) \equiv T_s(T_r(x)) \mod p$$

That is to say, the semi-group property is still there for extended chaotic maps.

There are two kinds of problems the Chebyshev polynomial can form: the discrete logarithm problem (DLP) and the Diffie–Hellman problem (DHP).

(1) Chebyshev chaotic map-based discrete logarithm problem (DLP)
Given x and y, it is difficult to find an integer r so that $T_r(x) = y$.

(2) Chebyshev chaotic map-based Diffie–Hellman problem (DHP)
Given x, $T_r(x)$, and $T_s(x)$, it is difficult to find $T_{rs}(x)$.

3 Review of Xie et al.'s scheme

In 2013, Xie et al. proposed a three-party password-authenticated key agreement scheme based on chaotic maps [7]. Unfortunately, we found some security flaws in Xie et al.'s scheme. In this section, we shall first briefly review Xie et al.'s scheme and then specify the weaknesses of the scheme.

3.1 Xie et al.'s scheme

In this subsection, we will review Xie et al.'s scheme. Before getting into the scheme itself, some notations have to be defined first. Table 1 shows the notations used in Xie et al.'s scheme and the definitions.

There are five steps to Xie et al.'s scheme. Note that $A \to B : (m)$ means A sends a message m to B.

Step 1: $A \to B : (m_1 = \{T_a(x), ID_A, C_A\})$

User A chooses a and computes $K_{AS} = T_a T_k(x)$, $H_A = h(T_a(x) \parallel ID_A \parallel ID_B \parallel pw_A)$ and $C_1 = E_{K_{AS}}(ID_A \parallel ID_B \parallel H_A)$. After computing these values, A sends $m_1 = \{T_a(x), ID_A, C_1\}$ to B.

Step 2: $B \to S : (m_2 = \{m_1, T_b(x), ID_B, C_2\})$

Upon receiving m_1 form A, User B chooses b and computes $K_{BS} = T_b T_k(x)$, $H_B = h(T_b(x) \parallel$

Table 1 Notations used in Xie et al.'s scheme

Notations	Descriptions
$(x, T_k(x))/k$	The server S's public key and secret key
$h()$	A chaotic map-based one-way hash function
$E_K()/D_K()$	Secure symmetric encryption/decryption functions with key K
A, B	Two participants
S	The trusted server
ID_A, ID_B	$A's$ and B's identities
pw_A, pw_B	$A's$ and B's passwords shared by the server

$ID_B \parallel ID_A \parallel pw_B$), and $C_2 = E_{K_{BS}}(ID_B \parallel ID_A \parallel H_B)$. After computing these values, B sends $m_2 = \{m_1, T_b(x), ID_B, C_2\}$ to S.

Step 3: $S \rightarrow B : (m_3 = \{C_3, C_4\})$

Upon receiving m_2 form B, the server S first computes $K_{SA} = T_k T_a(x)$ and $K_{SB} = T_k T_b(x)$ and then decrypts C_1 and C_2 to get ID_A, ID_B, H_A, and H_B. Then, S checks $H_A = ?h(T_a(x) \parallel ID_A \parallel ID_B \parallel pw_A)$ and $H_B = ?h(T_b(x) \parallel ID_B \parallel ID_A \parallel pw_B)$. If positive, then S computes $H_{SB} = h(T_a(x) \parallel pw_B)$, $C_3 = E_{K_{SB}}(ID_B \parallel ID_A \parallel T_a(x) \parallel H_{SB})$, $H_{SA} = h(T_b(x) \parallel pw_A)$, and $C_4 = E_{K_{SA}}(ID_A \parallel ID_B \parallel T_b(x) \parallel H_{SA})$ and then sends C_3 and C_4 to B.

Step 4: $B \rightarrow A : (m_4 = \{H_{BA}, C_4\})$

Upon receiving m_3 from S, User B first decrypts C_3 to get $ID_B, ID_A, T_a(x)$, and H_{SB}, and then B checks $h(T_a(x) \parallel pw_B) = ?H_{SB}$. If yes, B computes $SK = T_b T_a(x)$ and $H_{BA} = h(SK \parallel ID_B \parallel ID_A \parallel C_4)$. Then, B sends $m_4 = \{H_{BA}, C_4\}$ to A.

Step 5: $A \rightarrow B : (m_5)$

Upon receiving m_4 from B, User A first decrypts C_4 to get $ID_A, ID_B, T_b(x)$, and H_{SA}, and then A checks $h(T_b(x) \parallel pw_A) = ?H_{SA}$. If yes, A computes $SK = T_a T_b(x)$ and then checks $h(SK \parallel ID_B \parallel ID_A \parallel C_4) = ?H_{BA}$. After confirming the value, A computes $m_5 = h(SK \parallel ID_A \parallel ID_B)$ and sends it to B.

Upon receiving m_5 from A, B checks $h(SK \parallel ID_A \parallel ID_B) = ?m_5$. If positive, the session key between A and B, namely $SK' = h(T_a T_b(x))$, is confirmed.

Figure 1 is an illustration of the scheme structure.

