PLOS ONE

RESEARCH ARTICLE

# Anonymity preserving and round effective three-party authentication key exchange protocol based on chaotic maps

**Kyongsok Pak*, Songho Pak, Cholman Ho, Myongsuk Pak*, Choljin Hwang**

College of Information Science, Kim Il Sung University, Pyongyang, DPR of Korea

These authors contributed equally to this work.
* pks228@126.com (KP); info4@ryongnamsan.edu.kp (MP)





**Citation:** Pak K, Pak S, Ho C, Pak M, Hwang C (2019) Anonymity preserving and round effective three-party authentication key exchange protocol based on chaotic maps. PLoS ONE 14(3): e0213976. https://doi.org/10.1371/journal.pone.0213976

**Editor:** Muhammad Khurram Khan, King Saud University, SAUDI ARABIA

**Received:** June 15, 2018

**Accepted:** February 26, 2019

**Published:** March 20, 2019



**Data Availability Statement:** All relevant data are within the paper.

**Funding:** The authors received no specific funding for this work.

**Competing interests:** Lu proposed the 3PAKE protocol in the article "An extended chaotic maps-based three-party password-authenticated key agreement with user anonymity" (Plos One. 2016;11(4):e0153870) and attempted to provide user anonymity. We have found in his paper that there is a protocol design flaw and that it does not

## Abstract

Three-party authentication key exchange (3PAKE) is a protocol that allows two users to set up a common session key with the help of a trusted remote server, which is effective for secret communication between clients in a large-scale network environment. Since chaotic maps have superior characteristics, researchers have recently presented some of the studies that apply it to authentication key exchange and cryptography. Providing user anonymity in the authentication key exchange is one of the important security requirements to protect users' personal secrets. We analyse Lu et al.'s scheme which attempts to provide user anonymity and we prove that his scheme has errors in the key exchange phase and password change phase. We propose a round-effective three-party authentication key exchange (3PAKE) protocol that provides user anonymity and we analyse its security properties based on BAN logic and AVISPA tool.

## 1. Introduction

Along with the rapid development of the information technology and computer network, user authentication plays an important role in protecting resources, service and user's personal information in the computer network. The authentication key exchange protocol is one of the important mechanisms of network security aimed at setting a session key for secret communication between users via an open network. The authentication key exchange protocol is keys exchange for the secret communication based on authentication between the communicating parties in essence. The authentication key exchange protocol can be classified into Two-Party Authentication Key Exchange (2PAKE), Three-Party Authentication Key Exchange (3PAKE), and Multi-Party Authentication Key Exchange (MPAKE) depending on the number of participating in the key exchange. The key point of the 3PAKE protocol is that it does not need to remember various passwords for each user, and can establish secret communication between users with the help of a trusted remote server.

### 1.1 Cryptography for key exchange

Since the authentication key exchange protocol was proposed by Bellovin and Merritt [1] in 1992, there have been many studies on 2PAKE protocol [2,3], 3PAKE protocol and MPAKE

provide user anonymity. We proposed a computationally effective, round-effective 3PAKE protocol that provides user anonymity. We explicitly agree that you invite authors of the disputed work to sign in the first round of revisions. This does not alter our adherence to PLOS ONE policies on sharing data and materials.

[4–6] protocol based on the various cryptography algorithms for decades. The researchers used the Diffie-Hellman (DH) key exchange scheme [7–18], the Elliptic Curve Cryptosystem (ECC) based key exchange scheme [19–26], and the Chebyshev chaotic maps based key exchange scheme [27–38] for key exchange in 3PAKE protocol. The DH key exchange scheme based on modular exponentiation [39] requires a lot of computational cost. The ECC based scheme [40], in which the key length is small and the computational cost is low, has been used for key exchange. The ECC based scheme is more efficient in terms of key length and computational cost than the DH key exchange scheme using modular exponentiation [41].

In 2008, in order to enhance the property of the Chebyshev chaotic maps, Zhang [42] proved that the semi-group property holds for Chebyshev polynomials [43] defined over the interval $(-\infty, +\infty)$, and Chebyshev chaotic maps based key exchange schemes were widely used in the 3PAKE protocol. Chebyshev chaotic maps based scheme has advantages such as high safety, low computational cost, simple encryption, small storage capacity requirement, and low bandwidth [37, 44, 45]. Therefore, compared to DH and ECC based scheme, Chebyshev chaotic maps based scheme is more suitable for the wireless sensor network and the authentication system using smart card. In 2016, Kumari et al.[46] proposed mutual authentication and key agreement scheme for wireless sensor networks using Chebyshev chaotic maps, in which they described different chaotic maps that could be used in digital authentication and discussed a design methodology to present a robust authentication and key agreement for wireless sensor networks, and proposed a new authentication scheme for wireless sensor networks which provides user anonymity. However, his scheme is vulnerable to session-specific temporary information attack, sensor node impersonation attack, man-in-the-middle attack [47].

## 1.2 User authentication schemes in 3PAKE

In 3PAKE, the authentication server authenticates users and exchanges session key between users. In order for server to authenticate users in the 3PAKE protocol, researchers applied user password scheme [7–15, 19, 20, 27, 48], a combination of server public key and user password [17, 18, 23–26, 30–36], shared secret key scheme [21, 22, 28, 29, 49–51], and a combination of shared secret key and server public key [16, 38, 52–54].

The user password scheme without public key and shared secret key is easily revealed by password guessing attack as the information entropy of the password is low [8]. For example, in 2009 Huang [7] designed a 3PAKE protocol based on user password. However, Yoon et al. [10] proved that Huang's scheme is vulnerable to off-line password guessing attack and undetectable on-line password guessing attack. Wu et al. [17] proved that Huang's scheme is vulnerable to key-compromise impersonate attack, and proposed an updated 3PAKE protocol using user password and server public key. On the other hand, Chang et al. [8] proposed efficient 3PAKE protocol based on user password using modular exponentiation, and Wu et al. [19] pointed out that his scheme is vulnerable to password guessing attack and designed a 3PAKE protocol based on user password, however Wu et al.'s scheme is vulnerable to key-compromise impersonate attack [18]. Tso [12] also pointed out that Chang et al.'s scheme is vulnerable to password guessing attack, and Tso's scheme is vulnerable to the off-line password guessing attack and the impersonate attack [14]. Youn et al. [13] also designed efficient 3PAKE protocol based on user password, but his scheme is vulnerable to impersonate attack [15]. Farash et al. [27] proposed 3PAKE protocol based on the user password and the chaotic maps, but Li et al. [38] pointed out that his scheme is vulnerable to password disclosure attack, user impersonate attack, and off-line password guessing attack, and proposed a 3PAKE protocol based on chaotic maps with shared secret key.

