Soft Comput (2018) 22:2495–2506
https://doi.org/10.1007/s00500-017-2504-z

METHODOLOGIES AND APPLICATION

A novel three-party password-based authenticated key exchange protocol with user anonymity based on chaotic maps

Chun-Ta Li[1] · **Chin-Ling Chen**[2,3] · **Cheng-Chi Lee**[4,5] · **Chi-Yao Weng**[6] · **Chien-Ming Chen**[7]

Published online: 31 January 2017

Abstract Three-party authenticated key exchange (3PAKE) protocol allows two communication users to authenticate each other and to establish a secure common session key with the help of a trusted remote server. Recently, Farash and Attari propose an efficient and secure 3PAKE protocol based on Chebyshev chaotic maps and their protocol is supported by the formal proof in the random oracle model. However, in this paper, we analyze the security of Farash–Attari's protocol and show that it fails to resist password disclosure attack if the secret information stored in the server side is compromised. In addition, their protocol is insecure against user impersonation attack and the server is not aware of having caused problem. Moreover, the password change phase is insecure to identify the validity of request where insecurity in password change phase can cause offline password guessing attacks and is not easily reparable. To remove these security weaknesses, based on Chebyshev chaotic maps and quadratic residues, we further design an improved protocol for 3PAKE with user anonymity. In comparison with the existing chaotic map-based 3PAKE protocols, our proposed 3PAKE protocol is more secure with acceptable computation complexity and communication overhead.

Keywords Chebyshev chaotic maps · Quadratic residues · Password security · Three-party authenticated key exchange · User anonymity

Communicated by V. Loia.

✉ Chin-Ling Chen
clc@mail.cyut.edu.tw

✉ Cheng-Chi Lee
cclee@mail.fju.edu.tw

Chun-Ta Li
th0040@mail.tut.edu.tw

Chi-Yao Weng
cyweng@mail.nptu.edu.tw

Chien-Ming Chen
chienming.taiwan@gmail.com

[1] Department of Information Management, Tainan University of Technology, 529 Zhongzheng Road, Tainan City 71002, Taiwan, ROC

[2] Department of Computer Science and Information Engineering, Chaoyang University of Technology, 168 Jifeng East Road, Taichung City 41349, Taiwan, ROC

[3] School of Information Engineering, Changchun University of Technology, Changchun City, Jilin Province 130600, People's Republic of China

[4] Department of Library and Information Science, Fu Jen Catholic University, 510 Jhongjheng Road, New Taipei City 24205, Taiwan, ROC

[5] Department of Photonics and Communication Engineering, Asia University, 500 Lioufeng Road, Taichung City 41354, Taiwan, ROC

[6] Department of Computer Science, National Pingtung University, No. 4-18, Min-Sheng Road, Pingtung City 90003, Taiwan, ROC

[7] Harbin Institute of Technology Shenzhen Graduate School, Shenzhen University Town, Nanshan District, Shenzhen 518055, People's Republic of China

1 Introduction

With the rapid development of information and network technologies, user authentication plays an important role to protect resources or services from being accessed by unauthorized users (Brindha and Shaji 2016; He et al. 2015; He and Zeadally 2015; Sk and Islam 2015; Khan 2009; Khan and Kumari 2013; Li and Hwang 2010; Li et al. 2013b, 2015,

2016a; Mishra et al. 2015; Ramasamy and Muniyandi 2012; Wu et al. 2015; Yang et al. 2012). A three-party password-based authenticated key exchange (3PAKE) protocol allows two users over insecure channels negotiate a secure session key and establish a secure channel via the help of the authentication server for securing their subsequent communications. All legal users store their verifiers computed from their actual password in remote server's database and each user only needs to remember a single password with the trusted server. The main advantage of 3PAKE protocol is that it provides a convenient way for large-scale user-to-user communication environments and each user does not need to remember various passwords for different users who communicate with. Moreover, 3PAKE protocol can be applied for various electronic applications such as eBay.com and JobSearch International, etc. A trusted server assists in transactions between seller and buyer in eBay platform or a third party assists in employments between employer and employee in JobSearch Web site. Then these two users can exchange sensitive transactions or electronic job records securely and conveniently, as shown in Fig. 1.

Recently, due to the excellent properties of diffusion and confusion, the Chebyshev chaotic map has been used in the design of cryptographic protocols, especially secret key and public key cryptosystems. Many chaotic maps 3PAKE protocols have been proposed (Farash and Attari 2014; Lai et al. 2012; Lee et al. 2013; Wang and Zhao 2010; Xie et al. 2013; Yoon and Jeon 2011; Zhao et al. 2013). For example, Wang and Zhao (2010) proposed a chaotic map-based three-party key agreement protocol. However, Yoon and Jeon (2011) pointed out that Wang–Zhao's protocol is vulnerable to message modification attack. In addition, Lai et al. (2012) proposed an anonymous authentication protocol using the extended Chebyshev chaotic map, but Zhao et al. (2013) showed that Lai et al.'s protocol is vulnerable to privileged-insider attack and offline password guessing attack. Both Lai et al.'s and Zhao et al.'s 3PAKE protocols used smart card to store sensitive information and these protocols may cause lost/stolen smart card problems and the sensitive information stored in smart card can be extracted by using power analysis attacks and side channel attacks. Therefore, Lee et al. (2013) and Xie et al. (2013) presented the extended chaotic map-based 3PAKE protocol without using smart card. On ther other hand, 3PAKE protocol using modular exponentiation and symmetric-key cryptosystem has been addressed widely (Lin and Lee 2014; Lv et al. 2013), but these protocols are not practical due to heavy computation costs.

In order to design a secure and efficient chaotic map-based 3PAKE protocol without using smart cards, Farash and Attari (2014) proposed a provably 3PAKE protocol and the security of their protocol is proved in the random oracle model, which uses neither server's public key nor symmetric-key cryptosystems. Unfortunately, in this paper, we find that

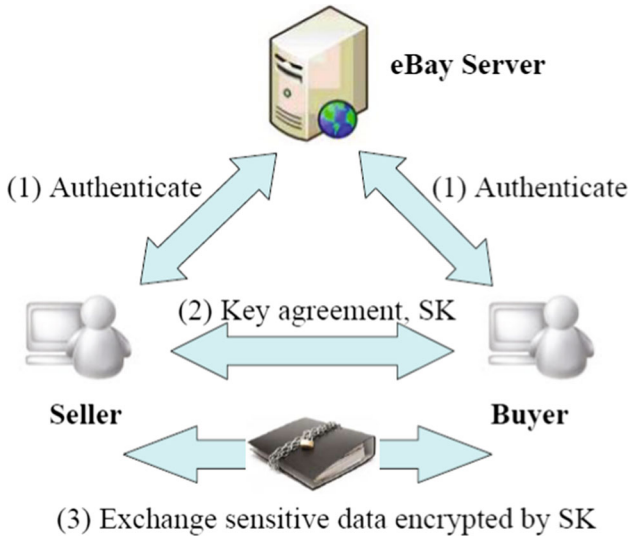

Fig. 1 An example of three-party authentication protocol for data exchange in eBay.com

Farash–Attari's 3PAKE protocol is vulnerable to the password disclosure attack, user impersonation attack and offline password-guessing attack. To enhance security, we present an improved version of Farash–Attari's 3PAKE protocol using Chebyshev chaotic maps and quadratic residues. Our extended 3PAKE protocol not only overcomes security weaknesses in their protocol but also provides user anonymity. The property of user anonymity (He et al. 2013, 2016a, b; Li and Lee 2012; Li 2013; Li et al. 2013c; Li 2016; Li et al. 2016b) means that a user's true identity and transmitted packets during the login session cannot be traced or linked by any outsiders.