3.2 Weaknesses of Xie et al.'s scheme

In spite of Xie et al.'s claim that their scheme can resist all possible attacks such as the off-line password guessing attack, the on-line password guessing attack, the replay attack, as well as the man-in-the-middle attack, we found that Xie et al.'s scheme is in fact vulnerable to the on-line password guessing attack and has a password table maintenance problem. In this subsection, we will point out where the weaknesses are and how to break Xie et al.'s scheme.

3.2.1 Anonymity of users

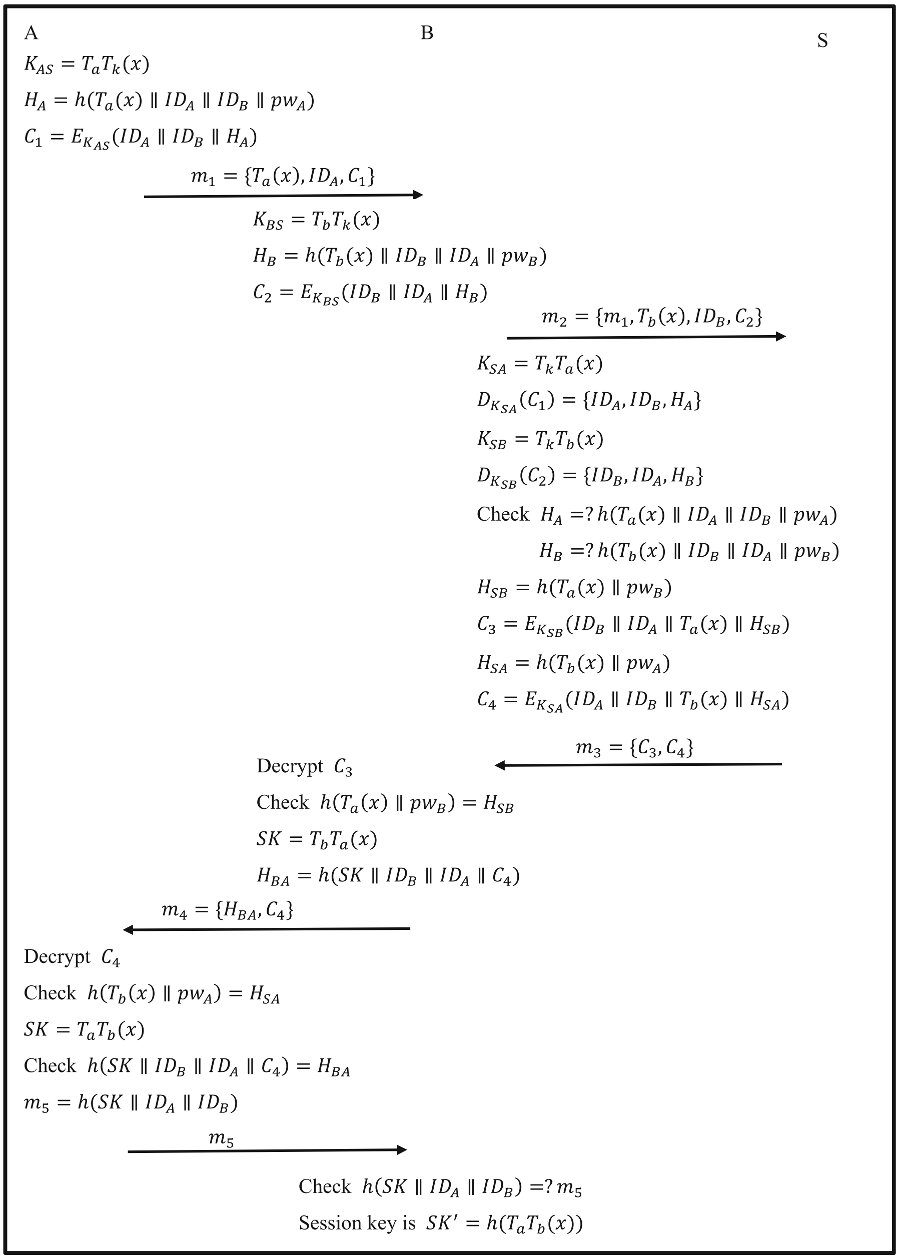
Both messages m_1 and m_2, sent, respectively, from User A to User B and from User B to the server S, contain the identity of the sender unencrypted. Once an attacker intercepts either of the messages, the attacker gets to know who the message sender is immediately. In other words, Xie et al.'s scheme fails to provide user anonymity.

3.2.2 On-line password guessing attack

Xie et al.'s claim that their scheme can resist the on-line password guessing attack turns out to be false. In this subsection, we will provide an example to demonstrate how the on-line password guessing attack can break Xie et al.'s scheme.

