Nonlinear Dyn (2013) 74:1021–1027
DOI 10.1007/s11071-013-1020-7

ORIGINAL PAPER

Chaotic maps-based three-party password-authenticated key agreement scheme

Qi Xie · Jianmin Zhao · Xiuyuan Yu

Received: 19 January 2013 / Accepted: 22 July 2013 / Published online: 22 August 2013

Abstract Since chaos theory related to cryptography has been addressed widely, many chaotic maps based two-party password-authenticated key agreement (2PAKA) schemes have been proposed. However, to the best of our knowledge, no chaotic maps based three-party password-authenticated key agreement (3PAKA) protocol without using a timestamp has been proposed, yet. In this paper, we propose the first chaotic maps-based 3PAKA protocol without a timestamp. The proposed protocol is not based on the traditional public key cryptosystem but is based on chaotic maps, which not only achieves perfect forward secrecy without using a timestamp, modular exponentiation and scalar multiplication on an elliptic curve, but is also robust to resist various attacks such as password guessing attacks, impersonation attacks, man-in-the-middle attacks, etc.

Keywords Chaos · Chaotic maps · Password · Authenticated key agreement · Information security

Q. Xie (✉) · X. Yu
Hangzhou Key Laboratory of Cryptography and Network Security, Hangzhou Normal University, Hangzhou, China
e-mail: qixie68@126.com

J. Zhao
Department of Computer Science, Zhejiang Normal University, Jinhua, China

1 Introduction

An authenticated key agreement protocol allows communicating parties to authenticate each other via insecure network, and establishes a secure session key for their subsequent communications. With the rapid development of chaos theory related to cryptography [1–6], many chaotic maps based two-party password-authenticated key agreement schemes have been proposed [7–15]. In 2005, Xiao et al. [7] proposed a key agreement protocol using a chaotic public-key cryptosystem, but Alvarez [8] showed that their scheme cannot resist the man-in-the-middle attack. Then, based on the semi-group property of the Chebyshev chaotic map, Xiao et al. [9] proposed a novel key agreement protocol. However, Han [10] demonstrated that Xiao et al.'s protocol is still insecure. Xiang et al. [11] also pointed out that Xiao et al.'s protocol is vulnerable to a stolen-verifier attack and an offline guessing attack. After that, Xiao et al. [12] and Han–Chang [13] improved the security of a chaotic maps-based key agreement protocol using a timestamp. In 2010, Guo–Zhang [14] proposed a new chaotic maps-based key agreement protocol, but He [15] showed that their scheme is vulnerable to an offline guessing attack. In 2012, Gong et al. [16] proposed a maps-based key agreement protocol without using smart cards.

Based on the merits of password-authenticated key agreement protocol, in 2009, Tseng et al. [17] proposed a password and chaotic maps based key agreement protocol with anonymity. However, Niu and

Wang [18] showed that Tseng et al.'s protocol cannot achieve user anonymity and perfect forward secrecy, and cannot resist an insider attack, and further proposed an improved scheme with a trusted third party (TTP). But Xue and Hong [19] demonstrated that the Niu and Wang's scheme has some disadvantages and proposed a new one without TTP. The same year, Yoon [20] also showed that Niu and Wang's anonymous key agreement protocol is vulnerable to a Denial-of-Service (DoS) attack based on illegal message modification. Tan [21] showed that Xue and Hong's scheme cannot provide strong anonymity and is vulnerable to the man-in-the-middle attack. Very recently, Lee and Hsu [22] proposed a biometric-based remote user authentication with key agreement scheme using extended chaotic maps, and Guo and Chang [23] proposed the first chaotic maps-based password-authenticated key agreement using smart cards.