The server public key scheme has to construct key management mechanism, so the protocol design is relatively complex and computational complexity is increased. But, using this scheme in the 3PAKE can provide user anonymity by encrypting the message exchanged between the user and the server. In 2014, Xie et al. [23] proposed a 3PAKE protocol based on ECC and the server public key, which provides user anonymity. However, his scheme is vulnerable to privileged insider attack, because there is a table stored user's password in the server side. Lou and Huang[24] also proposed a 3PAKE protocol based on ECC and the server public key, in which there is no encryption message using the server public key, but his scheme is vulnerable to off-line password guessing attack and key-compromise impersonate attack [26]. In 2013, Xie et al. [30] and Lee et al. [32] proposed a 3PAKE protocol based on the chaotic map and the server public key. However, Lee et al. [28] pointed out that Xie et al.'s scheme fails to provide user anonymity, is vulnerable to off-line password guessing attack, and has problems with password table management. Hu et al. [34] pointed out that Lee et al.'s scheme does not provide user anonymity and is vulnerable to MITM attack, and Farash et al. [33] pointed out that Lee et al.'s scheme is vulnerable to modification attack and impersonate attack.

In the shared secret key scheme, the server authenticates users by sharing his secret key with them. This scheme is safer than the password based scheme, because there is no user's private information in the server side. For example, it is resistant to privileged insider attack and stolen verifier attack. Tan [21] proposed a 3PAKE protocol based on ECC and the shared secret key, in which user keeps a private key combining with server secret key and user's identification. However his scheme is vulnerable to key-compromise impersonate attack [22]. Li [29] and Islam[50] proposed a 3PAKE protocol based on the chaotic map and the shared secret key, in which user encrypts the data for authentication with his private key derived by the server's private key, but user's identifier is exposed in the message, so their protocol does not provide user anonymity.

Meanwhile, in order to improve the effectiveness and safety of the authentication, there have been studies to implement the 3PAKE protocol by using devices such as smart cards [48–54]. In an authentication key exchange using a password that does not use a public key or shared secret key scheme, the user simply needs to remember the password. However, in an authentication key exchange that uses a public key or shared secret key scheme, the user must have a storage location for storing the server's shared secret key or his public key. The use of smart card not only allows users to carry their own authentication information, but also has the advantage of accessing service by using smart card reading devices anywhere. But in this scheme, there is a risk of losing the smart card. In 2012, Lai et al. [53] proposed the implementation of the 3PAKE protocol to use smart card based on chaotic maps. However, Zhao et al. [52] pointed out that Lai's scheme is vulnerable to privileged insider attack and off-line password guessing attack, and proposed an updated scheme to use smart card with server public key and shared secret key. Yang et al. [51] proposed a 3PAKE protocol that uses smart card with shared secret key, but Amin et al. [49] proved that Yang's scheme is vulnerable to off-line password attack, many logged-in user attack, privileged insider attack and has a security weakness in the password change phase, and proposed an updated scheme. In 2015, Xie et al. [48] proposed a 3PAKE protocol that uses smart card based on chaotic maps with user password, but his scheme had several weaknesses. In 2016, Lu et al. [31] pointed out that Xie's scheme is vulnerable to off-line password attack, user impersonate attack, does not provide user anonymity, and is deficient in session key security. He proposed an updated 3PAKE protocol that provides user anonymity using server public key and user password. However, Lu et al.'s scheme still has a series of weaknesses.

### 1.3 Our contribution

The user's identifier is a very important personal secret. If user anonymity is not provided, the attacker will know who is currently in the network conversation, and will be able to track the user's subscription history and current location. Chebyshev chaotic maps based authentication and key exchange scheme is suitable for the authentication system using smart card or the wireless sensor network, which requires low computational cost, simple encryption, small memory size, and low bandwidth. Based on such studies, we analyse the Lu et al.'s scheme [31] and point out its weakness, and propose a round-effective 3PAKE protocol based on chaotic maps using smart cards to provide user anonymity and protect against various attacks. In the proposed scheme, in order to provide the user anonymity the messages exchanged between the sender and the receiver is encrypted with the shared secret key based on the server's public key, and in order to authenticate the message, we use the user's private key derived by user's identifier and the server's secret key.

In Section 2, we describe the theory of chaotic maps, one-way function and Bio-hashing function, and In Section 3 we review Lu et al.'s scheme. Section 4 presents the proposed scheme, and Section 5 describes the security analysis of the proposed scheme. And Section 6 compares the proposed scheme with the previous schemes in terms of performance.

## 2. Preliminaries

This section describes Chebyshev chaotic maps and their computational problems, and Bio-hashing functions.

### 2.1 Chebyshev polynomials

Chebyshev polynomial $T_n(x)$ is defined as follows[43].

$T_n(x) = cos(n \cdot arcos(x))$, $x \in [-1,1]$, $n \in N$

Chebyshev polynomials satisfy the following recursive relationship[43].

$T_n(x) = 2x \cdot T_{n-1}(x) - T_{n-2}(x)$ $(n>2)$,

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

### 2.2 The property of Chebyshev polynomials

Chebyshev polynomials have the following two properties[43, 46].

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

Semi-group property: For $r,s \in N$ and any $x \in [-1,1]$, $T_r(T_s(x)) = T_{rs}(x) = T_s(T_r(x))$.

### 2.3 Enhanced Chebyshev polynomials

The semi-group property holds for Chebyshev polynomials on the interval $(-\infty,+\infty)$, which can enhance the property as follows [42, 43]:

$T_n(x) = 2x \cdot T_{n-1}(x) - T_{n-2}(x)$ $mod\ p (n \geq 2, x \in (-\infty,+\infty)$, $p$ is a large prime number),

$T_r(T_s(x)) \equiv T_{rs}(x) \equiv T_s(T_r(x))$ $mod\ p$ $(r,s \in N)$.

### 2.4 Computational problems based on Chebyshev polynomials

CDLP(Chaotic map-based Discrete Logarithm problem): For given two real numbers $x$ and $y$, it is infeasible to find the integer $r$ by any polynomial time bounded algorithm, where $y = T_r(x)$ $mod\ p$ [28, 42, 43].

CDHP(Chaotic map-based Diffie-Hellman problem): For given three elements $x$, $T_r(x)$ $mod\ p$ and $T_s(x)$ $mod\ p$, it is infeasible to compute the value $T_{rs}(x)$ $mod\ p$ by any polynomial time bounded algorithm [28, 42, 43].

### 2.5 Bio-hashing function

The biometric technique is very important for user authentication in the authentication system. Generally, imprint biometric characteristics (face, fingerprint, palm-print etc.) may not be exactly same at each time [49]. To solve this problem, Jina et al. [55] and Lumini et al. [56] proposed and updated Bio-hashing, which was used in many authentication schemes [45, 49, 57, 58]. Bio-hashing is used to map a user's biometric features to a user-specific random vectors [45, 57] and is useful for user authentication mechanisms that use small devices such as mobile devices, smart cards, and so on [57].