The remainder of the paper is organized as follows. In Sect. 2, we introduce the mathematical preliminaries of Chebyshev polynomial problem and quadratic residue problem. Section 3 reviews Farash–Attari's 3PAKE protocol and gives the cryptanalysis of Farash–Attari's 3PAKE protocol in Sect. 4. Our 3PAKE protocol with user anonymity is proposed in Sect. 5. Security analysis of our proposed 3PAKE protocol is presented in Sect. 6. Section 7 compares the proposed 3PAKE protocol with related protocols in terms of efficiency and functionality. Finally, the conclusion is given in Sect. 8.

2 Mathematical preliminaries

In this section, we give some basic knowledge about the Chebyshev polynomial problem and quadratic residue problem. More details could be found in Bergamo et al. (2005), Chen et al. (2008, 2013), He et al. (2012), Li et al. (2013c), Wen (2014).

2.1 Chebyshev polynomial problem

1. Discrete logarithm problem (DLP): Given two elements x and y, the task of DLP is to find the integer r, such that $T_r(x) = y$, where the Chebyshev polynomial $T_r(x)$ is a polynomial in x of degree r and x be a variable taking value over the interval $[-1, 1]$. It means that the probability of any polynomial-time algorithm to solve DLP is negligible.
2. Computational Diffie–Hellman problem (CDHP): Given three elements x, $T_r(x)$ and $T_s(x)$, it is difficult to compute the value $T_{rs}(x)$. It means that the probability of any polynomial-time algorithm to solve CDHP is negligible.

2.2 Quadratic residue problem

We assume that $N = x \times y$ and $b = a^2 \bmod N$ has a solution, where x and y are two large primes. There exists a square root for b, then b is called a quadratic residue mod$N$. The set of all quadratic residue numbers in $[1, N-1]$ is denoted by QR_N and the quadratic residue problem means that for $b \in QR_N$, it is hard to find a without knowing the knowledge of x and y due to the difficulty of factoring N.

3 Review of Farash–Attari's 3PAKE Protocol

In this section, Farash–Attari's 3PAKE protocol (Farash and Attari 2014) will be briefly reviewed. There are four phases in Farash–Attari's protocol: system setup, registration, authentication and key exchange, and password change. For convenience of description, terminology and notations used in the paper are summarized as follows:

- A and B: Two communication users.
- S: The remote server.
- ID_i: The identity of U_i.
- pw_i: The password of U_i.
- $H_1(\cdot)$: A hash function (Aboshosha et al. 2016; Drissi and Asimi 2017), $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p$, where p is a large prime.
- $H(\cdot)$: A hash function (National Institute of Standards and Technology 2002) and $H : \{0,1\}^* \rightarrow \{0,1\}^l$, where l is the secure parameter size.
- s: S's long-live secret key, which is kept secret and only known by S.
- α: A pubic parameter, where $\alpha \in \mathbb{Z}_p$ such that the minimal period of Chebyshev polynomial sequence $(T_n(\alpha) \bmod p)_{n>0}$ is $p+1$.
- SK_{ij}: The session key, which is established between entity i and entity j.

3.1 System setup phase

In this phase, the remote server S keeps the secret key s and publishes the parameters $\{p, \alpha, H_1(\cdot), H(\cdot)\}$.

3.2 Registration phase

In this phase, the user registers with the remote server S through a secure channel to be a legal user. The details of registration phase are as follows:

Step R1 The user chooses his/her identity ID_i and password pw_i and computes $PW_i = T_{pw_i}(\alpha) \bmod p$. Then user sends the registration request $\{ID_i, pw_i\}$ to S.

Step R2 The remote server S computes $VPW_i = H(ID_i, s) + PW_i \bmod p$ and stores (ID_i, VPW_i) in its database, and zeroizes PW_i.

3.3 Authentication and key exchange phase

When the users A and B want to authenticate the validity of each other and establish a common session key, they must perform the following steps with S to execute a session of the protocol:

Step A1 The user A chooses a random number $r_A \in [1, p+1]$ and computes $R_A = T_{r_A}(\alpha) \bmod p$. Then A sends $\{ID_A, ID_B, R_A\}$ to S.

Step A2 Upon receiving $\{ID_A, ID_B, R_A\}$ from A, S chooses two random numbers $r_{S1}, r_{S2} \in [1, p+1]$ and computes $R_{S1} = T_{r_{S1}}(\alpha) - PW_A \bmod p$ and $R_{S2} = T_{r_{S2}}(\alpha) - PW_B \bmod p$. Then S sends $\{ID_A, R_A, R_{S2}\}$ to B.

Step A3 Upon receiving $\{ID_A, R_A, R_{S2}\}$ from S, B chooses a random number $r_B \in [1, p+1]$ and computes $R_B = T_{r_B}(\alpha) \bmod p$, $K_{BS} = T_{r_B}(R_{S2} + PW_B) \bmod p = T_{r_B r_{S2}}(\alpha) \bmod p$, $K_{BA} = T_{r_B}(R_A) \bmod p = T_{r_B r_A}(\alpha) \bmod p$, $Z_{BA} = H(0, ID_B, ID_A, R_B, R_A, K_{BA})$ and $Z_{BS} = H(0, ID_B, ID_A, R_B, R_{S2}, Z_{BA}, K_{BS})$. Then B sends $\{R_B, Z_{BS}, Z_{BA}\}$ to S.

Step A4 Upon receiving $\{R_B, Z_{BS}, Z_{BA}\}$ from B, S computes $K_{SB} = T_{r_{S2}}(R_B) = T_{r_B r_{S2}}(\alpha) \bmod p$ and verifies if computed $H(0, ID_B, ID_A, R_B, R_{S2}, Z_{BA}, K_{SB})$ equals received Z_{BS}. If it holds, S computes $K_{SA} = T_{r_{S1}}(R_A) = T_{r_{S1} r_A}(\alpha) \bmod p$ and $Z_{SA} = H(0, ID_A, ID_B, R_{S1}, R_A, R_B, Z_{BA}, K_{SA})$ and sends $\{R_{S1}, R_B, Z_{BA}, Z_{SA}\}$ to A.

Step A5 After receiving $\{R_{S1}, R_B, Z_{BA}, Z_{SA}\}$ from S, A computes $K_{AS} = T_{r_A}(R_{S1} + PW_A) = T_{r_A r_{S1}}(\alpha) \bmod p$ and verifies if computed $H(0, ID_A, ID_B, R_{S1}, R_A, R_B, Z_{BA}, K_{AS})$ equals received Z_{SA}.

If it holds, A computes $K_{AB} = T_{r_A(R_B)} \bmod p = T_{r_A r_B}(\alpha) \bmod p$ and verifies if computed $H(0, ID_B, ID_A, R_B, R_A, K_{BA})$ equals received Z_{BA}. If it holds, B is authenticated by A and A computes $Z_{AB} = H(1, ID_A, ID_B, R_A, R_B, K_{AB})$ and $Z_{AS} = H(1, ID_A, ID_B, R_A, R_{S1}, Z_{AB}, K_{AS})$. Then A sends $\{Z_{AS}, Z_{AB}\}$ to S.

Step A6 After receiving $\{Z_{AS}, Z_{AB}\}$ from A, S verifies if computed $H(1, ID_A, ID_B, R_A, R_{S1}, Z_{AB}, K_{SA})$ equals received Z_{AS}. If it holds, S computes $Z_{SB} = H(1, ID_A, ID_B, R_A, R_B, Z_{AB}, K_{SB})$ and sends $\{Z_{AB}, Z_{SB}\}$ to B.

Step A7 After receiving $\{Z_{AB}, Z_{SB}\}$ from S, B verifies if computed $H(1, ID_A, ID_B, R_A, R_B, Z_{AB}, K_{SB})$ and $H(1, ID_A, ID_B, R_A, R_B, K_{BA})$ equal received Z_{SB} and Z_{AB}, respectively. If they are valid, A is authenticated by B.