In 2010, Wang and Zhao [24] proposed a three-party key agreement protocol based on chaotic maps, but Yoon and Jeon [25] showed that their scheme needs timestamp information and is vulnerable to an illegal message modification attack, and then they proposed an improved scheme. In Wang and Zhao and Yoon and Jeon's schemes, the server and the two users should share long-term secret keys to achieve mutual authentication. However, it is inconvenient that the users should protect the long-term secret keys. In 2012, Lai et al. [26] proposed a novel three-party key agreement protocol using the enhanced Chebyshev chaotic map, but Zhao et al. [27] showed that their protocol is vulnerable to the privileged insider attack and the off-line password guessing attack. Lee et al. [28] also proposed a three-party key agreement protocol using extended chaotic maps, but these schemes need the timestamp. In recent years, the three-party password-authenticated key agreement protocol using modular exponentiation or scalar multiplication on an elliptic curve has been addressed widely [29, 30]. However, these schemes need heavy computation costs.

To the best of our knowledge, no three-party password-authenticated key agreement (3PAKA) protocol without a timestamp, that utilizes chaotic maps has been proposed, yet. Generally speaking, a 3PAKA protocol with chaotic maps should achieve the following requirements:

(i) It should allow two users establish a secure session key over an insecure communication channel with the help of a trusted server with the shared passwords.

(ii) The protocol should be based on chaotic maps without using modular exponentiation and scalar multiplication on an elliptic curve.

(iii) The protocol should be able to resist all attacks, such as password guessing attacks, impersonation attacks, man-in-the-middle attacks, etc.

(iv) The protocol should achieve some well-known properties, such as perfect forward secrecy, no timestamp, and execution efficiency.

In this paper, based on Chebyshev chaotic maps, we propose a new three-party password-authenticated key agreement protocol which achieves the above requirements.

The rest of the paper is organized as follows. In Sect. 2, we review Chebyshev chaotic maps. A Chebyshev chaotic maps-based three-party password-authenticated key agreement protocol is described in Sect. 3. After that, security analysis and the performance comparison are presented in Sects. 4 and 5. The paper is concluded in Sect. 6.

2 Chebyshev chaotic maps

In this section, we briefly introduce the Chebyshev polynomial and Chebyshev chaotic map [24].

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

$$T_n(x) = \cos\bigl(n \arccos(x)\bigr).$$

According to Definition 1, we can conclude the following recurrence relation of the Chebyshev polynomial:

$$T_0(x) = 1, \qquad T_1(x) = x,$$

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x),$$

where $n \geq 2$.

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

Definition 3 One of the most important properties of Chebyshev polynomials is called the semi-group property, namely

$$T_r\big(T_s(x)\big) = \cos\big(r\cos^{-1}\big(s\cos^{-1}(x)\big)\big) = \cos\big(rs\cos^{-1}(x)\big) = T_{sr}(x) = T_s\big(T_r(x)\big).$$

In 2008, Zhang [31] proved that the semi-group property defined on interval $(-\infty, +\infty)$ hold, that is,

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

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

$$T_r\big(T_s(x)\big) \equiv T_{sr}(x) \equiv T_s\big(T_r(x)\big) \bmod p.$$

The following problems about Chebyshev polynomials are assumed to be intractable within polynomial time.

Definition 4 Given x and y, it is intractable to find the integer s, such that $T_s(x) = y$. It is called the Chaotic Maps-Based Discrete Logarithm problem.

Definition 5 Given x, $T_r(x)$ and $T_s(x)$, it is intractable to find $T_{rs}(x)$. It is called the Chaotic Maps-Based Diffie–Hellman problem.

3 The proposed scheme

In this section, we propose a chaotic maps-based three-party password authenticated key agreement scheme which consists of two phases: the setup phase and the authentication and key agreement phase.

3.1 Setup phase

In this phase, a server S chooses its public key $(x, T_k(x))$ and a secret key k based on Chebyshev chaotic maps, a chaotic maps-based one-way hash function $h()$ [32], secure symmetric encryption/decryption functions $E_K()/D_K()$ with key K. Additionally, the server S shares passwords pw_A and pw_B with users A and B; users A and B choose their identities ID_A and ID_B, respectively.

3.2 Authentication and key agreement phase

In this phase, users A and B can authenticate each other and establish a session key with the help of the trusted server S. Figure 1 illustrates this phase.

Round 1 User A chooses a, computes $K_{AS} = T_aT_k(x)$, $H_A = h(T_a(x)\|ID_A\|ID_B\|pw_A)$, and $C_1 = E_{K_{AS}}(ID_A\|ID_B\|H_A)$. Then sends $m_1 = \{T_a(x), ID_A, C_1\}$ to B.