Suppose an attacker intercepts both messages m_1 and m_2 that are sent, respectively, from A to B and from B to S. As we said earlier, the attacker gets to know A's and B's identities easily. Then, the attacker pretends to be User B and chooses b and then computes $K_{BS} = T_b T_k(x)$, $H_B = h(T_b(x) \parallel ID_B \parallel ID_A \parallel pw_B^*)$, and $C_2 = E_{K_{BS}}(ID_B \parallel ID_A \parallel H_B)$, where pw_B^* is the attacker's guess of the password. After computing these values, the attacker sends $m_2 = \{m_1, T_b(x), ID_B, C_2\}$ to the server S. If the server S does return m_3 to the attacker, that means the attacker's guess is correct. Until then, the attacker can try and fail time and time again. In addition, if the attacker simply sends a large number of messages to the server, the server will be very busy receiving these messages and authenticating or rejecting them. As a result, too much of the network resources will be occupied, temporarily stopping the system from functioning properly. This is a kind of DoS attack.

Fig. 1 Xie et al.'s scheme


A
B
S
$K_{AS} = T_a T_k(x)$
$H_A = h(T_a(x) \parallel ID_A \parallel ID_B \parallel pw_A)$
$C_1 = E_{K_{AS}}(ID_A \parallel ID_B \parallel H_A)$
$m_1 = \{T_a(x), ID_A, C_1\}$
$K_{BS} = T_b T_k(x)$
$H_B = h(T_b(x) \parallel ID_B \parallel ID_A \parallel pw_B)$
$C_2 = E_{K_{BS}}(ID_B \parallel ID_A \parallel H_B)$
$m_2 = \{m_1, T_b(x), ID_B, C_2\}$
$K_{SA} = T_k T_a(x)$
$D_{K_{SA}}(C_1) = \{ID_A, ID_B, H_A\}$
$K_{SB} = T_k T_b(x)$
$D_{K_{SB}}(C_2) = \{ID_B, ID_A, H_B\}$
Check $H_A =? h(T_a(x) \parallel ID_A \parallel ID_B \parallel pw_A)$
$H_B =? h(T_b(x) \parallel ID_B \parallel ID_A \parallel pw_B)$
$H_{SB} = h(T_a(x) \parallel pw_B)$
$C_3 = E_{K_{SB}}(ID_B \parallel ID_A \parallel T_a(x) \parallel H_{SB})$
$H_{SA} = h(T_b(x) \parallel pw_A)$
$C_4 = E_{K_{SA}}(ID_A \parallel ID_B \parallel T_b(x) \parallel H_{SA})$
$m_3 = \{C_3, C_4\}$
Decrypt C_3
Check $h(T_a(x) \parallel pw_B) = H_{SB}$
$SK = T_b T_a(x)$
$H_{BA} = h(SK \parallel ID_B \parallel ID_A \parallel C_4)$
$m_4 = \{H_{BA}, C_4\}$
Decrypt C_4
Check $h(T_b(x) \parallel pw_A) = H_{SA}$
$SK = T_a T_b(x)$
Check $h(SK \parallel ID_B \parallel ID_A \parallel C_4) = H_{BA}$
$m_5 = h(SK \parallel ID_A \parallel ID_B)$
m_5
Check $h(SK \parallel ID_A \parallel ID_B) =? m_5$
Session key is $SK' = h(T_a T_b(x))$

3.2.3 Password table maintenance problem

In Xie et al.'s scheme, the two users have to use their passwords to establish their common session key, but the communications between *A* and *B* as well as between *B* and *S* contain no information about the passwords. That means the server *S* has to keep a password table to store and update each user's password so as to verify the legitimacy of the users. Such a design gives a malicious insider a chance to modify the passwords stored in the password table.

4 The proposed scheme

In this section, we will show how our new three-party-authenticated key agreement scheme works step by step. First, the notations used in our scheme are defined

Table 2 Notations used in our scheme

Notations	Descriptions
A, B	Two participants
S	The trusted server
$T_k(ID_X)$	User's certificate issued by the server
k	The server S's secret key
$h()$	A one-way hash function based on chaotic maps
$E_K()/D_K()$	Secure symmetric encryption/decryption functions with key K
ID_A, ID_B	A's and B's identity

in Table 2. The structure of our scheme is illustrated in Fig. 2. Please note that in all five steps we use the same format of expression $A \rightarrow B : (m)$ to mean A sends a message m to B.

There are five steps to our scheme, which are detailed as follows:

Step 1: $A \rightarrow B : (m_1)$

User A chooses a and computes $K_{AS} = T_a T_k(ID_A)$, $H_A = h(T_a(ID_A) \parallel ID_A \parallel ID_B)$, and $C_A = E_{K_{AS}}(ID_A \parallel ID_B \parallel H_A \parallel T_a(ID_B))$ and then sends $m_1 = \{T_a\,(ID_A)\,, C_A\}$ to B.

Step 2: $B \rightarrow S : (m_1, m_2)$

Upon receiving m_1 form A, User B chooses b and computes $K_{BS} = T_b T_k(ID_B)$, $H_B = h(T_b(ID_B) \parallel ID_B)$, and $C_B = E_{K_{BS}}(ID_B \parallel H_B \parallel T_b(ID_B))$ and then sends m_1 and $m_2 = \{T_b\,(ID_B)\,, C_B\}$ to S.