Finally, A computes the session key $SK_{AB} = H(2, ID_A, ID_B, R_A, R_B, K_{AB})$ and B computes the session key $SK_{BA} = H(2, ID_A, ID_B, R_A, R_B, K_{BA})$. Note that $K_{AB} = K_{BA} = T_{r_A r_B}(\alpha) \bmod p$ and $SK_{AB} = SK_{BA}$.

3.4 Password change phase

When the user A wants to change his/her old password pw_A to a new password pw_A^*, A must notify the remote server S to update the old password verifier $VPW_A = H(ID_A, s) + PW_A \bmod p$ to a new password verifier $VPW_A^* = H(ID_A, s) + PW_A^* \bmod p$, where $PW_A = T_{pw_A}(\alpha) \bmod p$ and $PW_A^* = T_{pw_A^*}(\alpha) \bmod p$.

Step C1 The user A randomly chooses a random number $r_A \in [1, p+1]$, computes $R_A = T_{r_A}(\alpha) \bmod p$ and sends $\{ID_A, R_A\}$ to S.

Step C2 Upon receiving $\{ID_A, R_A\}$ from A, S chooses a random number $r_S \in [1, p+1]$ and computes $R_S = T_{r_S}(\alpha) - PW_A \bmod p$, $K_{SA} = T_{r_S}(R_A) = T_{r_S r_A}(\alpha) \bmod p$ and $Z_{SA} = H(0, ID_A, R_S, R_A, K_{SA})$. Then S sends $\{R_S, Z_{SA}\}$ to A.

Step C3 Upon receiving $\{R_S, Z_{SA}\}$ from S, A computes $K_{AS} = T_{r_A}(R_S) = T_{r_S r_A}(\alpha) \bmod p$ and verifies if computed $H(0, ID_A, R_S, R_A, K_{AS})$ equals received Z_{SA}. If it holds, A computes $Z_{AS} = H(1, ID_A, R_A, R_S, K_{AS})$, $PW_A^* = T_{pw_A^*}(\alpha) \bmod p$, $PWD = H_1(K_{AS}, ID_A) + PW_A^* \bmod p$ and $V_3 = H(K_{AS}, PW_A^*)$ and sends $\{Z_{AS}, PWD, V_3\}$ to S, where pw_A^* is A's new password.

Step C4 Upon receiving password change request $\{Z_{AS}, PWD, V_3\}$ from A, S verifies if computed $H(1, ID_A, R_A, R_S, K_{SA})$ equals received Z_{AS}. If it holds, S computes $PW_A^* = PWD - H(K_{SA}, ID_A) \bmod p$ and verifies if computed $H(K_{SA}, PW_A^*)$ equals received V_3. If it holds, S accepts A's password change request, computes $R_1 = H(1, ID_A, PWD, V_3, K_{SA})$ and $VPW_A^* = H(ID_A, s) + PW_A^* \bmod p$ and replaces VPW_A with VPW_A^*. Then S sends {Accept, R_1} to A. Otherwise, S rejects A's password change request, computes $R_2 = H(0, ID_A, PWD, V_3, K_{SA})$ and sends {Reject, R_2} to A. If the message is {Accept, R_1}, A verifies if computed $H(1, ID_A, PWD, V_3, K_{AS})$ equals received R_1. If it holds, A confirms pw_A^* as the new password. Otherwise, A returns to **Step C1** and follows the process. If the message is {Reject, R_2}, A returns to **Step C1** with another new password and follows the process.

4 Weaknesses of Farash–Attari's 3PAKE protocol

In this section, we will show that Farash–Attari's protocol is vulnerable to password disclosure attacks, user impersonation attacks and offline password-guessing attacks.

4.1 Password disclosure attacks

In real environments, the user A may register with a number of servers by using a common password pw_A and the identity ID_A for his/her convenience. Thus, the privileged-insider of S may try to use the knowledge of A's pw_A and ID_A to access another servers. The details of password disclosure attack in Farash–Attari's protocol are described as follows:

Step 1 The privileged-insider of S steals the password verifier $PW_A = T_{pw_A}(\alpha) \bmod p$ from S's database.

Step 2 The privileged-insider of S guesses a password pw'_A and computes $PW'_A = T_{pw'_A}(\alpha) \bmod p$.

Step 3 The privileged-insider of S compares PW'_A with PW_A.

A match in **Step 3** above indicates the correct guessing of A's password and the privileged-insider of S succeeds to guess the low-entropy password $pw'_A = pw_A$. Otherwise, the privileged-insider of S repeats **Step 2**. Note that above-mentioned steps can be done by offline manner and S is not aware of having caused problem.

4.2 User impersonation attacks

In user impersonation attack, a malicious attacker C may try to impersonate the user A to spoof the remote server S and

the victim user B. The details of user impersonation attack in Farash–Attari's protocol are described as follows:

Step 1 The attacker C chooses a random number $r_C \in [1, p+1]$ and computes $R_C = T_{r_C}(\alpha) \bmod p$. Then C sends $\{ID_A, ID_B, R_C\}$ to S.

Step 2 Upon receiving $\{ID_A, ID_B, R_C\}$ from C, in **Step A2** of Farash–Attari's protocol, S sends $\{ID_A, R_C, R_{S2}\}$ to B.

Step 3 Upon receiving $\{ID_A, R_C, R_{S2}\}$ from S, in **Step A3** of Farash–Attari's protocol, the victim user B will send $\{R_B, Z_{BS}, Z_{BA}\}$ to S, where $R_B = T_{r_B}(\alpha) \bmod p$, $Z_{BA} = H(0, ID_B, ID_A, R_B, R_C, K_{BA})$, $K_{BA} = T_{r_B}(R_C) = T_{r_B r_C}(\alpha) \bmod p$, $Z_{BS} = H(0, ID_B, ID_A, R_B, R_{S2}, Z_{BA}, K_{BS})$ and $K_{BS} = T_{r_B}(R_{S2} + PW_B) = T_{r_B r_{S2}}(\alpha) \bmod p$.

Step 4 Upon receiving $\{R_B, Z_{BS}, Z_{BA}\}$ from B, in **Step A4** of Farash–Attari's protocol, S will sends $\{R_{S1}, R_B, Z_{BA}, Z_{SA}\}$ to C, where $R_{S1} = T_{r_{S1}}(\alpha) - PW_A \bmod p$, $Z_{SA} = H(0, ID_A, ID_B, R_{S1}, R_C, R_B, Z_{BA}, K_{SA})$ and $K_{SA} = T_{r_{S1}}(R_C) = T_{r_{S1} r_C}(\alpha) \bmod p$.

Step 5 Upon receiving $\{R_{S1}, R_B, Z_{BA}, Z_{SA}\}$ from S, C guesses a password pw'_A and computes $PW'_A = T_{pw'_A}(\alpha) \bmod p$ and $K'_{AS} = Tr_C(R_{S1} + PW'_A) \bmod p$. Then C verifies if computed $H(0, ID_A, ID_B, R_{S1}, R_C, R_B, Z_{BA}, K'_{AS})$ equals received Z_{SA}. If it matches, it indicates the correct guessing of A's password and the attacker C succeeds to guess the low-entropy password $pw'_A = pw_A$. Otherwise, C guesses another password until success. Note that **Step 5** can be done by offline manner.

Step 6 If **Step 5** is passed, it indicates that C knows $K'_{AS} = T_{R_C}(r_{S1}) = T_{r_C r_{S1}}(\alpha) \bmod p = K_{SA}$. Then C computes $K'_{AB} = Tr_C(R_B) = T_{r_C r_B}(\alpha) \bmod p$, $Z'_{AB} = H(1, ID_A, ID_B, R_C, R_B, K'_{AB})$ and $Z'_{AS} = H(1, ID_A, ID_B, R_C, R_{S1}, Z'_{AB}, K'_{AS})$ and sends $\{Z'_{AS}, Z'_{AB}\}$ to S.