Round 2 Upon receiving m_1 from A, B chooses b, computes $K_{BS} = T_bT_k(x)$, $H_B = h(T_b(x)\|ID_B\|ID_A\|pw_B)$, and $C_2 = E_{K_{BS}}(ID_B\|ID_A\|H_B)$. Then, B sends $m_2 = \{m_1, T_b(x), ID_B, C_2\}$ to S.

Round 3 Upon receiving m_2 from B, S first computes

$$K_{SA} = T_kT_a(x), \qquad D_{K_{SA}}(C_1) = \{ID_A, ID_B, H_A\},$$
$$K_{SB} = T_kT_b(x), \qquad D_{K_{SB}}(C_2) = \{ID_B, ID_A, H_B\}.$$

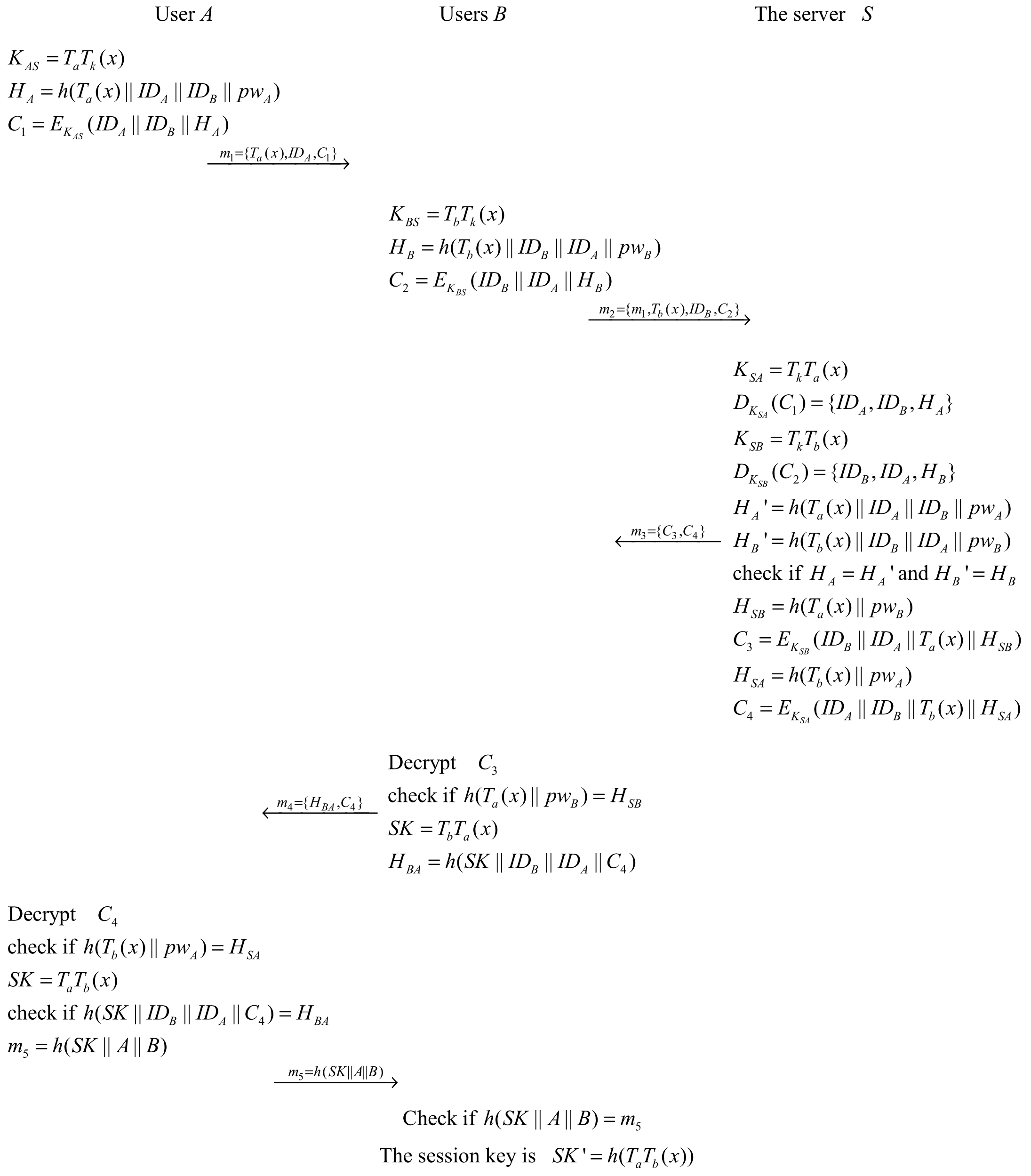
Then, S computes $H'_A = h(T_a(x)\|ID_A\|ID_B\|pw_A)$ and $H'_B = h(T_b(x)\|ID_B\|ID_A\|pw_B)$, and checks if $H_A = H'_A$ and $H'_B = H_B$. If so, S computes