Step 3: $S \rightarrow B : (C'_A, C'_B)$

Upon receiving m_1, m_2 form B, the server S first computes $K_{SA} = T_k T_a(ID_A)$, $K_{SB} = T_k T_b(ID_B)$, $D_A = D_{K_{SA}}(C_A) = \{ID_A \parallel ID_B \parallel H_A \parallel T_a(ID_B)\}$, and $D_B = D_{K_{SB}}(C_B) = \{ID_B \parallel H_B \parallel T_b(ID_B)\}$. Then, S checks ID_A, ID_B, $H_A = ?h(T_a(ID_A) \parallel ID_A \parallel ID_B)$, and $H_B = ?h(T_b(ID_B) \parallel ID_B)$. If both checks out, S computes $H_{SA} = h(T_k(ID_A) \parallel T_a(ID_A))$, $H_{SB} = h(T_k(ID_B) \parallel T_b(ID_B))$, $C'_A = E_{K_{SA}}(ID_A \parallel ID_B \parallel T_b(ID_B) \parallel H_{SA})$, and $C'_B = E_{K_{SB}}(ID_A \parallel ID_B \parallel T_a(ID_A) \parallel H_{SB})$. After computing C'_A and C'_B, S sends them to B.

Step 4: $B \rightarrow A : (C'_A,\ H_{BA})$

Upon receiving C'_AandC'_B from S, B first decrypts C'_B and checks ID_A and H_{SB}. Then B computes $SK = T_b T_a(ID_B)$ and $H_{BA} = h(SK \parallel C'_A)$. After computing H_{BA}, B sends C'_A and H_{BA} to A.

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

Upon receiving C'_AandH_{BA} from B, A first decrypts C'_A and checks H_{SA}. Then, A computes $SK = T_a T_b(ID_B)$. After computing SK, A checks $H_{BA} = ?h(SK \parallel C'_A)$. If positive, A computes $H_{AB} = h(SK \parallel ID_A \parallel T_a(ID_B))$ and sends it to B.

Upon receiving H_{AB} from A, B confirms H_{AB}. If it checks out, $SK' = h(SK)$ will be the session key between A and B.

5 Analysis of our scheme

In general, the security of a scheme can be checked by performing either a formal analysis or a heuristic analysis. In this study, we follow the latter route. Before discussing the results of our heuristic security analysis, let's first take a look at how our new scheme compares with Zhao et al.'s [22], Xie et al.'s [7] as well as Farash and Attari's [2] scheme in terms of security and performance. Then, the security of our new scheme will be further analyzed by applying the BAN logic [1,6,23] to check the correctness.

5.1 Comparisons

In this subsection, we will compare the security and performance of our new scheme with those of Zhao et al.'s, Xie et al.'s, as well as Farash and Attari's scheme. Please note that Zhao et al.'s, Xie et al.'s, and the Farash–Attari scheme are password-based authenticated key agreement schemes, and therefore, there is a password table to keep on the server's side. By contrast, our scheme works without passwords and is therefore free from password table maintenance problems.

5.1.1 Security comparisons

Before getting into the details of the security comparisons, let's define the expressions we are going to use later in our summary table, namely Table 3. First of all, the term On-line AK is short for the on-line password guessing attack; similarly, the term Off-line AK is short for the off-line password guessing attack. Then, the single word Anonymity is used in the table to mean user anonymity, and finally, the expression PW TablePro is short for password table problem.

As Table 3 reveals, neither Xie et al.'s scheme nor the Farash–Attari scheme provides user anonymity. In other words, in those two schemes, the user's identity is sent in the form of plaintext as part of the mes-