## 3. Review of Lu et al.'s scheme

This section shows that the scheme proposed by Lu et al. has series of deficiencies in the design. Lu et al. designed 3PAKE protocol based on chaotic maps providing user anonymity. However, his scheme has some errors in the session key exchange phase and the password change phase. Below is a brief description of the scheme proposed by Lu et al. and its deficiencies.

### 3.1 Lu et al.'s scheme

**Notations used in his paper.**

$S$: a remote server.
$A$ and $B$: two users.

$ID_A$ and $ID_B$: users' identities of $A$ and $B$.

$pwd_A$ and $pwd_B$: users' passwords of $A$ and $B$.

$k$ and $T_k(x)$ $mod\ p$: private and public keys of $S$.

$s$: a secret key of $S$.

$q$: shared secret key between $A$ and $S$.

$h_1()$: a one-way hash function.

$h()$: a chaotic maps-based one-way hash function.

$p$: a large prime number.

**System initialization.** The server selects random number $x \in Z_p$ and private key $k \in [1, p+1]$, computes public key $T_k(x)$ $mod\ p$ and publishes $\{p, x, T_k(x)\ mod\ p, h(\cdot)\}$.

**Registration.**

a. User $A$ submits $\{ID_A, g_A = h_1(pwd_A, r_A)\}$ to $S$, where $r_A$ is random number.

b. Upon receiving the registration request, $S$ computes $VPW_A = h_1(ID_A, k) \oplus g_A$. Next $S$ randomly chooses a secret key $q$ for $A$ and sends it to $A$ via the secure channel. Note that $q$ is kept securely by $A$ and is different for each user $A$. Finally, $S$ stores $k \oplus q$ and $VPW_A$ into its memory.

**Session key exchange.**

Step 1: Using the stored shared secret key $q$, user $A$ computes his own version of $C_A = E_{KA\text{-}S}(ID_A, ID_B, T_a(x), F_A)$ and sends them to $S$, where $K_{AS} = T_q(T_k(x))$, $F_A = h(ID_A, ID_B, T_a(x), g_A)$, $a \in [1, p+1]$ is a random number.

Step 2: Once receiving the message, $S$ first derives $q$ by computing $k \oplus q \oplus k$ and derives $\{ID_A, ID_B, T_a(x), F_A\}$ by decrypting $C_A$ with computed symmetric key $K_{AS} = T_k(T_q(x))$. The next steps are omitted here.

**Password update.**

Step 1: $A$ selects a new password $pwd_A{}^*$ and computes $R_A = E_{Tq(x)}(ID_A, h(pwd_A{}^*, r_A), h(pwd_A, r_A), Z_{AS})$, $Z_{AS} = h(ID_A, T_{S1}(x), K_{AS})$ and sends them to $S$.

Step 2: $S$ decrypts $R_A$ to retrieve $\{ID_A, h(pwd_A{}^*, r_A), h(pwd_A, r_A), Z_{AS}\}$ using the shared secret key $q$. The next steps are omitted here.

### 3.2 Defects in the design of Lu et al.'s scheme

**Session key exchange.** In the registration phase, Lu et al. pointed that $q$ is kept securely by $A$ and is different for each user $A$, and $S$ stores $k \oplus q$ into its memory. Therefore, $S$ must keep $k \oplus q$ for each user and can obtain it by user identifier. In the step2 of session key exchange phase, Lu et al. pointed that $S$ derives $q$ by computing $k \oplus q \oplus k$ and derives $\{ID_A, ID_B, T_a(x), F_A\}$ by decrypting $C_A$ with computed symmetric key $K_{AS} = T_k(T_q(x))$. In order for $S$ to retrieve $k \oplus q$ of $A$, the $A$'s identifier must be present, but $A$'s message $C_A$ is encrypted for providing user anonymity and has not yet been decrypted. Therefore, $S$ cannot know user $A$'s identifier, and cannot compute $q = (k \oplus q) \oplus k$. If $S$ stores a single $k \oplus q$ for all users, $S$ can decrypt the $A$'s message $C_A$ as in the protocol. But, in this case, other users can also decrypt $A$'s message because they also have $q$, so user anonymity cannot be provided in his scheme.

**Password update.** In the password change step, the same defects exist as seen in the session key exchange step. That is, $S$ does not obtain the key $K_{SA} = T_k(T_q(x))$ to decrypt the message $R_A$ or cannot update password.

## 4. Proposed scheme

This section describes an improved 3PAKE protocol using smart card that overcomes the limitations of the Lu et al.'s scheme. The proposed scheme consists of four steps: system initialization phase, registration phase, authentication and session key exchange phase, and password change phase. The notation presented in Table 1 is used to describe the proposed schemes in this paper.

### 4.1 System initialization phase

1. $S$ selects a large prime number $p$ and $x \in Z_p$ for Chebyshev polynomials $T_n(x)$.
2. $S$ selects secure one-way hash function $H(\cdot)$ and a symmetric encryption/decryption algorithm $E_K(\cdot)/D_K(\cdot)$.
3. $S$ selects $s \in [1, p+1]$ and keeps it as his secret key, and then computes public-key $K_S = T_s(x) \bmod p$.
4. $S$ publishes $\{p, x, K_S, H(\cdot), E_K(\cdot), D_K(\cdot)\}$ as system's parameters.

**Table 1. Notation used in proposed scheme.**

| Notation | Description |
|---|---|
| IDS | Identifier of trusted server S |
| SCA, SCB | smart card of user A and B |
| IDA, IDB | Identifier of user A and B |
| pwA, pwB | Password of A and B |
| bmA, bmB | Biometrics of A and B |
| s | Private key of S |
| p | A large prime number chosen by S |
| x | seed of Chebyshev polynomials. x ∈ Zp |
| Tn(x) | Chebyshev polynomials of degree n |
| KS | S's public-key (KS = Ts(x)) |
| H(·) | One-way hash function (0,1)* → (0, 1)n |
| h(·) | Bio-hashing function |
| EK(·) | Symmetric encrypt algorithm with secret key K |
| DK(·) | Symmetric decrypt algorithm with secret key K |
| \|\| | String concatenation operator |
| ⊕ | XOR operator |

https://doi.org/10.1371/journal.pone.0213976.t001

## 4.2 User registration phase

All users who want to exchange session keys using the proposed scheme must register on S.

Fig 1 shows an example of user A's registration process.

User $A$ sends his/her identifier $ID_A$ to $S$ via secure channel. $S$ checks whether user $A$ has already been registered, otherwise it computes $X_A = H(ID_A||s)$ and stores $\{p, x, X_A, K_S, H(\cdot), E_K(\cdot), D_K(\cdot)\}$ in $SC_A$ and delivers it to user $A$ via secure channel.

User $A$, which receives $SC_A$ from $S$, inputs password $pw_A$ and biometric $bm_A$ to access $SC_A$. The $SC_A$ that receives the user input computes $G_A = H(ID_A||pw_A||h(bm_A)) \oplus X_A$, $F_A = H(ID_A||pw_A||h(bm_A)||X_A)$ and stores $\{p, x, G_A, F_A, K_S, H(\cdot), E_K(\cdot), D_K(\cdot)\}$ in his memory.