$$H_{SB} = h\big(T_a(x)\|pw_B\big),$$
$$C_3 = E_{K_{SB}}\big(ID_B\|ID_A\|T_a(x)\|H_{SB}\big),$$
$$H_{SA} = h\big(T_b(x)\|pw_A\big),$$
$$C_4 = E_{K_{SA}}\big(ID_A\|ID_B\|T_b(x)\|H_{SA}\big),$$

and sends $m_3 = \{C_3, C_4\}$ to B. Otherwise, S terminates this request.

Round 4 When$B$ obtains m_3, he decrypts C_3 by K_{BS} and obtains $\{ID_B, ID_A, T_a(x), H_{SB}\}$, then computes $h(T_a(x)\|pw_B)$ and checks if it equals H_{SB}. If not, he terminates it. Otherwise, B computes $SK = T_bT_a(x)$ and $H_{BA} = h(SK\|ID_B\|ID_A\|C_4)$, and sends $m_4 = \{H_{BA}, C_4\}$ to A.

Round 5 Upon receiving m_4, A decrypts C_4 by K_{AS} and obtains $\{ID_B, ID_A, T_b(x), H_{SA}\}$, then computes $h(T_b(x)\|pw_A)$ and checks if it equals H_{SA}. If not, he terminates it. Otherwise, A computes $SK = T_aT_b(x)$ and verifies if $h(SK\|ID_B\|ID_A\|C_4)$ equals H_{BA}. If not, he terminates it. Otherwise, A computes and sends $m_5 = h(SK\|A\|B)$ to B.