Step 7 Upon receiving $\{Z'_{AS}, Z'_{AB}\}$ from C, in **Step A6** of Farash–Attari's protocol, C will pass S's verification and S will send $\{Z'_{AB}, Z'_{SB}\}$ to B, where $Z'_{SB} = H(1, ID_A, ID_B, R_C, R_B, Z'_{AB}, K_{SB})$ and $KSB = T_{r_{S2}}(R_B) = T_{r_{S2} r_B}(\alpha) \bmod p$.

Step 8 Upon receiving $\{Z'_{AB}, Z'_{SB}\}$ from S, in **Step A7** of Farash–Attari's protocol, B verifies if computed $H(1, ID_A, ID_B, R_C, R_B, Z'_{AB}, K_{SB})$ and $H(1, ID_A, ID_B, R_C, R_B, K_{BA})$ equal received Z'_{SB} and Z'_{AB}, respectively. If they are valid, C is authenticated by B.

Finally, C succeeded in impersonating A to spoof the remote server S and the victim user B. Moreover, C can easily establish the common session key $SK'_{AB} = SK'_{BA}$ shared between C and B and B is not aware of having caused problem, where $SK'_{AB} = H(2, ID_A, ID_B, R_C, R_B, K'_{AB}) = SK'_{BA}$ and $K'_{AB} = Tr_C(R_B) = T_{r_C r_B}(\alpha) \bmod p$.

4.3 Offline password-guessing attacks

In password change phase of Farash–Attari's protocol, the attacker C can impersonate a legitimate user and guess a legitimate user A's password with the help of achieved values from the remote server S. The details of offline password-guessing attack in Farash–Attari's protocol are described as follows:

Step 1 During the password change phase, the attacker C randomly chooses a random number $r_C \in [1, p+1]$, computes $R_C = T_{r_C}(\alpha) \bmod p$ and sends $\{ID_A, R_C\}$ to S.

Step 2 Upon receiving $\{ID_A, R_C\}$ from C, S chooses a random number $r_S \in [1, p+1]$ and computes $R_S = T_{r_S}(\alpha) - PW_A \bmod p$, $K_{SA} = T_{r_S}(R_C) = T_{r_S r_C}(\alpha) \bmod p$ and $Z_{SA} = H(0, ID_A, R_S, R_C, K_{SA})$. Then S sends $\{R_S, Z_{SA}\}$ to C.

Step 3 Upon receiving $\{R_S, Z_{SA}\}$ from S, C guesses a password pw'_A and computes $PW'_A = T_{pw'_A}(\alpha) \bmod p$, $T_{r'_S}(\alpha) \bmod p = R_S + PW'_A \bmod p$ and $K'_{AS} = T_{r_C}(T_{r'_S}) = T_{r_C r'_S}(\alpha) \bmod p$. Then C compares if computed $Z'_{AS} = H(0, ID_A, R_C, R_S, K'_{AS})$ equals received Z_{SA}.

A match in **Step 3** above indicates the correct guessing of A's password and the attacker C succeeds to guess the low-entropy password $pw'_A = pw_A$. Otherwise, C repeats **Step 3**. Note that above-mentioned steps can be done by offline manner and S is not aware of having caused problem.

5 The proposed 3PAKE protocol with user anonymity

This section proposes a simple improvement on Farash–Attari's protocol, which not only keeps the merits of original protocol but also resists the security weaknesses described in previous section. Moreover, we extend the proposed 3PAKE protocol to provide the user anonymity and two communication users' true identity cannot be traced by any outsiders over public channels. Figure 2 shows the flowchart of our proposed 3PAKE protocol with user anonymity.

5.1 System setup phase

In this phase, the remote server S keeps one secret key s and two secret numbers (x, y) and publishes the parameters

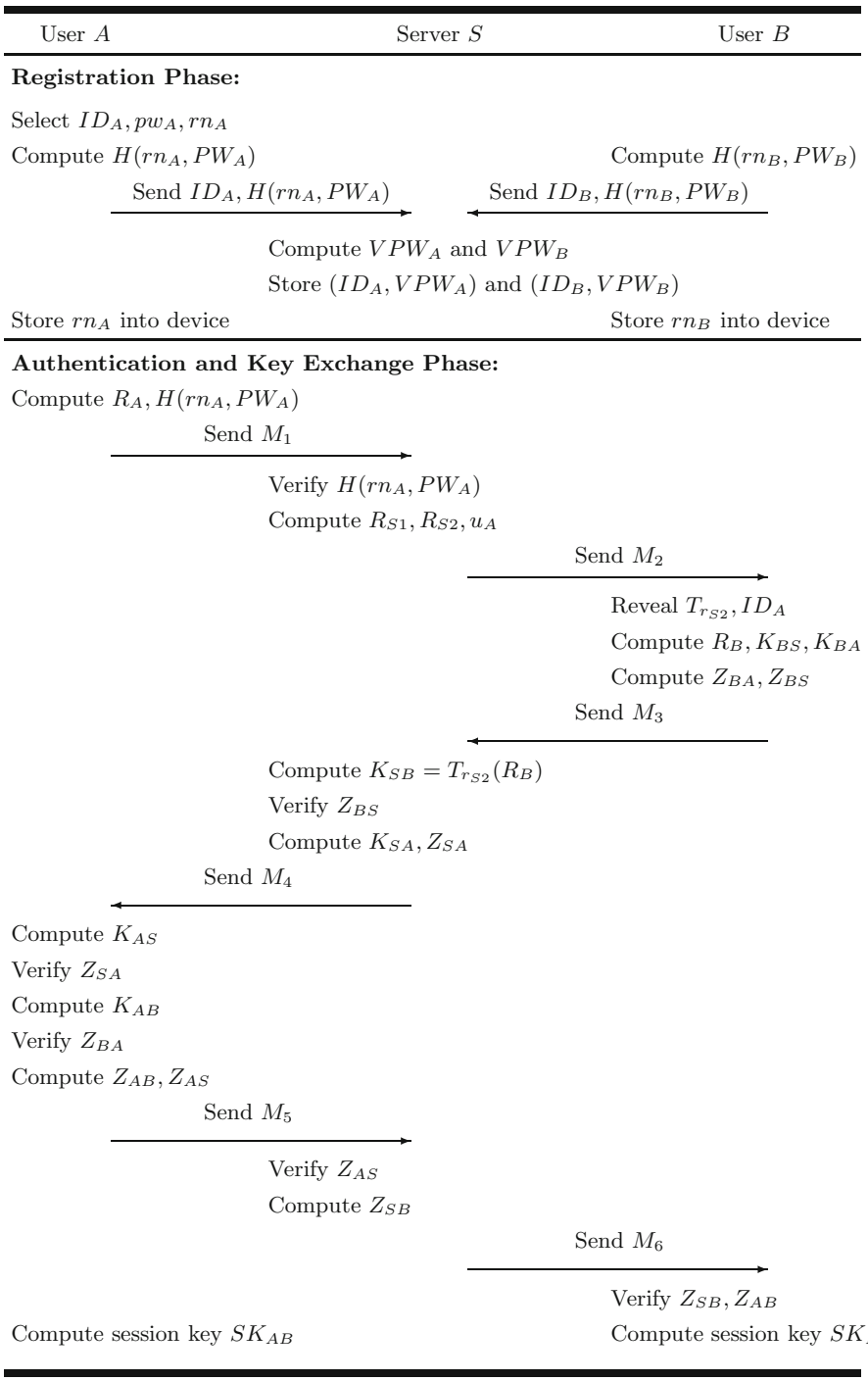

Fig. 2 The flowchart of our proposed 3PAKE protocol with user anonymity

$\{p, \alpha, H_1(\cdot), H(\cdot), N\}$, where the number $N = x \times y$ and (x, y) are two large primes maintained by S.