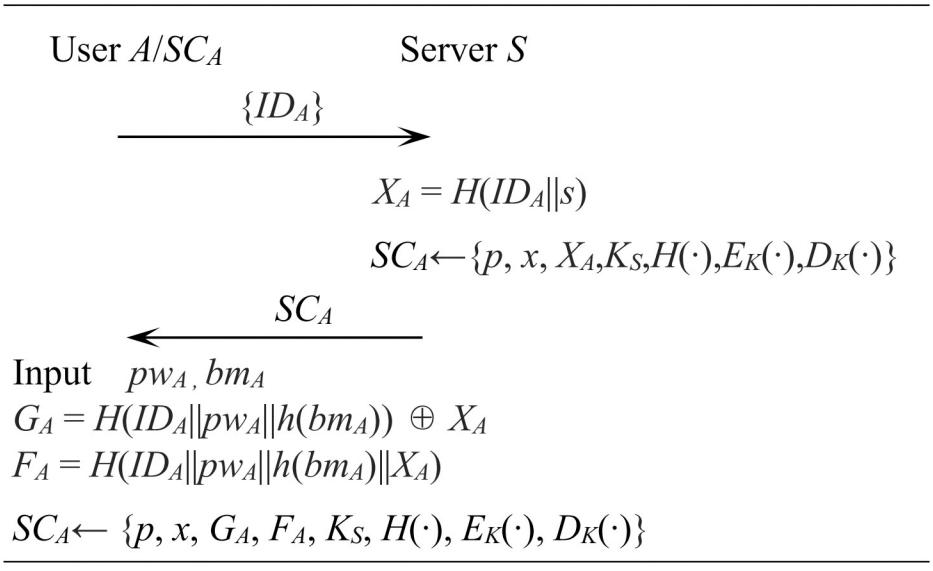


**Fig 1. User registration phase of the proposed scheme.**

https://doi.org/10.1371/journal.pone.0213976.g001

### 4.3 Authentication and session key exchange phase

Fig 2 show the authentication and session key exchange steps of the proposed scheme.

1. User $A$ connects his smart card $SC_A$ to the terminal and inputs his identifier $ID_A$, password and biometrics $bm_A$. $SC_A$ computes
   $X_A^* = G_A \oplus H(ID_A||pw_A||h(bm_A))$, $F_A^* = H(ID_A||pw_A||h(bm_A)||X_A^*)$.
   If $F_A \neq F_A^*$, $SC_A$ aborts the process. Otherwise $SC_A$ selects any $a \in [1, p+1]$ and computes
   $K_A = T_a(x)\ mod\ p$, $K_{AS} = T_a(K_S) = T_{as}(x)\ mod\ p$, $Z_{AS} = H(ID_A||ID_B||K_A\ ||X_A)$, $M_{AS} = E_{KA\text{-}S}(ID_A, ID_B, Z_{AS})$.
   $A$ sends $M_1 = \{M_{AS}, K_A\}$ to $B$.

2. After receiving $\{M_{AS}, K_A\}$ from $A$, $B$ connects his smart card $SC_B$ to the terminal and inputs his identifier $ID_B$, password and biometrics $pw_B$. $SC_B$ computes
   $X_B^* = G_B \oplus H(ID_B||pw_B||h(bm_B))$, $F_B^* = H(ID_B||pw_B||h(bm_B)||X_B^*)$.
   If $F_B \neq F_B^*$, $SC_B$ aborts the process. Otherwise $SC_B$ selects any $b \in [1, p+1]$ and computes
   $K_B = T_b(x)\ mod\ p$, $K_{BS} = T_b(K_S) = T_{bs}(x)\ mod\ p$, $K_{AB} = T_b(K_A) = T_{ba}(x)\ mod\ p$,
   $Z_{BA} = H(ID_B||K_{AB})$, $Z_{BS} = H(ID_B||K_B\ ||K_A||X_B)$, $M_{BS} = E_{KBS}(ID_B, Z_{BS}, Z_{BA})$.
   $B$ sends $M_2 = \{M_{AS}, K_A, M_{BS}, K_B\}$ to $S$.

3. After receiving $\{M_{AS}, K_A, M_{BS}, K_B\}$ from $B$, $S$ computes
   $K_{AS} = T_s(K_A) = T_{sa}(x)\ mod\ p$, $\{ID_A, ID_B^*, Z_{AS}^*\} = D_{KAS}(M_{AS})$, $X_A = H(ID_A||s)$, $Z_{AS} = H(ID_A||ID_B^*||K_A||X_A)$.
   $S$ checks whether $Z_{AS}$ and $Z_{AS}^*$ are same. If $Z_{AS} \neq Z_{AS}^*$, $S$ aborts the process. $S$ also computes
   $K_{BS} = T_s(K_B) = T_{sb}(x)\ mod\ p$, $\{ID_B, Z_{BS}^*, Z_{BA}\} = D_{KBS}(M_{BS})$, $X_B = H(ID_B||s)$, $Z_{BS} = H(ID_B||K_B||K_A||X_B)$.
   $S$ checks whether $Z_{BS}$ and $Z_{BS}^*$ are same. If $Z_{BS} \neq Z_{BS}^*$, $S$ aborts the process. $S$ also checks whether $ID_B^*$ of $A$'s message and $ID_B$ of $B$'s message are same. If not, $S$ aborts the process.
   After that, $S$ computes
   $Z_{SA} = H(ID_A||ID_B||K_A||K_B||X_A)$, $Z_{SB} = H(ID_B||ID_A||K_B||K_A||X_B)$, $M_{SA} = E_{KAS}(ID_B, K_B, Z_{SA}, Z_{BA})$, $M_{SB} = E_{KBS}(ID_A, K_A, Z_{SB})$.
   $S$ sends $M_3 = \{M_{SA}, M_{SB}\}$ to $A$.

4. After receiving $\{M_{SA}, M_{SB}\}$ from $S$, $A$ computes
   $\{ID_B, K_B, Z_{BA}^*, Z_{SA}^*\} = D_{KAS}(M_{SA})$, $Z_{SA} = H(ID_A||ID_B||K_A||K_B||X_A)$.
   If $Z_{SA} \neq Z_{SA}^*$, $A$ aborts the process. $A$ also computes
   $K_{AB} = T_a(K_B) = T_{ab}(x)\ mod\ p$, $Z_{BA} = H(ID_B||K_{AB})$.
   If $Z_{BA} \neq Z_{BA}^*$, $A$ aborts the process, otherwise $A$ sets $K_{AB}$ as a session key. $A$ also computes
   $Z_{AB} = H(ID_A||ID_B||K_{AB})$.
   $A$ sends $M_5 = \{M_{SB}, Z_{AB}\}$ to $B$.