User A Users B The server S

$K_{AS} = T_a T_k(x)$

$H_A = h(T_a(x) \| ID_A \| ID_B \| pw_A)$

$C_1 = E_{K_{AS}}(ID_A \| ID_B \| H_A)$

$\xrightarrow{m_1=\{T_a(x), ID_A, C_1\}}$

$K_{BS} = T_b T_k(x)$

$H_B = h(T_b(x) \| ID_B \| ID_A \| pw_B)$

$C_2 = E_{K_{BS}}(ID_B \| ID_A \| H_B)$

$\xrightarrow{m_2=\{m_1, T_b(x), ID_B, C_2\}}$

$K_{SA} = T_k T_a(x)$

$D_{K_{SA}}(C_1) = \{ID_A, ID_B, H_A\}$

$K_{SB} = T_k T_b(x)$

$D_{K_{SB}}(C_2) = \{ID_B, ID_A, H_B\}$

$H_A{}' = h(T_a(x) \| ID_A \| ID_B \| pw_A)$

$\xleftarrow{m_3=\{C_3, C_4\}}$ $H_B{}' = h(T_b(x) \| ID_B \| ID_A \| pw_B)$

check if $H_A = H_A{}'$ and $H_B{}' = H_B$

$H_{SB} = h(T_a(x) \| pw_B)$

$C_3 = E_{K_{SB}}(ID_B \| ID_A \| T_a(x) \| H_{SB})$

$H_{SA} = h(T_b(x) \| pw_A)$

$C_4 = E_{K_{SA}}(ID_A \| ID_B \| T_b(x) \| H_{SA})$

Decrypt C_3

$\xleftarrow{m_4=\{H_{BA}, C_4\}}$ check if $h(T_a(x) \| pw_B) = H_{SB}$

$SK = T_b T_a(x)$

$H_{BA} = h(SK \| ID_B \| ID_A \| C_4)$

Decrypt C_4

check if $h(T_b(x) \| pw_A) = H_{SA}$

$SK = T_a T_b(x)$

check if $h(SK \| ID_B \| ID_A \| C_4) = H_{BA}$

$m_5 = h(SK \| A \| B)$

$\xrightarrow{m_5=h(SK \| A \| B)}$

Check if $h(SK \| A \| B) = m_5$

The session key is $SK' = h(T_a T_b(x))$

Fig. 1 The proposed 3PAKA protocol

Round 6 When B obtains m_5, he verifies whether $m_5 = h(SK \| A \| B)$ or not. If it does not hold, B terminates it. Otherwise, A and Bshare the session key $SK' = h(T_a T_b(x))$.

4 Security analysis

In this section, we will show that the proposed scheme can resist various known attacks.

4.1 Off-line password guessing attack

An adversary may eavesdrop the communication among A, B and S, get all messages $\{m_1, m_2, m_3, m_4, m_5\}$, and launch an off-line password guessing attack. As we know, all messages except m_5 are related with A's or B's passwords. However, the adversary cannot verify whether his guessed password is right or not since these messages are all encrypted by K_{AS} or

K_{BS}, and the adversary cannot compute K_{AS} or K_{BS} from $T_k(x)$, $T_a(x)$ and $T_b(x)$ due to the intractability of the Chaotic Maps-Based Computational Diffie–Hellman (CM-CDH) problem. On the other hand, if an adversary can know K_{AS} or K_{BS}, then he can decrypt C_1, C_2, C_3, or C_4, and obtain H_A, H_B, H_{SA}, H_{SB}, respectively, thus the adversary can guess the correct passwords pw_A or pw_B. However, it is impossible due to the CM-CDH problem. Therefore, our scheme can resist an off-line password guessing attack.

4.2 On-line password guessing attack

If an adversary wants to guess B's password, he guesses a password pw, chooses b', computes $K'_{BS} = T_{b'}T_k(x)$, $H'_B = h(T_{b'}(x)\|ID_B\|ID_A\|pw)$, and $C'_2 = E_{K'_{BS}}(ID_B\|ID_A\|H'_B)$. Then the adversary intercepts and sends $m'_2 = \{m_1, T_{b'}(x), ID_B, C'_2\}$ to S. However, S can detect this attack if the adversary guessed a wrong password since S needs to check if $H'_B = h(T_{b'}(x)\|ID_B\|ID_A\|pw_B)$. Therefore, our scheme can resist an on-line password guessing attack.

4.3 Perfect forward secrecy

In the proposed scheme, the session key $SK' = h(T_aT_b(x))$ is related with nonces a and b, which were chosen by user A and user B, respectively. Because of the intractability of the CM-CDH problem, an adversary cannot compute the previously established session keys even if he knows the server's secret key k, A's and B's passwords.

4.4 Known-key security

Since the session key $SK' = h(T_aT_b(x))$ is depended on the random nonces a and b, and the generation of nonces is independent in all sessions, an adversary cannot compute the previous and the future session keys when he knows one session key.

4.5 Denning–Sacco attack

If an adversary knows $SK = T_aT_b(x)$ from $SK' = h(SK) = h(T_aT_b(x))$, than he can generate the correct H_{BA} and m_5, thus he can impersonate one party to cheat another. However, it is impossible due to the intractability of the solving Hash function. Also, the adversary cannot get the server's secret key k, A's and B's passwords from $SK' = h(T_aT_b(x))$.

4.6 Replay attack

Even if an adversary impersonates A and replays A's message $m_1 = \{T_a(x), ID_A, C_1\}$ to B, he cannot verify H_{BA} and respond the correct m_5 to B since the adversary cannot decrypt C_4, and cannot compute SK for the new nonce b. For the same reason, even if an adversary impersonates B and replays B's message $\{T_b(x), ID_B, C_2\}$ to the server, he cannot succeed.