5.2 Registration phase

In this phase, the user registers with the remote server S through a secure channel to be a legal user. The details of registration phase are as follows:

Step R1 The user chooses his/her identity ID_i and password pw_i and computes $PW_i = T_{pw_i}(\alpha)$ mod p and $H(rn_i, PW_i)$, where $rn_i \in [1, p+1]$. Then user sends the registration request $\{ID_i, H(rn_i, PW_i)\}$ to S.

Step R2 The remote server S computes $VPW_i = H(ID_i, s) + H(rn_i, PW_i)$ mod p and stores (ID_i, VPW_i) in its database, and zeroizes $H(rn_i, PW_i)$.

Step R3 The user stores rn_i into his/her end-user device.

5.3 Authentication and key exchange phase

When the users A and B want to anonymously authenticate each other and establish a common session key, they must perform the following steps with S to execute a session of the protocol:

Step A1 The user A chooses a random number $r_A \in [1, p+1]$ and enters his/her password pw_A. Next, A computes $R_A = T_{r_A}(\alpha) \bmod p$, $PW_A = T_{pw_A}(\alpha) \bmod p$ and $H(rn_A, PW_A)$, where rn_A is retrieved from his/her end-user device. Then A sends $M_1 = (ID_A, ID_B, H(rn_A, PW_A), R_A)^2 \bmod N$ to S.

Step A2 Upon receiving M_1 from A, S reveals M_1 by using the Chinese Remainder Theorem with x and y to obtain $(ID_A, ID_B, H(rn_A, PW_A), R_A)$. Next, S verifies the revealed $H(rn_A, PW_A)$ with the stored $VPW_A = H(ID_A, s) + H(rn_A, PW_A) \bmod p$ corresponding to ID_A. If $VPW_A - H(ID_A, s) = H(rn_A, PW_A)$, S accepts A's request message M_1. Then S chooses two random numbers $r_{S1}, r_{S2} \in [1, p+1]$ and computes $R_{S1} = T_{r_{S1}}(\alpha) - H(rn_A, PW_A) \bmod p$, $R_{S2} = T_{r_{S2}}(\alpha) - H(rn_B, PW_B) \bmod p$ and $u_A = ID_A \oplus T_{r_{S2}}(\alpha) \bmod p$. Then S sends $M_2 = \{u_A, R_A, R_{S2}\}$ to B.

Step A3 Upon receiving M_2 from S, B enters his/her password pw_B and computes $PW_B = T_{pw_B}(\alpha) \bmod p$ and $H(rn_B, PW_B)$, where rn_B is retrieved from his/her end-user device. Next, B reveals $T_{r_{S2}}(\alpha) \bmod p$ and ID_A by computing $R_{S2} - H(rn_B, PW_B) \bmod p$ and $u_A \oplus T_{r_{S2}}(\alpha) \bmod p$, respectively. Then B chooses a random number $r_B \in [1, p+1]$ and computes $R_B = T_{r_B}(\alpha) \bmod p$, $K_{BS} = T_{r_B}(T_{r_{S2}}(\alpha)) \bmod p = T_{r_B r_{S2}}(\alpha) \bmod p$, $K_{BA} = T_{r_B}(R_A) \bmod p = T_{r_B r_A}(\alpha) \bmod p$, $Z_{BA} = H(0, ID_B, ID_A, R_B, R_A, K_{BA})$ and $Z_{BS} = H(0, ID_B, ID_A, R_B, R_{S2}, Z_{BA}, K_{BS})$. Then B sends $M_3 = (ID_B, ID_A, R_B, Z_{BS}, Z_{BA})^2 \bmod N$ to S.

Step A4 Upon receiving M_3 from B, S reveals M_3 by using the Chinese Remainder Theorem with x and y to obtain $(ID_B, ID_A, R_B, Z_{BS}, Z_{BA})$. S computes $K_{SB} = T_{r_{S2}}(R_B) = T_{r_B r_{S2}}(\alpha) \bmod p$ and verifies if computed $H(0, ID_B, ID_A, R_B, R_{S2}, Z_{BA}, K_{SB})$ equals received Z_{BS}. If it holds, B is authenticated by S and S computes $K_{SA} = T_{r_{S1}}(R_A) = T_{r_{S1} r_A}(\alpha) \bmod p$ and $Z_{SA} = H(0, ID_A, ID_B, R_{S1}, R_A, R_B, Z_{BA}, K_{SA})$. Then S sends $M_4 = \{R_{S1}, R_B, Z_{BA}, Z_{SA}\}$ to A.

Step A5: After receiving M_4 from S, A computes $K_{AS} = T_{r_A}(R_{S1} + H(rn_A, PW_A)) = T_{r_A r_{S1}}(\alpha) \bmod p$ and verifies if computed $H(0, ID_A, ID_B, R_{S1}, R_A, R_B, Z_{BA}, K_{AS})$ equals received Z_{SA}. If it holds, S is authenticated by A. Next, A computes $K_{AB} = T_{r_A(R_B)} \bmod p = T_{r_A r_B}(\alpha) \bmod p$ and verifies if computed $H(0, ID_B, ID_A, R_B, R_A, K_{AB})$ equals received Z_{BA}. If it holds, B is authenticated by A and A computes $Z_{AB} = H(1, ID_A, ID_B, R_A, R_B, K_{AB})$ and $Z_{AS} = H(1, ID_A, ID_B, R_A, R_{S1}, Z_{AB}, K_{AS})$. Then A sends $M_5 = \{Z_{AS}, Z_{AB}\}$ to S.

Step A6 After receiving M_5 from A, S verifies if computed $H(1, ID_A, ID_B, R_A, R_{S1}, Z_{AB}, K_{SA})$ equals received Z_{AS}. If it holds, A is authenticated by S. Then S computes $Z_{SB} = H(1, ID_A, ID_B, R_A, R_B, Z_{AB}, K_{SB})$ and sends $M_6 = \{Z_{AB}, Z_{SB}\}$ to B.

Step A7 After receiving M_6 from S, B verifies if computed $H(1, ID_A, ID_B, R_A, R_B, Z_{AB}, K_{SB})$ and $H(1, ID_A, ID_B, R_A, R_B, K_{BA})$ equal received Z_{SB} and Z_{AB}, respectively. If they are valid, A and S are authenticated by B.

Finally, A computes the session key $SK_{AB} = H(2, ID_A, ID_B, R_A, R_B, K_{AB})$ and B computes the session key $SK_{BA} = H(2, ID_A, ID_B, R_A, R_B, K_{BA})$. Note that $K_{AB} = K_{BA} = T_{r_A r_B}(\alpha) \bmod p$ and $SK_{AB} = SK_{BA}$.

5.4 Password change phase

When the user A wants to change his/her old password pw_A to a new password pw_A^*, A must notify the remote server S to update the old password verifier $VPW_A = H(ID_A, s) + H(rn_A, PW_A) \bmod p$ to a new password verifier $VPW_A^* = H(ID_A, s) + H(rn_A^*, PW_A^*) \bmod p$, where $PW_A = T_{pw_A}(\alpha) \bmod p$ and $PW_A^* = T_{pw_A^*}(\alpha) \bmod p$.

Step C1 The user A randomly chooses a random number $r_A \in [1, p+1]$ and enters his/her old password pw_A. Next, A computes $R_A = T_{r_A}(\alpha) \bmod p$, $PW_A = T_{pw_A}(\alpha) \bmod p$ and $H(rn_A, PW_A)$, where rn_A is retrieved from his/her end-user device. Then A sends $C_1 = (ID_A, H(rn_A, PW_A), R_A)^2 \bmod p$ to S.