5. After receiving $\{M_{SB}, Z_{AB}^*\}$ from $A$, $B$ computes
   $\{ID_A, K_A, Z_{SB}^*\} = D_{KBS}(M_{SB})$, $Z_{SB} = H(ID_B||ID_A||K_B||K_A||X_B)$.
   If $Z_{SB} \neq Z_{SB}^*$, $B$ aborts the process. $B$ also computes
   $Z_{AB} = H(ID_A||ID_B||K_{AB})$.
   If $Z_{AB} \neq Z_{AB}^*$, $B$ aborts the process. Otherwise $B$ sets $K_{AB}$ as a session key.

### 4.4 Password change phase

User $A$ connects his smart card $SC_A$ to the terminal and inputs his identifier $A$, password and biometrics $bm_A$. $SC_A$ computes $X_A = G_A \oplus H(ID_A||pw_A||h(bm_A))$ and $F_A^* = H(ID_A||pw_A||h$

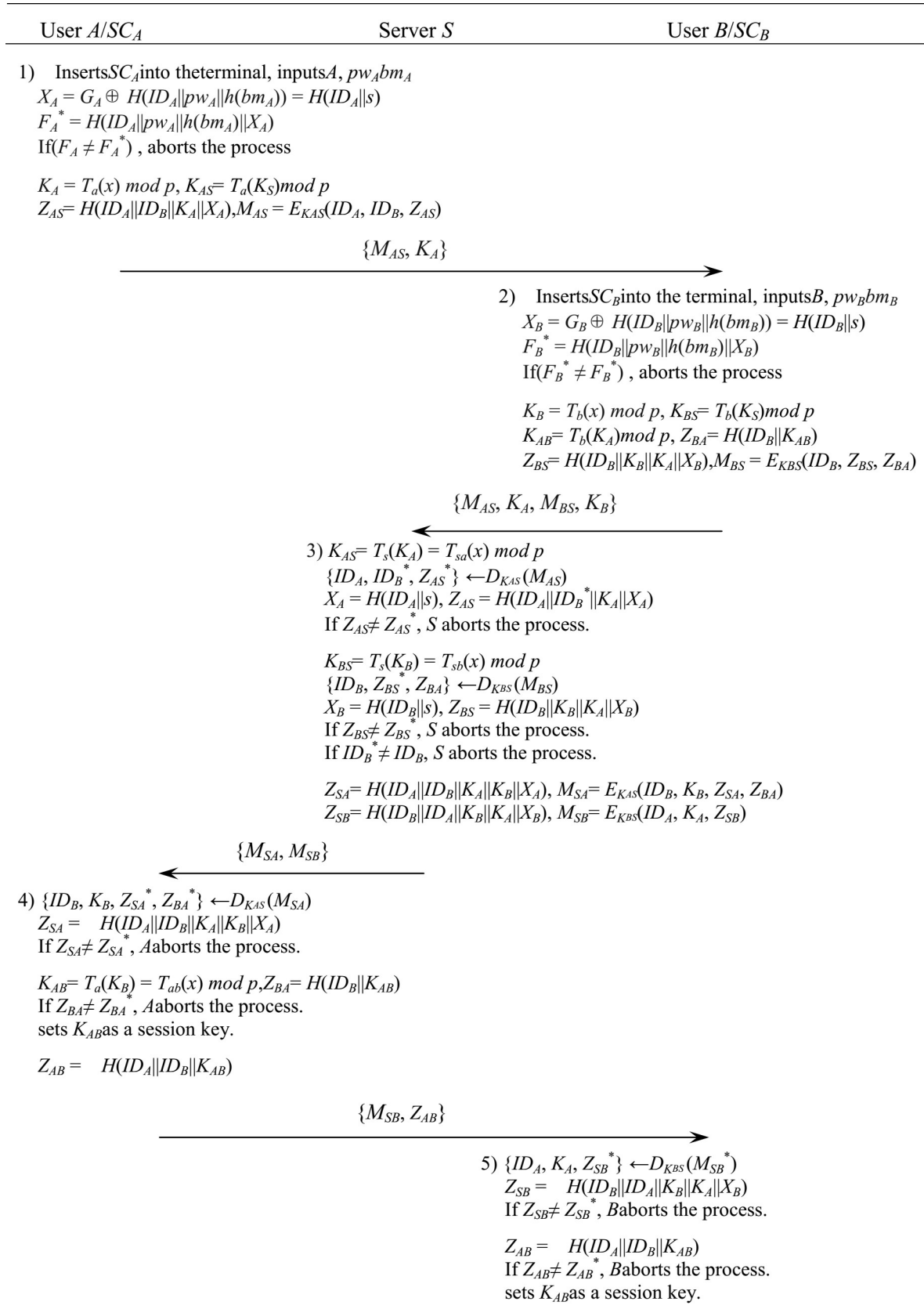


**Fig 2. Authentication and session key exchange phase of the proposed scheme.**

https://doi.org/10.1371/journal.pone.0213976.g002

$(bm_A)||X_A)$, and checks whether $F_A$ and $F_A^*$ are same. If $F_A \neq F_A^*$, $SC_A$ aborts the process. Otherwise $SC_A$ requests the user to input a new password $newpw_A$. $SC_A$ computes $G_A^{new} = H(ID_A||newpw_A||h(bm_A)) \oplus X_A$ and $F_A^{new} = H(ID_A||newpw_A||h(bm_A)||X_A)$, and replaces $<G_A, F_A>$ of his memory with $<G_A^{new}, F_A^{new}>$.

## 5. Security analysis of the proposed scheme

In this section, we analyse the security properties of the proposed scheme. First, we prove the correctness of the session key between users by using BAN logic [59]. Next, we simulate the proposed scheme for the formal security analysis by using AVISPA(Automated validation of internet security protocol and application) tool [60]. Last, we demonstrate the proposed scheme can resist various kinds of attacks.

### 5.1 Authentication proof based on BAN logic

**Notations and Rules.** We define $P$ and $Q$ as the specific participators, $S$ is the trusted server, and $X$ is the formula (statement). Some notations and rules of BAN logic are as follows [59].

$P \mid\equiv X$: $P$ believes $X$.

$P \lhd X$: $P$ sees $X$.

$P \mid\sim X$: $P$ once said $X$.

$P \mid\Rightarrow X$: $P$ has jurisdiction over $X$.

$\#(X)$: $X$ is fresh.

$P \overset{K}{\leftrightarrow} Q$: $K$ is a shared secret key between $P$ and $Q$.

$\{X\}_K$: Formula $X$ are encrypted under the key $K$.

$<X>_Y$: $X$ combined with the formula $Y$.

$R_1 : \frac{P|\equiv Q \overset{K}{\leftrightarrow} P, P \lhd \{X\}_K}{P|\equiv Q|\sim X}$ (Message-meaning rule): if $P$ believes that the key $K$ is shared with $Q$ and receives a message containing $X$ encrypted under $K$, then $P$ believes that $Q$ once said $X$.