If an adversary replays the server's messages C_3 and C_4, because a and b are new nonces chosen by A and B, the replayed C_3 and C_4 cannot pass the verification process of both A and B.

4.7 Forgery and impersonation attacks

If an adversary impersonates A (or B) and sends $m_1 = \{T_a(x), ID_A, C_1\}$ (or $\{T_b(x), ID_B, C_2\}$) to B (or the server), the message cannot pass through the authentication of the server since he does not know the password.

4.8 Man-in-the-middle attack

From the above analysis, we can know that an adversary is unable to achieve success by impersonating and replaying. On the other hand, because C_1, C_2, C_3, and C_4 contain the users' identities, a man-in-the-middle attack cannot succeed.

5 Performance comparison

In our proposed scheme, no time-consuming modular exponentiation and scalar multiplication on elliptic curves are needed, instead Chebyshev polynomial computations are needed. However, Xiao et al. [7] and Wang–Zhao [24] proposed several methods to solve the Chebyshev polynomial computation problem.

Let T, E, D, H, and M be the time for performing a Chebyshev polynomial computation, a modular exponentiation, a symmetric encryption/decryption, a one-way hash function, and a scalar multiplication on an elliptic curve, respectively. The performance comparison of authentication and key agreement phase between our scheme and other four recently proposed related schemes in [24, 25, 27–30] is given in Table 1.

From Table 1, we can conclude that both Wang–Zhao and Yoon–Jeon's protocols are more efficient

Table 1 Performance comparison of authentication and key agreement phase

	User A	User B	Server	Total
Wang–Zhao (2010) [24]	2T + 2D	2T + 3D	3D	4T + 8D
Yoon–Jeon (2011) [25]	2T + 2H + 1D	2T + 1D	2H + 2D	4T + 4H + 4D
Zhao et al. (2013) [27]	3T + 6H + 1D	3T + 6H + 1D	2T + 8H + 2D	8T + 20H + 4D
Lee et al. (2013) [28]	3T + 4H	3T + 4H	2T + 7H	8T + 15H
Yang–Cao (2012) [29]	6E + 3H	6E + 3H	10E + 4H	22E + 10H
Wu et al. (2012) [30]	4M + 5H + 1D	4M + 5H + 1D	6M + 4H + 2D	14M + 14H + 4D
Our scheme	3T + 5H + 2D	3T + 5H + 2D	2T + 4H + 4D	8T + 14H + 8D

than the rest, but their protocols are not password based schemes, and the users should protect long-term secret keys. For 3PAKA schemes, the computation cost among our scheme, Zhao et al.'s, and Lee et al.'s schemes are of the same level, but both Zhao et al.'s and Lee et al.'s schemes use a timestamp; also Yang–Cao and Wu et al.'s schemes use modular exponentiation and scalar multiplication on an elliptic curve, respectively.

As far as the same security level is concerned, the key sizes for the elliptic curve on the finite field GF(2^{160}) and the composition modulo N are 160 bits and 1024 bits, respectively. From [33], we can conclude that the performances of the ECC and RSA do not differ until the larger key sizes (e.g., the composition modulo N is 7680 bits), where ECC outperforms RSA. On the other hand, the computation of $T_n(x)$ takes some iterations of the Chebyshev polynomial [24, 34], which consumes roughly similar time as when computing modular exponentiation. However, for the Chebyshev polynomial, r and s are of 900–1024 bits [24, 35], and p is less than 1024 bits, for example, p is 256 bits [36]. In this scenario, the Chebyshev polynomial computation is much faster than that of RSA exponentiation and ECC scalar multiplication, since the modulo for RSA is 1024 bits.