Step C2 Upon receiving C_1 from A, S reveals C_1 by using the Chinese Remainder Theorem with x and y to obtain $(ID_A, H(rn_A, PW_A), R_A)$. Next, S verifies the revealed $H(rn_A, PW_A)$ with the stored $VPW_A = H(ID_A, s) + H(rn_A, PW_A) \bmod p$ corresponding to ID_A. If $VPW_A - H(ID_A, s) = H(rn_A, PW_A)$, S accepts A's request message C_1. Then S chooses a random number $r_S \in [1, p+1]$ and computes $R_S = T_{r_S}(\alpha) - H(rn_A, PW_A) \bmod p$, $K_{SA} = T_{r_S}(R_A) = T_{r_S r_A}(\alpha) \bmod p$ and

$Z_{SA} = H(0, ID_A, R_S, R_A, K_{SA})$. Then S sends $C_2 = \{R_S, Z_{SA}\}$ to A.

Step C3 Upon receiving C_2 from S, A computes $K_{AS} = T_{r_A}(R_S + H(rn_A, PW_A)) = T_{r_S r_A}(\alpha)$ mod p and verifies if computed $H(0, ID_A, R_S, R_A, K_{AS})$ equals received Z_{SA}. If it holds, S is authenticated by A. Next, A randomly selects a new password pw_A^* and a new random number rn_A^* and computes $Z_{AS} = H(1, ID_A, R_A, R_S, K_{AS})$, $PW_A^* = T_{pw_A^*}(\alpha)$ mod p, and $H(rn_A^*, PW_A^*)$. Then A sends $C_3 = (Z_{AS}, ID_A, H(rn_A^*, PW_A^*))^2$ mod p to S.

Step C4 Upon receiving C_3 from A, S verifies if computed $H(1, ID_A, R_A, R_S, K_{SA})$ equals received Z_{AS}. If it holds, S accepts A's password change request, computes $R_1 = H(1, ID_A, H(rn_A^*, PW_A^*), K_{SA})$ and $VPW_A^* = H(ID_A, s) + H(rn_A^*, PW_A^*)$ mod p and replaces VPW_A with VPW_A^*. Then S sends {Accept, R_1} to A. Otherwise, S rejects A's password change request, computes $R_2 = H(0, ID_A, H(rn_A^*, PW_A^*), K_{SA})$ and sends {Reject, R_2} to A. If the message is {Accept, R_1}, A verifies if computed $H(1, ID_A, H(rn_A^*, PW_A^*), K_{AS})$ equals received R_1. If it holds, A confirms pw_A^* as the new password and replaces rn_A with rn_A^* in his/her end-user device. Otherwise, A returns to **Step C1** and follows the process. If the message is {Reject, R_2}, A returns to **Step C1** with another new password and follows the process.

6 Analysis of the proposed 3PAKE protocol

In this section, we analyzed the proposed 3PAKE protocol in terms of security and functionality requirements. The details are described below.

Proposition 1 *The proposed 3PAKE protocol ensures anonymous interactions between the users (A, B) and the server S and no outsiders can ascribe any session to a particular user during authentication and key exchange phase.*

Proof In the authentication and key exchange phase of proposed protocol, A's real identity ID_A and B's real identity ID_B are implicitly involved in M_1 and M_3, where $M_1 = (ID_A, ID_B, H(rn_A, PW_A), R_A)^2$ mod N and $M_3 = (ID_B, ID_A, R_B, Z_{BS}, Z_{BA})^2$ mod N. If the outsider C would like to reveal ID_A and ID_B from M_1 and M_3, C needs to solve the quadratic residue problem by knowing the secret primes (x, y) which only kept by the remote server S. On the other hand, if C wants to reveal ID_A from the parameter u_A transmitted in M_2, C should collect u_A and R_{S2} and know $T_{r_{S2}}(\alpha)$ mod p or B's password verifier $H(rn_B, PW_B)$, where $u_A = ID_A \oplus T_{r_{S2}}(\alpha)$ mod p and $R_{S2} = T_{r_{S2}}(\alpha) - H(rn_B, PW_B)$ mod p. However, if SHA-256 (National Institute of Standards and Technology 2002) is used, due to the bit-length of $H(rn_B, PW_B)$ is 256 bits and the probability to guess a correct $H(rn_B, PW_B)$ is $\frac{1}{2^{256}}$. As a result, it is computationally infeasible for C to perform this attack in polynomial time. On the other hands, the transmitted messages $\{M_1, M_2, M_3, M_4, M_5, M_6\}$ are independent and different in every session since login parameters and random numbers are randomly selected and updated in every session. Finally, the proposed 3PAKE protocol achieves user anonymity. □

Proposition 2 *The proposed 3PAKE protocol can withstand password disclosure and stolen-verifier attacks and the attacker C cannot find any opportunity to acquire user's sensitive password including a privileged-insider of S.*

Proof In the registration phase of proposed 3PAKE protocol, the user U_i sends the registration request $\{ID_i, H(rn_i, PW_i)\}$ to S via a secure channel, where rn_i is a 128-bit random number and rn_i is unknown to S. In order to derive the password pw_i of the user U_i from $H(rn_i, PW_i)$, the privileged-insider C needs to guess correctly both rn_i and pw_i at the same time. We assume the probability of guessing pw_i composed of exact m characters and rn_i composed of exact k bits (in our proposed protocol, $k = 128$) is approximately $\frac{1}{2^{6m+k}} = \frac{1}{2^{6m+128}}$. This probability is very negligible and the privileged-insider C has no feasible way to derive pw_i of the user U_i in polynomial time. Therefore, the proposed 3PAKE protocol is secure against the password disclosure and stolen-verifier attacks. □

Proposition 3 *The proposed 3PAKE protocol is secured against impersonation attack while ensuring the system integrity of the entities that have participated in a three-party session over public channels; including A, B and S.*

Proof In impersonation attacks, two cases are taken into consideration. Case 1 is an attempt by the attacker to generate a malicious request to impersonate a legal user to login to the remote server. Case 2 is an attempt by the attacker to submit faked responses to impersonate a remote server to cheat two communication users.

Case 1 Assume that attacker C tries to login to S on behalf of A. Therefore, the attacker C needs to compute a valid authentication message $M_1 = (ID_A, ID_B, H(rn_A, PW_A), R_C)^2$ mod N, where $R_C = T_{r_C}(\alpha)$ mod p and $r_C \in [1, p+1]$ is chosen by C. However, in the proposed 3PAKE protocol, C knows neither rn_A nor $PW_A = T_{pw_A}(\alpha)$ mod p. As a result, C it is impossible for C to compute the value of $H(rn_A, PW_A)$ and forge a valid authentication message M_1 to cheat S and B.

Case 2 Suppose that attacker C also tries to impersonate the remote server S to cheat user B, C needs to generate a valid response $R_{S2} = T_{r_{S2}}(\alpha) - H(rn_B, PW_B)$ by knowing the values rn_B and PW_B, which are concealed in M_2. However, the probability to guess correct rn_B and PW_B is approximately $\frac{1}{2^{6m+128}}$, C cannot impersonate S to B. In addition, we assume that C tries to impersonate S to cheat user A, C must collect transmitted messages M_1, M_2, and M_3. Unfortunately, C cannot derive K_{SA}, K_{AB}, Z_{SA} and Z_{AB} from messages (M_1, M_2, M_3) due to the infeasibility of CDH problem and quadratic residue problem. □

Proposition 4 *The proposed 3PAKE protocol is secured against replay and modification attacks and the attacker C cannot replay and modify the authentication messages of the users and the remote server.*

Proof Suppose the attacker C intercepts the previous authentication message (i.e., M_1, M_3) and tries to impersonate the valid users A and B by immediately replaying the messages. To protect the proposed 3PAKE protocol from replay attacks, we use random numbers into the messages and the server would reject the request due to the invalid random numbers (i.e., R_A, R_B) will be detected in **Step A1** and **Step A3** of the authentication procedure. In addition, due to the protection of Chinese Remainder Theorem and one-way hash function, we ensure that authentication messages cannot be modified and the modified packets can be easily identified by checking the hash values. As such, the proposed 3PAKE protocol can resist replay and modification attacks. □

Proposition 5 *The proposed 3PAKE protocol is secured against offline password- guessing attacks and the attacker C cannot offline guess a legitimate user's password with the help of transmitted values from the remote server.*