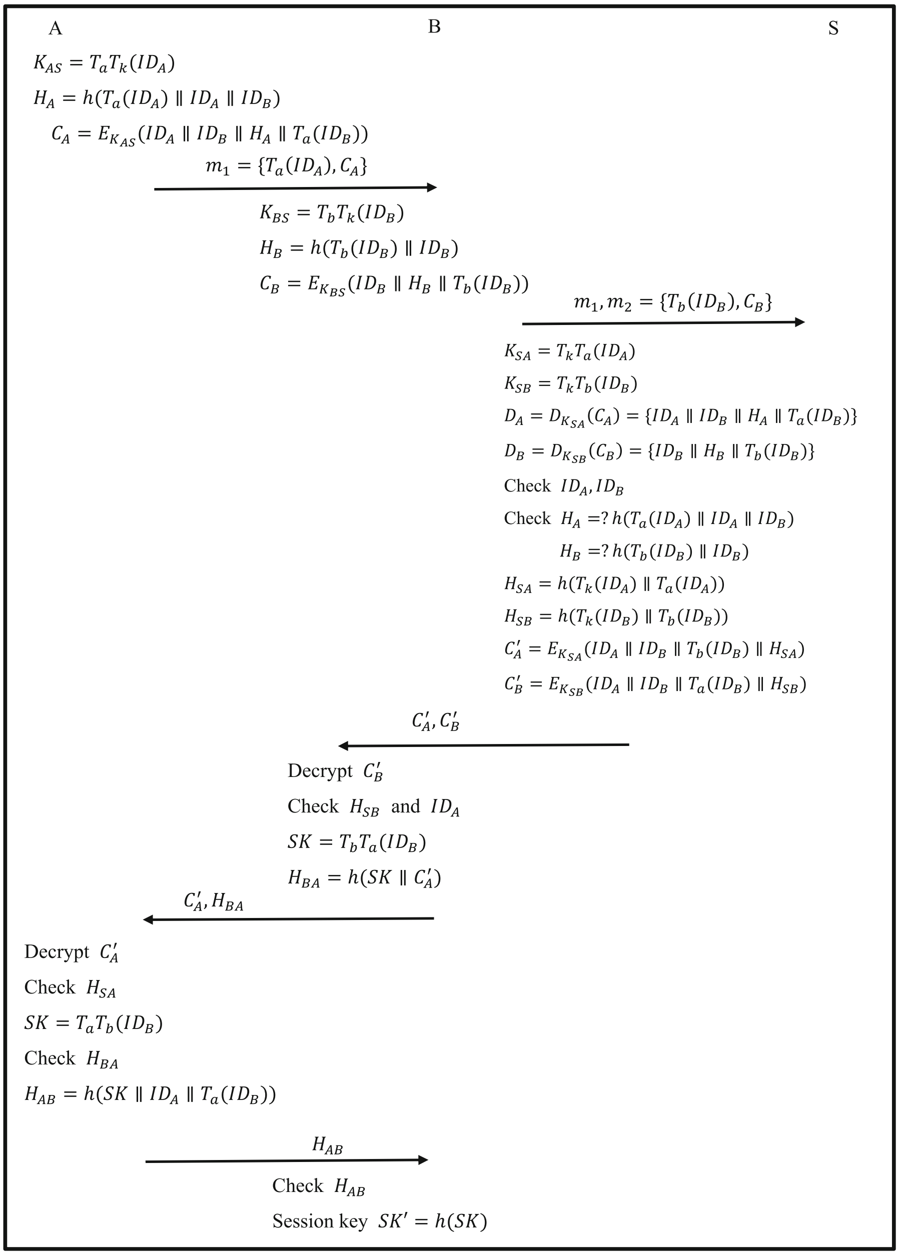

A
B
S
$K_{AS} = T_a T_k(ID_A)$
$H_A = h(T_a(ID_A) \parallel ID_A \parallel ID_B)$
$C_A = E_{K_{AS}}(ID_A \parallel ID_B \parallel H_A \parallel T_a(ID_B))$
$m_1 = \{T_a(ID_A), C_A\}$
$K_{BS} = T_b T_k(ID_B)$
$H_B = h(T_b(ID_B) \parallel ID_B)$
$C_B = E_{K_{BS}}(ID_B \parallel H_B \parallel T_b(ID_B))$
$m_1, m_2 = \{T_b(ID_B), C_B\}$
$K_{SA} = T_k T_a(ID_A)$
$K_{SB} = T_k T_b(ID_B)$
$D_A = D_{K_{SA}}(C_A) = \{ID_A \parallel ID_B \parallel H_A \parallel T_a(ID_B)\}$
$D_B = D_{K_{SB}}(C_B) = \{ID_B \parallel H_B \parallel T_b(ID_B)\}$
Check ID_A, ID_B
Check $H_A =? h(T_a(ID_A) \parallel ID_A \parallel ID_B)$
$H_B =? h(T_b(ID_B) \parallel ID_B)$
$H_{SA} = h(T_k(ID_A) \parallel T_a(ID_A))$
$H_{SB} = h(T_k(ID_B) \parallel T_b(ID_B))$
$C'_A = E_{K_{SA}}(ID_A \parallel ID_B \parallel T_b(ID_B) \parallel H_{SA})$
$C'_B = E_{K_{SB}}(ID_A \parallel ID_B \parallel T_a(ID_B) \parallel H_{SB})$
C'_A, C'_B
Decrypt C'_B
Check H_{SB} and ID_A
$SK = T_b T_a(ID_B)$
$H_{BA} = h(SK \parallel C'_A)$
C'_A, H_{BA}
Decrypt C'_A
Check H_{SA}
$SK = T_a T_b(ID_B)$
Check H_{BA}
$H_{AB} = h(SK \parallel ID_A \parallel T_a(ID_B))$
H_{AB}
Check H_{AB}
Session key $SK' = h(SK)$

Fig. 2 Our proposed scheme

sage being communicated and can very easily be intercepted and used to break the security without having to decrypt anything. On the other hand, among the three password-based schemes compared, only Xie et al.'s scheme offers no protection against the on-line password guessing attack. As for our new scheme, just like Zhao et al.'s scheme, it offers user anonymity. In addition, since our scheme uses no password, there is certainly no password table maintenance problem to worry about, and nor will the on-line/off-line password guessing attacks be a problem. Some important aspects of the security our new scheme has to offer are specified as follows:

1. User anonymity
   When the two users need to establish their common session key, they need to inform each other as well as the server of their own identity. In other

Table 3 Security comparisons

	On-line AK	Off-line AK	Anonymity	PW TablePro
Zhao et al.	x	x	v	v
Xie et al.	v	x	x	v
Farash and Attari	x	x	x	v
Our scheme	–	–	v	–

words, in the message sent from one user to the other, there is the sender's identity. In our scheme, the identity in the message sent during communication is not in the form of plaintext but encrypted by using the Chebyshev polynomial. This way, should a malicious attacker intercept the message, there is no way the attacker can obtain the user's real identity by analyzing the message.