$R_2 : \frac{P|\equiv \#(X), P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}$ (Nonce-verification rule): if $P$ believes $X$ is fresh and $Q$ once said $X$, $P$ believes $Q$ believes $X$.

$R_3 : \frac{P|\equiv Q|\Rightarrow X, P|\equiv Q|\equiv X}{P|\equiv Q|\equiv X}$ (Jurisdiction rule): if $P$ believes that $Q$ had jurisdiction right to $X$ and believes $Q$ believes $X$, $P$ believes $X$.

$R_4 : \frac{P|\equiv \#\#(X)}{P|\equiv \#(X,Y)}$ (Freshness rule): If $X$ is a part of message($X$, $Y$) and $X$ is fresh, message ($X$, $Y$) is also fresh.

$R_5 : \frac{P|\equiv Q|\equiv (X,Y)}{P|\equiv Q|\equiv X}$ (Belief rule 1): If $P$ believes $Q$ believes the message set ($X$, $Y$), $P$ also believes $Q$ believes the message $X$.

$R_6 : \frac{P|\equiv X, P|\equiv Y}{P|\equiv (X,Y)}$ (Belief rule 2): If $P$ believes the message $X$ and $Y$, $P$ also believes the message set ($X$, $Y$).

$R_7 : \frac{P|\equiv Q \overset{K}{\leftrightarrow} P, P \lhd \{X\}_K}{P \lhd X}$ (See rule): if $P$ believes that the key $K$ is shared with $Q$ and receives a message containing $X$ encrypted under $K$, then $P$ sees $X$.

**Goals.** The session key exchange protocol should achieve the following goals:

$$Goal_1 : A| \equiv A \overset{K_{AB}}{\longleftrightarrow} B$$

$$Goal_2 : B| \equiv A \overset{K_{AB}}{\longleftrightarrow} B$$

$$Goal_3 : A| \equiv B| \equiv A \overset{K_{AB}}{\longleftrightarrow} B$$

$$Goal_4 : B| \equiv A| \equiv A \overset{K_{AB}}{\longleftrightarrow} B$$

**Idealize.** We idealize the communication messages of the proposed scheme as follows:

$$M_1 : A \rightarrow S : \{ID_A, ID_B, Z_{AS}\}_{H(ID_A \| s)}, T_a(x)$$

$$M_2 : B \rightarrow S : \{ID_B, Z_{BS}, Z_{BA}\}_{H(ID_B \| s)}, T_b(x)$$

$$M_3 : S \rightarrow A : \{ID_B, T_b(x), Z_{SA}, Z_{BA}\}_{H(ID_A \| s)}$$

$$M_4 : S \rightarrow A : \{(ID_B, A \overset{K_{AB}}{\longleftrightarrow} B)_{A \overset{K_{AB}}{\longleftrightarrow} B}\}_{H(ID_A \| s)}$$

$$M_5 : S \rightarrow B : \{ID_A, T_a(x), Z_{SB}\}_{H(ID_B \| s)}$$

$$M_6 : A \rightarrow B : (ID_A, ID_B, A \overset{K_{AB}}{\longleftrightarrow} B)_{A \overset{K_{AB}}{\longleftrightarrow} B}$$

**Assumptions.** The initial assumptions of the proposed scheme are as follows:

$$A_1 : A| \equiv a$$

$$A_2 : A| \equiv \#(a)$$

$$A_3 : B| \equiv b$$

$$A_4 : B| \equiv \#(b)$$

$$A_5 : A| \equiv A \overset{H(ID_A \| s)}{\longleftrightarrow} S$$

$$A_6 : B| \equiv B \overset{H(ID_B \| s)}{\longleftrightarrow} S$$

$$A_7 : A| \equiv S| \Rightarrow T_b(x)$$

$$A_8 : B| \equiv S| \Rightarrow T_a(x)$$

**Analysis.** According to $M_3$ and $A_5$, we apply the message meaning rule ($R_1$) and the See rule ($R_7$), we can obtain:

$$S_1 : \frac{A| \equiv A \overset{H(ID_A||s)}{\longleftrightarrow} S, A \lhd \{ID_B, T_b(x), Z_{SA}, Z_{BA}\}_{H(ID_A||s)}}{A| \equiv S| \sim \{ID_B, T_b(x), Z_{SA}, Z_{BA}\}, A \lhd Z_{BA}}$$

According to $Z_{SA} = H(ID_A||ID_B||T_a(x)||T_b(x)||X_A)$, $A_2$ and $M_3$, we apply the Freshness rule ($R_4$), we can obtain:

$$S_2 : \frac{A| \equiv \#(a)}{A| \equiv \#(Z_{SA})}, \frac{A| \equiv \#(Z_{SA})}{A| \equiv \#(ID_B, T_b(x), Z_{SA}, Z_{BA})}$$

According to $S_1$ and $S_2$, we apply the Nonce-verification rule ($R_2$) and Belief rule 1($R_5$), we can obtain:

$$\frac{A| \equiv \#(ID_B, T_b(x), Z_{SA}), A| \equiv S| \sim \{ID_B, T_b(x), Z_{SA}, Z_{BA}\}}{A| \equiv S| \equiv (ID_B, T_b(x), Z_{SA}, Z_{BA})}$$

$$S_3 : \frac{A| \equiv S| \equiv (ID_B, T_b(x), Z_{SA}, Z_{BA})}{A| \equiv S| \equiv T_b(x)}$$

According to $S_3$ and $A_7$, we apply the Jurisdiction rule ($R_3$), we can obtain:

$$S_4 : \frac{A| \equiv S| \Rightarrow T_b(x), A| \equiv S| \equiv T_b(x)}{A| \equiv T_b(x)}$$

According to $S_4$, $A_1$ and $K_{AB} = T_a(T_b(\mathrm{x})) = (a, T_b(x))$, we apply the Belief rule 2($R_6$), we can obtain:

$$S_5 : \frac{A| \equiv a, A| \equiv T_b(x)}{A| \equiv A \overset{K_{AB}}{\longleftrightarrow} B} : (Goal_1)$$

According to $M_5$ and $A_6$, we apply the message meaning rule ($R_1$), we can obtain:

$$S_6 : \frac{B| \equiv B \overset{H(ID_B||s)}{\longleftrightarrow} S, B \lhd \{ID_A, T_a(x), Z_{SB}\}_{H(ID_B||s)}}{B| \equiv S| \sim \{ID_A, T_a(x), Z_{SB}\}}$$

According to $Z_{SB} = H(ID_B||ID_A||T_b(x)||T_a(x)||X_B)$, $A_4$ and $M_5$, we apply the Freshness rule ($R_4$), we can obtain:

$$S_7 : \frac{B| \equiv \#(b)}{B| \equiv \#(Z_{SB})}, \frac{B| \equiv \#(Z_{SB})}{B| \equiv \#(ID_A, T_a(x), Z_{SB})}$$