Proof In the password change phase of our proposed 3PAKE protocol, the attacker C may attempt to impersonate a legitimate user and send the password change request to the remote server S. Then C can guess a legitimate user A's password with the help of transmitted values from S. Thus, in **Step C1** of the password change phase, the attacker C must generate a password change request $C_1 = (ID_A, H(rn_A, PW_A), R_C)^2 \bmod p$ and sends C_1 to S, where $R_C = T_{r_C}(\alpha) \bmod p$ and the random number $r_C \in [1, p+1]$ is chosen by C. However, C needs to guess correctly both rn_A and pw_A at the same time and the probability of guessing pw_A composed of exact m characters and rn_A composed of exact $k = 128$ bits is approximately $\frac{1}{2^{6m+k}} = \frac{1}{2^{6m+128}}$. This probability is very negligible and the server would reject the password change request due to the invalid secret value $H(rn_A, PW_A)$ will be detected in **Step C2** of the verification procedure. It is clear from the above discussion that offline password-guessing attack cannot work with the help of the server. □

Proposition 6 *The proposed 3PAKE protocol achieves mutual authentication and the remote server and two communication users can verify the validity of each other to establish mutual confidence before transmitting users' private data.*

Proof Mutual authentication means that the remote server can verify two communication users are legal and two users can ensure that the remote server is not a forged one. In the authentication and key exchange phase, **Step A4** shows that the server authenticates the user B and **Step A6** shows that the server authenticates the user A. Next, **Step A5** shows that the user A authenticates the user B and the server S. Finally, **Step A7** shows that the user B authenticates the user A and the server S and the proposed mutual authentication property makes the man-in-the-middle attacks necessarily unsuccessful. □

Proposition 7 *The proposed 3PAKE protocol achieves session key security and perfect forward secrecy.*

Proof The new session key $SK_{AB} = SK_{BA}$, which has been established by A and B with the help of S is only known by the two participants A and B themselves. In the proposed 3PAKE protocol, S knows $R_A = T_{r_A}(\alpha) \bmod p$ and $R_B = T_{r_B}(\alpha) \bmod p$, while S still cannot derive $K_{AB} = T_{r_A r_B}(\alpha) \bmod p = K_{BA}$ with the unknown of r_A or r_B which is the secret parameter of A and B by themselves. Perfect forward secrecy means that if some secret parameters are compromised by the attacker C, C still cannot derive previous session keys from them. In the proposed protocol, even if the parameters ID_A, ID_B, R_A and R_B are disclosed, the session key $SK_{AB} = H(2, ID_A, ID_B, R_A, R_B, K_{AB})$ still remains secure. Since the random numbers are different in every authentication session and it is equivalent to a CDH problem, which is assumed to be computationally hard. □

7 Performance and functionality comparisons with related 3PAKE protocols

In this section, we evaluate the efficiency and functionality of our proposed protocol and related 3PAKE protocols (Farash and Attari 2014; Lv et al. 2013; Zhao et al. 2013). For convenience to evaluate the computational costs and functional requirements, we define some notations as follows.

- t_h: The time of executing a one-way hash function.
- t_s: The time of executing a symmetric encryption/decryption operation.
- t_c: The time of executing a Chebyshev chaotic maps operation.

Table 1 Performance comparison between our proposed protocol and other typical 3PAKE protocols

	Lv et al. (2013)	Zhao et al. (2013)	Farash and Attari (2014)	Proposed protocol
P1	$2T_m + 3T_s + 1T_h$	$3T_c + 5T_h + 1T_s$	$2T_c + 4T_h$	$4T_c + 5T_h + 1T_m$
P2	$2T_m + 4T_s + 1T_h$	$3T_c + 5T_h + 1T_s$	$4T_c + 4T_h$	$4T_c + 5T_h + 1T_m$
P3	$2T_m + 3T_s + 2T_h$	$2T_c + 6T_h + 2T_s$	$2T_c + 4T_h$	$4T_c + 5T_h + 2T_q$
P4	13	12	17	13

P1 Computation cost of the user A
P2 Computation cost of the user B
P3 Computation cost of the server S
P4 Total messages transmitted between A, B and S

Table 2 Functionality comparison between our proposed protocol and other typical 3PAKE protocols

	Lv et al. (2013)	Zhao et al. (2013)	Farash and Attari (2014)	Proposed protocol
F1	×	√	×	√
F2	×	×	√	√
F3	√	√	×	√
F4	√	√	×	√
F5	√	√	×	√
F6	√	×	√	√
F7	√	×	√	√
F8	×	√	√	√
F9	×	×	√	√

F1 Provision of user anonymity
F2 Provision of password update
F3 Prevention of password disclosure attack
F4 Prevention of password-guessing attack
F5 Prevention of user impersonation attack
F6 Without using timestamp
F7 Without using smart card
F8 Without using secret key pre-shared between U_i and S
F9 Without using symmetric-key encryption/decryption
√ Yes; × No

– t_m: The time of executing a modular squaring operation.
– t_q: The time of executing a square root modulo N.

As shown in Table 1, we summarize the efficiency comparison between our proposed protocol and other previous 3PAKE protocols in terms of computation and communication cost during the authentication and key exchange phase. According to recent researches (Chen et al. 2008; Peris-Lopez et al. 2006), an implementation of a cheap modular squaring is using $f(x) = x^2 - an$ to substitute $f(x) = x^2 \bmod n$, where a is a carefully computed coefficient. Then the implementation of such a modular squaring can be reduced to a few hundred gate-equivalents and this is cheaper than traditional one-way hashing function such as MD5 and SHA-1, cost 16 K gates and 20 K gates, respectively. Although the total computational cost of Farash–Attari's protocol is fewest than other protocols, their protocol generates the most number of message flows. In fact, the authenticated Diffie–Hellman protocol needs more chaotic maps operations for providing anonymity of communication session and it is the reason why there are four additional chaotic maps operations $K_{SB} = T_{r_B r_{S2}}(\alpha) \bmod p = K_{BS}$, $K_{AB} = T_{r_A r_B}(\alpha) \bmod p = K_{BA}$ and $K_{SB} = T_{r_B r_{S2}}(\alpha) \bmod p = K_{BS}$ in our protocol than Farash–Attari's and Zhao et al.'s protocols. In addition, the proposed protocol requires two extra modular squaring computations and two computations of a square root modulo N than Zhao et al.'s protocol, it does not use symmetric encryption/decryption operations. Therefore, it is obvious that the execution time of the proposed protocol is still well suited for 3PAKE.

In Table 2, the security requirements and functional properties of four protocols are summarized. It is clear to note from Table 2 that our proposed protocol is superior when compared with other previous 3PAKE protocols. The proposed protocol and Zhao et al.'s protocol ensure user anonymity property, where Lv et al.'s and Farash–Attari's protocols does not support this property. Without protecting user anonymity, it may cause the leaking of the network users' sensitive data and the proposed protocol provides user anonymity during authentication phase to ensure communication privacy. In addition, Lv et al.'s and Zhao et al.'s protocols does not provide password update

procedure and require symmetric-key cryptosystem. Without providing password update mechanism for the user, he/she has to go to the remote server in person and ask for changing his/her password. Moreover, Zhao et al.'s protocol needs to use timestamp and smart card during authentication procedure and Lv et al.'s protocol needs to pre-shared a secret key between users and the server. Finally, considering the security and other extra important properties provided by our proposed protocol, we conclude that the proposed protocol outperforms than other related 3PAKE protocols.

8 Conclusions

In this paper, we pointed out that Farash–Attari's protocol is vulnerable to password disclosure, user impersonation and offline password-guessing attacks. To remedy these security weaknesses, we propose a chaotic maps and quadratic residues based three-party password-authenticated key exchange protocol. According to the comparisons, the proposed 3PAKE protocol is more secure and practical than other related protocols. Moreover, we extend our protocol to provide the user anonymity and two communication users' true identities and locations cannot be traced by any outsiders over public channels. Without doubt, two users and the remote server still can provide mutual authentication and compute a common session key with perfect forward secrecy.