2. Protection against user identity forgery
   The message sent to the server S contains the two users' identities (one encrypted by using the Chebyshev polynomial and the other not). Upon receiving the message, the server S can immediately decrypt them and then check the identities of both users.
3. Keeping no password table
   For a password authentication scheme to work properly, there must be a password table on the server's side so that the legal participants' passwords can be stored and updated. If the server has evil intentions, an insider attack can happen, and the passwords may be abused or manipulated. Since our scheme does not keep the participants' passwords, there is no password table on the server's side and therefore no risk of insider attack.
4. Avoidance of password guessing attack
   A password authentication scheme always runs the risk of being damaged by the password guessing attack. With a message intercepted, the attacker will try to guess the correct password. If the password is guessed correctly, then the attacker can use it to do something illegal. Distinct from password authentication schemes, our scheme does not use passwords and is therefore secure from password guessing attacks.

5.1.2 Performance comparisons

In 2011, Xue and Hong [8] estimated the average running times of some commonly used operations. Xue and Hong's experiments were conducted in an environment where the processing speed of the CPU was 3.2 GHz with the RAM of 3.0 G, and the results are shown in Table 4. As the table reveals, the average running time of the Chebyshev polynomial is about 32.2 ms, the average hash function operation takes about 0.2 ms, and the average symmetric encryption/decryption operation takes about 0.45 ms.

Table 4 The running time of different operations

Operations	Chebyshev polynomial	Hash function	Symmetric encryption/decryption
Time (ms)	32.2	0.2	0.45

In our summary table (Table 5), C refers to a Chebyshev polynomial computation operation, H refers to a hash function operation, and E refers to a symmetric encryption/decryption operation. The total time each scheme averagely consumes is computed based on Xue and Hong's experiment results. Based solely on Table 5, it appears that our scheme is not the most efficient of the schemes compared. However, in fact some hidden factors that Table 5 fails to cover can also affect the real efficiency performance. For all three password-based schemes compared, due to the use of the password table, quite a number of time-consuming id–password table search operations, including database connection, search algorithm execution, decryption of encrypted passwords, etc., will inevitably have to be carried out. In practice, search operations are always more expensive when the number of registered users grows bigger. Unfortunately, Table 5 does not have those included. By contrast, no matter how the number of registered users increases, our scheme can always keep the cost fixed. Therefore, in the long run, our new scheme does have an advantage over the other schemes.

5.2 Correctness analysis

The BAN logic is a well-established way to verify the correctness of information exchange protocols. In this subsection, we will use the BAN logic [1,6] to analyze the correctness of the session key between A and B. To begin with, the notations, goals, and assumptions are defined as follows.

Table 5 Performance comparisons

	User A	User B	Server S	Total	Total time
Zhao et al.	$3C+6H+E$	$3C+6H+E$	$2C+8H+2E$	$8C+20H+4E$	263.4
Xie et al.	$3C+4H+2E$	$3C+4H+2E$	$2C+4H+4E$	$8C+12H+8E$	263.6
Farash and Attari	$3C+4H$	$3C+4H$	$2C+4H$	$8C+12H$	260
Our scheme	$4C+4H+2E$	$3C+4H+2E$	$4C+4H+4E$	$11C+12H+8E$	360.2

5.2.1 Notations

Here, the syntax and notations of the BAN logic are specified. We define A and B as the specific participators, S is the trusted server, and X is the formula (statement). There are some rules as follows [1,6,23]:

1. $A \mid\equiv X$ means A believes the formula X is true.
2. $A \mid\equiv B$ means A believes B's action.
3. $A \lhd X$ means A holds or sees the formula X.
4. $A \mid\sim X$ means A has said the formula X.
5. $A \mid\Rightarrow X$ means A has complete control over the formula X.
6. $\overset{K_A}{\mapsto} A$ means K is the public key for A and K_A^{-1} is the private key for A.
7. $\frac{\text{Rule 1}}{\text{Rule 2}}$ means Rule 2 is from Rule 1.
8. $A \overset{x}{\leftrightarrow} B$ means x is a secret key or secret information share between A and B.
9. $\{X\}_K$ means X is encrypted by the key K.

5.2.2 Goals

First, there are three roles in our scheme: A and B are the users who need to generate a common session key between them with the help of the trusted server (S). There are four goals our scheme is to achieve in the language of the BAN logic:

G1. $B \mid\equiv S \mid\equiv A \lhd T_k(ID_A)$
G2. $A \mid\equiv S \mid\equiv B \lhd T_k(ID_B)$
G3. $A \mid\equiv B \lhd A \overset{SK}{\leftrightarrow} B$
G4. $B \mid\equiv A \lhd A \overset{SK}{\leftrightarrow} B$

Since A and B need to generate a common session key for their communication, A must believe that the server believes B and that B holds the session key SK, and vice versa.