According to $S_6$ and $S_7$, we apply the Nonce-verification rule ($R_2$) and the Belief rule 1($R_5$), we can obtain:

$$\frac{B| \equiv \#(ID_A, T_a(x), Z_{SB}), B| \equiv S| \sim \{ID_A, T_a(x), Z_{SB}\}}{B| \equiv S| \equiv (ID_A, T_a(x), Z_{SB})}$$

$$S_8 : \frac{B| \equiv S| \equiv (ID_A, T_a(x), Z_{SB})}{B| \equiv S| \equiv T_a(x)}$$

According to $S_8$ and $A_8$, we apply the Jurisdiction rule ($R_3$), we can obtain:

$$S_9 : \frac{B| \equiv S| \Rightarrow T_a(x), B| \equiv S| \equiv T_a(x)}{B| \equiv T_a(x)}$$

According to $S_9$, $A_3$ and $K_{AB} = T_b(T_a(\mathrm{x})) = (b, T_a(x))$, we apply the Belief rule 2 ($R_6$), we can obtain:

$$S_{10} : \frac{B| \equiv b, B| \equiv T_a(x)}{B| \equiv A \overset{K_{AB}}{\longleftrightarrow} B} : (Goal_2)$$

According to $M_4$, $S_1$ and $S_5$, we apply the message meaning rule ($R_1$), we can obtain:

$$S_{11} : \frac{A| \equiv A \overset{K_{AB}}{\longleftrightarrow} B, A \triangleleft (ID_B, A \overset{K_{AB}}{\longleftrightarrow} B)_{A \overset{K_{AB}}{\longleftrightarrow} B}}{A| \equiv B| \sim \{ID_B, A \overset{K_{AB}}{\longleftrightarrow} B\}}$$

According to $A_2$ and $K_{AB} = T_b(T_a(\mathrm{x})) = (a, T_b(x))$, we apply the Freshness rule ($R_4$), we can obtain:

$$S_{12} : \frac{A| \equiv \#(a)}{A| \equiv \#(A \overset{K_{AB}}{\longleftrightarrow} B)}$$

According to $S_{11}$ and $S_{12}$, we apply the Nonce-verification rule ($R_2$), we can obtain:

$$S_{13} : \frac{A| \equiv \#(A \overset{K_{AB}}{\longleftrightarrow} B), A| \equiv B| \sim \{A \overset{K_{AB}}{\longleftrightarrow} B\}}{A| \equiv B| \equiv (A \overset{K_{AB}}{\longleftrightarrow} B)} : (Goal_3)$$

According to $M_6$ and $S_{10}$, we apply the message meaning rule ($R_1$), we can obtain:

$$S_{14} : \frac{B| \equiv A \overset{K_{AB}}{\longleftrightarrow} B, B \triangleleft (ID_A, A \overset{K_{AB}}{\longleftrightarrow} B)_{A \overset{K_{AB}}{\longleftrightarrow} B}}{B| \equiv A| \sim \{ID_A, A \overset{K_{AB}}{\longleftrightarrow} B\}}$$

According to $A_4$ and $K_{AB} = T_a(T_b(\mathrm{x})) = (b, T_a(x))$, we apply the Freshness rule($R_4$), we can obtain:

$$S_{15} : \frac{B| \equiv \#(b)}{B| \equiv \#(A \overset{K_{AB}}{\longleftrightarrow} B)}$$

According to $S_{14}$ and $S_{15}$, we apply the Nonce-verification rule ($R_2$), we can obtain:

$$S_{16} : \frac{B| \equiv \#(A \overset{K_{AB}}{\longleftrightarrow} B), B| \equiv A| \sim \{A \overset{K_{AB}}{\longleftrightarrow} B\}}{B| \equiv A| \equiv (A \overset{K_{AB}}{\longleftrightarrow} B)} : (Goal_4)$$

## 5.2 Validation test based on AVISPA

In this section, we simulate the proposed scheme for the formal security analysis using AVISPA, which is widely used to verify the security properties of designed protocol such as resistance against replay attack and man-in-the-middle attack. This tool implements four back-ends: On-the-Fly-Model-Check(OFMC), Constraint Logic based Attack Searcher (CL-AtSe), SAT-based Model-Checker(SATMC) and Three Automata based on Automatic Approximations for the Analysis of Security Protocols(TA4SP), which are given in details in [60]. In order to verify the security properties of the protocol using AVISPA, it needs to be specified in HLPSL(High Level Protocol Specification Language), which is a role-based languages: basic roles for representing each participant role, and composition roles for

representing scenarios of basic roles. Each role is independent from the other, communicating with the other roles by channels [60]. The output format is generated by using one of the four back-ends.

**Specifying the proposed protocol.** In our HLPSL implementation, we define three basic roles for users $A$, $B$, and server $S$. Figs 3, 4 and 5 shows the specifications in HLPSL for the role of users $A$, $B$, and server $S$.

In Fig 6, we shows the HLPSL implementation for the role of the session, environment and goal.

In our implementation, we verified the following five secrecy goals and six authentication properties.

- secrecy_of sec_ida: It represents that user $A$'s identifier $ID_A$ is kept secret to the user $A$, $B$ and server $S$ only.
- secrecy_of sec_idb: It represents that user $B$'s identifier $ID_B$ is kept secret to the user $A$, $B$ and server $S$ only.
- secrecy_of sec_xa: It represents that user $A$'s secret key $X_A$ is kept secret to the user $A$ and server $S$ only.
- secrecy_of sec_xb: It represents that user $B$'s secret key $X_B$ is kept secret to the user $B$ and server $S$ only.
- secrecy_of sec_kab: It represents that session key $K_{AB}$ is kept secret to the user $A$ and $B$ only.
- authentication_on auth_a_s_kas: When user $A$ receives the messages from server $S$ and decrypts the message with $K_{AS}$, $A$ authenticates $S$ based on $K_{AS}$.
- authentication_on auth_a_b_zba: When user $A$ receives $Z_{BA}$ from the messages from $B$, $A$ authenticates $B$ based on $Z_{BA}$.
- authentication_on auth_b_s_kbs: When user $B$ receives the messages from server $S$ and decrypts the message with $K_{BS}$, $B$ authenticates $S$ based on $K_{BS}$.
- authentication_on auth_b_a_zab: When user $B$ receives $Z_{AB}$ from the messages from $A$, $B$ authenticates $A$ based on $Z_{AB}$.
- authentication_on auth_s_a_xa: When server $S$ receives $X_A$ from the messages from $A$, $S$ authenticates $A$ based on $X_A$.
- authentication_on auth_s_b_xb: When server $S$ receives $X_B$ from the messages from $B$, $S$ authenticates $B$ based on $X_B$.