Acknowledgements The authors would like to thank the anonymous reviewers and the Editor for their constructive and generous feedback on this paper. In addition, this research was partially supported and funded by the Ministry of Science and Technology, Taiwan, R.O.C., under contract no.: MOST 105-2221-E-165-005 and MOST 105- 2221-E-030-012.

Compliance with ethical standards

Conflict of interest Chun-Ta Li, Chin-Ling Chen , Cheng-Chi Lee, Chi-Yao Weng declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants performed by any of the authors.

References

Aboshosha A, ElDahshan KA, Elsayed EK, Elngar AA (2016) Secure authentication protocol based on machine-metrics and RC4-EA hashing. Int J Netw Secur 18(6):1080–1088

Bergamo P, Arco P, Santis A, Kocarev L (2005) Security of public-key cryptosystems based on Chebyshev polynomials. IEEE Trans Circuits Syst I 52(7):1382–1393

Brindha T, Shaji RS (2016) A secure transaction of cloud data using conditional source trust attributes encryption mechanism. Soft Comput. doi:10.1007/s00500-016-2405-6

Chen Y, Chou JS, Sun HM (2008) A novel mutual authentication scheme based on quadratic residues for RFID systems. Comput Netw 52(12):2373–2380

Chen Y, Chou JS, Sun HM (2013) A novel biometric-based remote user authentication scheme using quadratic residues. Int J Inf Electron Eng 3(4):419–422

Drissi A, Asimi A (2017) Behavioral and security study of the OHFGC hash function. Int J Netw Secur 19(3):335–339

Farash MS, Attari MA (2014) An efficient and provably secure three-party password-based authenticated key exchange protocol based on Chebyshev chaotic maps. Nonlinear Dyn 77(1–2):399–411

Guo C, Chang CC (2013) Chaotic maps-based password-authenticated key agreement using smart cards. Commun Nonlinear Sci Numer Simul 18(6):1433–1440

He D, Chen Y, Chen J (2012) Cryptanalysis and improvement of an extended chaotic maps-based key agreement protocol. Nonlinear Dyn 69(3):1149–1157

He D, Zhao W, Wu S (2013) Security analysis of a dynamic ID-based authentication scheme for multi-server environment using smart cards. Int J Netw Secur 15(5):350–356

He D, Zeadally S, Wu L (2015) Certificateless public auditing scheme for cloud-assisted wireless body area networks. IEEE Syst J. doi:10.1109/JSYST.2015.2428620

He D, Zeadally S (2015) Authentication protocol for ambient assisted living system. IEEE Commun Mag 35(1):71–77

He D, Zeadally S, Kumar N, Lee JH (2016) Anonymous authentication for wireless body area networks with provable security. IEEE Syst J. doi:10.1109/JSYST.2016.2544805

He D, Wang H, Wang L, Shen J, Yang X (2016) Efficient certificateless anonymous multi-receiver encryption scheme for mobile devices. Soft Comput. doi:10.1007/s00500-016-2231-x

Islam Sk H, Khan MK, Li X (2015) Security analysis and improvement of 'a more secure anonymous user authentication scheme for the integrated EPR information system. Plos ONE 10(8):e0131368

Khan MK (2009) Fingerprint biometric-based self-authentication and deniable authentication schemes for the electronic world. IETE Tech Rev 26(3):191–195

Khan MK, Kumari S (2013) An authentication scheme for secure access to healthcare services. J Med Syst 37:9954. doi:10.1007/s10916-013-9954-3

Lai H, Xiao J, Li L, Yang Y (2012) Applying semigroup property of enhanced Chebyshev polynomials to anonymous authentication protocol. Math Probl Eng, Article ID 454823. doi:10.1155/2012/454823

Lee CC, Li CT, Hsu CW (2013) A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps. Nonlinear Dyn 73(1–2):125–132

Li CT, Hwang MS (2010) An efficient biometrics-based remote user authentication scheme using smart cards. J Netw Comput Appl 33(1):1–5

Li CT, Lee CC (2012) A novel user authentication and privacy preserving scheme with smart cards for wireless communications. Math Comput Model 55(1–2):35–44

Li CT (2013) A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card. IET Inf Secur 7(1):3–10

Li CT, Lee CC, Weng CY, Fan CI (2013) An extended multi-server-based user authentication and key agreement scheme with user anonymity. KSII Trans Internet Inf Syst 7(1):119–131

Li CT, Weng CY, Lee CC (2013) An advanced temporal credential-based security scheme with mutual authentication and key agreement for wireless sensor networks. Sensors 13(8):9589–9603

Li CT, Lee CC, Weng CY (2013) 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

Li X, Niu J, Kumari S, Khan MK, Liao J, Liang W (2015) Design and analysis of a chaotic maps-based three-party authenticated key agreement protocol. Nonlinear Dyn 80(3):1209V1220

Li CT (2016) A secure chaotic maps-based privacy-protection scheme for multi-server environments. Secur Commun Netw. doi:10.1002/sec.1487

Li CT, Lee CC, Weng CY (2016a) A secure cloud-assisted wireless body area network in mobile emergency medical care system. J Med Syst 40(5):1–15. Article no. 117

Li CT, Lee CC, Weng CY (2016b) A secure dynamic identity and chaotic maps based user authentication and key agreement scheme for e-healthcare systems. J Med Syst 40(11):1–10. Article no. 233

Lin TH, Lee TF (2014) Secure verifier-based three-party authentication schemes without server public keys for data exchange in telecare medicine information systems. J Med Syst 38:30

Lv C, Ma M, Li H, Ma J, Zhang Y (2013) An novel three-party authenticated key exchange protocol using one-time key. J Netw Comput Appl 36(1):498–503

Mishra D, Kumari S, Khan MK, Mukhopadhyay S (2015) An anonymous biometric-based remote user-authenticated key agreement scheme for multimedia systems. Int J Commun Syst. doi:10.1002/dac.2946

National Institute of Standards and Technology (2002) US department of commerce, secure hash standard. US Federal Information Processing Standard Publication, Gaithersburg, pp 180–182

Peris-Lopez P, Hernandez-Castro JC, Estevez-Tapiador JM, Ribagorda A (2006) M2AP: a minimalist mutual-authentication protocol for low-cost RFID tags. In: Proceedings of international conference on ubiquitous intelligence and computing, vol 4195. LNCS, pp 912–923

Ramasamy R, Muniyandi AP (2012) An efficient password authentication scheme for smart card. Int J Netw Secur 14(3):180–186

Wen F (2014) A more secure anonymous user authentication scheme for the integrated EPR information system. J Med Syst 38:42

Wang X, Zhao J (2010) An improved key agreement protocol based on chaos. Commun Nonlinear Sci Numer Simul 15(12):4052–4057

Wu W, Hu S, Yang X, Liu JK, Au MH (2015) Towards secure and cost-effective fuzzy access control in mobile cloud computing. Soft Comput. doi:10.1007/s00500-015-1964-2

Xie Q, Zhao J, Yu X (2013) Chaotic maps-based three-party password-authenticated key agreement scheme. Nonlinear Dyn 74(4):1021–1027

Yang L, Ma JF, Jiang Q (2012) Mutual authentication scheme with smart cards and password under trusted computing. Int J Netw Secur 14(3):156–163

Yoon EJ, Jeon IS (2011) An efficient and secure DiffieVHellman key agreement protocol based on Chebyshev chaotic map. Commun Nonlinear Sci Numer Simul 16(6):2383–2389

Zhao F, Gong P, Li S, Li M, Li P (2013) Cryptanalysis and improvement of a three-party key agreement protocol using enhanced Chebyshev polynomials. Nonlinear Dyn 74(1–2):419–427