Let's consider the other scenario, equation $T_n(x) \equiv (2xT_{n-1}(x) - T_{n-2}(x)) \bmod p$ can be rewritten as

$$\begin{bmatrix} T_{n-1}(x) \\ T_n(x) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2x \end{bmatrix} \begin{bmatrix} T_{n-2}(x) \\ T_{n-1}(x) \end{bmatrix}$$

$$= A \begin{bmatrix} T_{n-2}(x) \\ T_{n-1}(x) \end{bmatrix} = A^{n-1} \begin{bmatrix} T_0(x) \\ T_1(x) \end{bmatrix},$$

matrix exponentiation can be performed effectively by the square and multiply algorithm, which is faster than that of the iteration algorithm [37]. It only takes 70 ms to compute $T_n(x) \bmod p$ using the above equation with GNU multiple precision libraries on an Intel Pentium 1700 MHz processor with 512 MB RAM, where N and P are 1024 bits long [35].

Therefore, the proposed 3PAKA scheme not only achieves security requirements, but also enjoys acceptable efficiency.

6 Conclusion

In this paper, we proposed a chaotic maps-based three-party password-authenticated key agreement scheme. To the best of our knowledge, this is the first chaotic maps-based three-party password-authenticated key agreement scheme without a timestamp. The scheme has many advantages: it achieves perfect forward secrecy, does not use a timestamp, modular exponentiation and scalar multiplication on an elliptic curve, can resist all known attacks such as password guessing attacks, impersonation attacks, man-in-the-middle attacks, etc, and meets almost all requirements of a 3PAKA protocol.

Acknowledgements This research was supported by National Natural Science Foundation of China (No. 61070153), the Major State Basic Research Development (973) Program of China (No. 2013CB834205), Natural Science Foundation of Zhejiang province (No. LZ12F02005), and Opening Fund of Top Key Discipline of Computer Software and Theory in Zhejiang Provincial Colleges at Zhejiang Normal University (No. ZSDZZZZXK35).

References

1. Baptista, M.S.: Cryptography with chaos. Phys. Lett. A **240**, 50–54 (1998)
2. Özkaynak, F., Yavuz, S.: Designing chaotic S-boxes based on time-delay chaotic system. Nonlinear Dyn. (2013). doi:10.1007/s11071-013-0987-4