5.2.3 Assumptions

With the goals set, the assumptions also need to be stated:

A1. $A \lhd ID_A$
A2. $A \lhd ID_B$
A3. $B \lhd ID_B$
A4. $A \mid\Rightarrow a$
A5. $B \mid\Rightarrow b$
A6. $S \mid\Rightarrow (T_k(ID_A), T_K(ID_B))$

In assumptions $A1$ through $A3$, A and B each hold their own identities. Since A wishes to generate a common session key with B, A needs to hold B's identity, and the server S can then check the identities of both participants in this communication. In assumptions $A4$ through $A6$, A, B and S each need to select their own secret keys, which they have complete control over.

5.2.4 Verification

In this subsection, we will check the correctness of our proposed scheme by exploiting the BAN logic. The main steps of the proof are as follows:

A computes K_{AS} and H_A
Message 1: $A \rightarrow B : (m_1 = T_a(ID_A), \{ID_A \parallel ID_B \parallel H_A \parallel T_a(ID_B)\}_{K_{AS}})$
$V1.\quad B \lhd m_1$
B computes K_{AS} and H_B
Message 2: $B \rightarrow S : (m_1, m_2 = T_b(ID_B), \{ID_B \parallel H_B \parallel T_b(ID_B)\}_{K_{BS}})$
$V2.\quad S \lhd m_1, m_2$
S computes K_{SA}, K_{SB}
$V3.\quad \frac{S \lhd K_{SA}, K_{SB}}{S \lhd ID_A, ID_B, H_A, H_B, T_a(ID_A), T_b(ID_B)}$
$V4.$
$\frac{S \lhd ID_A, ID_B, H_A, H_B, T_a(ID_A), T_b(ID_B), S \mid\Rightarrow (T_k(ID_A), T_K(ID_B))}{S \mid\equiv K_{SA}, K_{SB}}$
$V5.\quad \frac{S \mid\equiv K_{SA}, K_{SB}, S \mid\Rightarrow (T_k(ID_A), T_K(ID_B))}{S \mid\equiv A \lhd T_k(ID_A), S \mid\equiv B \lhd T_k(ID_B)}$
S computes H_{SA}, H_{SB}
Message 3:

$S \rightarrow B : (\{ID_A \parallel ID_B \parallel T_b(ID_B) \parallel H_{SA}\}_{K_{SA}}, \{ID_A \parallel ID_B \parallel T_a(ID_B) \parallel H_{SB}\}_{K_{SB}})$

$V6.\ B \lhd \{ID_A \parallel ID_B \parallel T_b(ID_B) \parallel H_{SA}\}_{K_{SA}}, \{ID_A \parallel ID_B \parallel T_a(ID_B) \parallel H_{SB}\}_{K_{SB}}$

$V7.\ \frac{B \lhd K_{BS}}{B \lhd ID_A, T_a(ID_B), H_{SB}}$

$V8.\ \frac{B \lhd T_k(ID_B)}{B| \equiv S| \sim H_{SB}}$

$V9.\ \frac{B| \equiv S| \sim H_{SB}}{B| \equiv S| \equiv A \lhd T_k(ID_A), B| \equiv T_a(ID_B)}$

B computes $A \overset{SK}{\leftrightarrow} B$, H_{BA}

Message 4: $B \rightarrow A$: $\{ID_A \parallel ID_B \parallel T_b(ID_B) \parallel H_{SA}\}_{K_{SA}}, H_{BA}$

$V10.\ A \lhd \{ID_A \parallel ID_B \parallel T_b(ID_B) \parallel H_{SA}\}_{K_{SA}}, H_{BA}$

$V11.\ \frac{A \lhd K_{AS}}{A \lhd T_b(ID_B), H_{SA}}$

$V12.\ \frac{A \lhd T_k(ID_A)}{A| \equiv S| \sim H_{SA}}$

$V13.\ \frac{A| \equiv S| \sim H_{SA}}{A| \equiv S| \equiv B \lhd T_k(ID_B), A| \equiv T_b(ID_B)}$

$V14.\ \frac{A| \Rightarrow a, A| \equiv T_b(ID_B)}{A| \equiv H_{BA}}$

$V15.\ \frac{A| \equiv H_{BA}}{A| \equiv B \lhd A \overset{SK}{\leftrightarrow} B}$

A computes H_{AB}

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

$V16.\ \frac{B| \equiv A \overset{SK}{\leftrightarrow} B, T_a(ID_B)}{B| \equiv H_{AB}}$

$V17.\ \frac{B| \equiv H_{AB}}{B| \equiv A \lhd A \overset{SK}{\leftrightarrow} B}$

In formula $V9$ and formula $V13$, B and A believe that the server has said H_{SB} and H_{SA}. Because the server has to verify the certificate before issuing H_{SB} and H_{SA}, A and B each believes that the other party is a legal user. In formula $V15$, since A has a, $T_b(ID_B)$, A can compute the session key SK. When A can decrypt C'_A and holds SK, A can believe H_{BA}, so A believes that B holds the secret value SK. Similarly, in formula $V17$, B believes that A holds the secret value SK. With this secret value, A and B can generate their common session key. Form formulas $V9$, $V13$, $V15$ and $V17$, we can infer that our scheme achieves the goals.