3. Behnia, S., Akhshani, A., Ahadpour, S., Mahmodi, H., Akhavan, A.: A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps. Phys. Lett. A **366**, 391–396 (2007)
4. Hussain, I., Shah, T., Gondal, M.: A novel approach for designing substitution-boxes based on nonlinear chaotic algorithm. Nonlinear Dyn. **70**, 1791–1794 (2012)
5. Hussain, I., Shah, T., Gondal, M., Mahmood, H.: An efficient approach for the construction of LFT S-boxes using chaotic logistic map. Nonlinear Dyn. **71**, 133–140 (2013)
6. Khan, M., Shah, T., Mahmood, H., Gondal, M.: An efficient method for the construction of block cipher with multi-chaotic systems. Nonlinear Dyn. **71**, 489–492 (2013)
7. Xiao, D., Liao, X., Wong, K.: An efficient entire chaos-based scheme for deniable authentication. Chaos Solitons Fractals **23**, 1327–1331 (2005)
8. Alvarez, G.: Security problems with a chaos-based deniable authentication scheme. Chaos Solitons Fractals **26**, 7–11 (2005)
9. Xiao, D., Liao, X., Deng, S.: A novel key agreement protocol based on chaotic maps. Inf. Sci. **177**, 1136–1142 (2007)
10. Han, S.: Security of a key agreement protocol based on chaotic maps. Chaos Solitons Fractals **38**, 764–768 (2008)
11. Xiang, T., Wong, K., Liao, X.: On the security of a novel key agreement protocol based on chaotic maps. Chaos Solitons Fractals **40**, 672–675 (2009)
12. Xiao, D., Liao, X., Deng, S.: Using time-stamp to improve the security of a chaotic maps-based key agreement protocol. Inf. Sci. **178**, 1598–11602 (2008)
13. Han, S., Chang, E.: Chaotic map based key agreement with/out clock synchronization. Chaos Solitons Fractals **39**, 1283–1289 (2009)
14. Guo, X., Zhang, J.: Secure group key agreement protocol based on chaotic Hash. Inf. Sci. **180**, 4069–4074 (2010)
15. He, D.: Cryptanalysis of a key agreement protocol based on chaotic Hash. eprint.iacr.org/2011/333.pdf
16. Gong, P., Li, P., Shi, W.: A secure chaotic maps-based key agreement protocol without using smart cards. Nonlinear Dyn. **70**, 2401–2406 (2012)
17. Tseng, H., Jan, R., Yang, W.: A chaotic maps-based key agreement protocol that preserves user anonymity. In: IEEE International Conference on Communications (ICC09), pp. 1–6 (2009)
18. Niu, Y., Wang, X.: An anonymous key agreement protocol based on chaotic maps. Commun. Nonlinear Sci. Numer. Simul. **16**(4), 1986–1992 (2011)
19. Xue, K., Hong, P.: Security improvement on an anonymous key agreement protocol based on chaotic maps. Commun. Nonlinear Sci. Numer. Simul. **17**, 2969–2977 (2012)
20. Yoon, E.: Efficiency and security problems of anonymous key agreement protocol based on chaotic maps. Commun. Nonlinear Sci. Numer. Simul. **17**, 2735–2740 (2012)
21. Tan, Z.: A chaotic maps-based authenticated key agreement protocol with strong anonymity. Nonlinear Dyn. **72**, 311–320 (2013)
22. Lee, C., Hsu, C.: A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps. Nonlinear Dyn. **71**, 201–211 (2013)
23. Guo, C., Chang, C.C.: Chaotic maps-based password-authenticated key agreement using smart cards. Commun. Nonlinear Sci. Numer. Simul. (2012). doi:10.1016/j.cnsns.2012.09.032
24. Wang, X., Zhao, J.: An improved key agreement protocol based on chaos. Commun. Nonlinear Sci. Numer. Simul. **15**, 4052–4057 (2010)
25. Yoon, E., Jeon, I.: An efficient and secure Diffie–Hellman key agreement protocol based on Chebyshev chaotic map. Commun. Nonlinear Sci. Numer. Simul. **16**, 2383–2389 (2011)
26. Lai, H., Xiao, J., Li, L., Yang, Y.: Applying semigroup property of enhanced Chebyshev polynomials to anonymous authentication protocol. Math. Probl. Eng. (2012). doi:10.1155/2012/454823
27. 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. (2013). doi:10.1007/s11071-013-0979-4
28. Lee, C., Li, C., Hsu, C.: A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps. Nonlinear Dyn. **73**, 125–132 (2013)
29. Yang, J., Cao, T.: Provably secure three-party password authenticated key exchange protocol in the standard model. J. Syst. Softw. **85**, 340–350 (2012)
30. Wu, S., Chen, K., Pu, Q., Zhu, Y.: Cryptanalysis and enhancements of efficient three-party password-based key exchange scheme. Int. J. Commun. Syst. (2012). doi:10.1002/dac.1362
31. Zhang, L.: Cryptanalysis of the public key encryption based on multiple chaotic systems. Chaos Solitons Fractals **37**(3), 669–674 (2008)
32. Xiao, D., Shih, F., Liao, X.: A chaos-based hash function with both modification detection and localization capabilities. Commun. Nonlinear Sci. Numer. Simul. **15**, 2254–2261 (2010)
33. Jansma, N., Arrendondo, B.: Performance comparison of elliptic curve and RSA digital signatures. http://www.nicj.net/files/performance_comparison_of_elliptic_curve_and_rsa_digital_signatures.pdf (2013/5/1)
34. Bergamo, P., D'Arco, P., Santis, A., Kocarev, L.: Security of public key cryptosystems based on Chebyshev polynomials. IEEE Trans. Circuits Syst. I **52**, 1382–1393 (2005)
35. Kocarev, L., Lian, S.: Chaos-Based Cryptography: Theory, Algorithms and Applications, pp. 53–54. Springer, Berlin (2011)
36. Pareek, N.K., Patidar, V., Sud, K.K.: Discrete chaotic cryptography using external key. Phys. Lett. A **309**, 75–82 (2003)
37. Li, Z., Cui, Y., Xu, H.: Fast algorithms of public key cryptosystem based on Chebyshev polynomials over finite field. J. China Univ. Post Telecommun. **18**(2), 86–93 (2011)