6 Conclusion

In this paper, we pointed out the security leaks in Xie et al.'s three-party password-authenticated key agreement scheme based on chaotic maps. Then, we solved the problem by proposing our new three-party-authenticated key agreement scheme based on chaotic maps. Compared with Xie et al.'s scheme, our new scheme performs on the same efficiency level but offers better security protection. Besides demonstrating the superiority of our new scheme in security by comparing it with several other schemes, we also performed a BAN logic test and confirmed the correctness of our scheme.

Acknowledgments The authors would like to express their appreciation to the anonymous referees for their valuable suggestions and comments. This research was partially supported by the National Science Council, Taiwan, R.O.C., under Contract No.: NSC102-2221-E-030-003.

References

1. Burrows, M., Abadi, M., Needham, R.: A logic of authentication. ACM Trans. Comput. Syst. **8**(1), 18–36 (1990)
2. 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). doi:10.1007/s11071-014-1304-6
3. Farash, M.S., Attari, M.A.: An enhanced authenticated key agreement for session initiation protocol. Inf. Technol. Control **42**(4), 333–342 (2013)
4. Farash, M.S., Attari, M.A.: A secure and efficient identity-based authenticated key exchange protocol for mobile client-server networks. J. Supercomput. (2014). doi:10.1007/s11227-014-1170-5
5. Farash, M.S., Attari, M.A.: An anonymous and untraceable password-based authentication scheme for session initiation protocol using smart cards. Int. J. Commun. Syst. (2014). doi:10.1002/dac.2848
6. Farash, M.S., Attari, M.A.: An improved password-based authentication scheme for session initiation protocol using smart cards without verification table. Int. J. Commun. Syst. (2014). doi:10.1002/dac.2879
7. Farash, M.S., Attari, M.A.: An enhanced and secure three-party password-based authenticated key exchange protocol without using server's public-keys and symmetric cryptosystems. Inf. Technol. Control **43**(2), 143–150 (2014)
8. Farash, M.S., Attari, M.A.: An efficient client-client password-based authentication scheme with provable security. J. Supercomput. (2014). doi:10.1007/s11227-014-1273-z
9. Guo, C., Chang, C.C.: Chaotic maps-based password authenticated key agreement using smart cards. Commun. Nonlinear Sci. Numer. Simul. **18**(6), 1433–1440 (2012)
10. Lai, H., Xiao, J., Li, L., Yang, Y.: Applying semigroup property of enhanced Chebyshev polynomials to anonymous authentication protocol. Math. Probl. Eng. **2012**, 17 (2012)
11. Lamport, L.: Password authentication with insecure communication. Commun. ACM **24**(11), 770–772 (1981)
12. Lee, C.C., Lou, D., 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)
13. 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)

14. Lee, C.C.: A simple key agreement scheme based on chaotic maps for VSAT satellite communications. Int. J. Satell. Commun. Netw. **31**(4), 177–186 (2013)
15. 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)
16. 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), 79–87 (2012)
17. Lee, C.C., Chen, S.D., Chen, C.L.: A computation-efficient three-party encrypted key exchange protocol. Appl. Math. Inf. Sci. **6**(3), 573–579 (2012)
18. Lee, C.C., Li, C.T., Chang, R.X.: An undetectable on-line password guessing attack on Nam et al'.s three-party key exchange protocol. J. Comput. Methods Sci. Eng. **13**(5–6), 455–460 (2013)
19. Mason, J.C., Handscomb, D.C.: Chebyshev Polynomials. Chapman & Hall/CRC Press, London (2003)
20. Niu, Y., Wang, X.: An anonymous key agreement protocol based on chaotic maps. Commun. Nonlinear Sci. Numer. Simul. **16**(4), 1986–1992 (2011)
21. Tseng, H., Jan, R., Yang, W.: A chaotic maps-based key agreement protocol that preserves user anonymity. In: IEEE International Conference on Communications, pp. 1–6 (2009)
22. Wang, X., Zhao, J.: An improved key agreement protocol based on chaos. Commun. Nonlinear Sci. Numer. Simul. **15**(12), 4052–4057 (2010)
23. Wessels, J.: Application of BAN-logic. CMG Financ. B.V., pp. 1–22 (2001)
24. Xie, Q., Zhao, J., Yu, X.: Chaotic maps-based three-party password-authenticated key agreement scheme. Nonlinear Dyn. **74**(4), 1021–1027 (2013)
25. Xue, K., Hong, P.: Security improvement on an anonymous key agreement protocol based on chaotic maps. Commun. Nonlinear Sci. Numer. Simul. **17**(7), 2969–2977 (2012)
26. Yoon, E.: Efficiency and security problems of anonymous key agreement protocol based on chaotic maps. Commun. Nonlinear Sci. Numer. Simul. **17**(7), 2735–2740 (2012)
27. Zhang, L.: Cryptanalysis of the public key encryption based on multiple chaotic systems. Chaos Solitons Fract. **37**(3), 669–674 (2008)
28. Zhao, F., Gong, P., Li, S., Li, M., Li, P.: Cryptanalysis and improvement of a three-party key agreement protocol using enhanced Chebyshev polynomials. Nonlinear Dyn. **74**(1–2), 419–427 (2013)