**Analysis of the results.** We have simulated the proposed scheme using FMC and CL-AtSe back-ends of AVISPA. The simulation results for the security verification is shown in Figs 7 and 8.

The results ensure that the proposed scheme is secure under the test of AVISPA using OFMC and CL-AtSe back-ends, and guarantees user anonymity, and it is also secure against the passive attacks and the active attacks, such as the replay attack and man-in-the-middle attack.

## 5.3 Informal security analysis

In this part, we demonstrate the proposed scheme can resist various kinds of attacks.

```
role alice(A,B,S :agent, XA, KS: symmetric_key, H: hash_func,
           SNDB, RCVS : channel(dy))
played_by A
def=
local   State: nat,
        IDA, IDB, Ta, Tb, Tx: text,
        KA, KAS, KAB, ZAB, ZBA: text,
        MSB: {text.hash_func.hash_func}_hash_func
const   auth_s_a_xa, auth_a_s_kas, auth_b_a_zab, auth_a_b_zba,
        sec_ida, sec_xa, sec_kab : protocol_id
init    State := 0
transition
1. State = 0 /\ RCVS(start)    =|>
   State':= 1/\ secret({XA}, sec_xa, {A,S})
        /\ secret({IDA}, sec_ida, {A,B, S})
        /\ witness(A, S, auth_s_a_xa, XA)
        /\ Ta':= new()
        /\ KA':= H(Tx.Ta')
        /\ KAS':= H(KS.Ta')
%   Send the login request message to server
        /\ SNDB({IDA.IDB.H(IDA.IDB.KA'.XA)}_KAS'.KA')
%   Receive the authentication reply message from server
2. State = 1 /\ RCVS({IDB.H(Tx.Tb').H(IDA.IDB.H(Tx.Ta').
                          H(Tx.Tb').XA).ZBA'}_KAS'.MSB' )    =|>
   State':= 2/\ KAB':=    H(Tx.Tb'.Ta')
        /\ secret({KAB'}, sec_kab, {A, B})
        /\ ZAB':= H(IDA.IDB.KAB')
        /\ witness(A, B, auth_b_a_zab, ZAB')
        /\ request(A, B, auth_a_b_zba, ZBA')
        /\ request(A, S, auth_a_s_kas, KAS')
%   Send the authentication message to bob
        /\ SNDB(MSB'.ZAB')
end role
```

**Fig 3. Role specification in HLPSL for the user *A*.**

https://doi.org/10.1371/journal.pone.0213976.g003

```
role bob(A,B,S :agent, XB, KS: symmetric_key, H: hash_func,
         SNDS, RCVA : channel(dy))
played_by B
def=
local State: nat,
      IDB, IDA, Tb, Ta, Tx: text,
      KB, KBS, KAB, ZAB, ZBA: text,
      MAS:{text.text.hash_func}_hash_func
const auth_s_b_xb, auth_b_s_kbs, auth_a_b_zba, auth_b_a_zab,
      sec_idb, sec_xb, sec_kab: protocol_id
init  State := 0
transition
%  Receive the authentication message from alice
1. State = 0 /\ RCVA(MAS'.H(Tx.Ta')) =|>
   State':= 1/\ secret({XB}, sec_xb, {B,S})
      /\ secret({IDB}, sec_idb, {A,B,S})
      /\ witness(B, S, auth_s_b_xb, XB)
      /\ Tb':= new()
      /\ KB':= H(Tx.Tb')
      /\ KBS':= H(KS.Tb')
      /\ KAB':= H(Tx.Ta'.Tb')
      /\ secret({KAB'}, sec_kab, {A, B})
      /\ ZBA':= H(IDB.KAB')
      /\ witness(B, A, auth_a_b_zba, ZBA')
%  Send the login request message to server
      /\ SNDS(MAS'.H(Tx.Ta').{IDB.H(IDB.KB'.H(Tx.Ta').XB).
                               ZBA'}_KBS'.KB')
%  Receive the authentication message from alice
2. State = 1 /\ RCVA({IDA.H(Tx.Ta').H(IDB.IDA.H(Tx.Ta').XB)}_
                               KBS'.ZAB')    =|>
   State':= 2/\ request(B, A, auth_b_a_zab, ZAB')
      /\ request(B, S, auth_b_s_kbs, KBS')
end role
```

**Fig 4. Role specification in HLPSL for the user *B*.**

https://doi.org/10.1371/journal.pone.0213976.g004

```
role server(A,B,S :agent,    XA,XB,KS: symmetric_key, H: hash_func,
                SNDA,RCVB : channel(dy))
played_by S
def=
local State: nat,
        IDA, IDB, Ta, Tb, Tx: text,
        KA, KB, KAS, KBS, KAB, ZBA: text,
const auth_s_a_xa, auth_s_b_xb, auth_a_s_kas, auth_b_s_kbs : protocol_id
init     State := 0
transition
%    Receive the login request message from bob
1. State = 0/\ RCVB({IDA.IDB.H(IDA.IDB.H(Tx.Ta').XA)}_KAS'.H(Tx.Ta').
                        {IDB.H(IDB.H(Tx.Tb'),H(Tx.Ta').XB).ZBA'}_
                                        KBS'.H(Tx.Tb')) =|>
%    Send the authentication reply message to alice
   State':= 1/\ request(S, A, auth_s_a_xa, XA)
        /\ request(S, B, auth_s_b_xb, XB)
        /\ SNDA({IDB.H(Tx.Tb').H(IDA.IDB.H(Tx.Ta').
                                        H(Tx.Tb').XA).ZBA'}_KAS'.
                {IDA.H(Tx.Ta').H(IDB.IDA.H(Tx.Ta').XB)}_KBS' )
        /\ witness(S, A, auth_a_s_kas, KAS')
        /\ witness(S, B, auth_b_s_kbs, KBS')
end role
```

**Fig 5. Role specification in HLPSL for the server *S*.**

https://doi.org/10.1371/journal.pone.0213976.g005

**User anonymity.** The proposed scheme provides user anonymity for key exchange. All message ($M_{AS}$, $M_{BS}$, $M_{SA}$ and $M_{SB}$) associated with the user's identifier is encrypted with the shared secret key $K_{XS}$ between the server $S$ and the user $X$. The shared secret key $K_{AS}$ is calculated from the random number $a$ of the user $A$ and the secret key $s$ of the server $S$ as follows: $K_{AS} = T_a(T_s(x)) = T_s(T_a(x))$.

Even if $T_a(x)$ and $T_s(x)$ is exposed, it is impossible to calculate $K_{AS}$ or $a$, $s$ according to *CDLP* and *CDHP* assumptions. Therefore, a third party cannot know the user's identifier except user and server.

**Off-line password guessing attack.** The proposed scheme resists the password guessing attack. The proposed scheme does not use passwords during the authentication process but only uses passwords when accessing the smart card. The information registered on the user $A$'s smart card is $\{G_A, F_A, p, x, K_S, R_S, H(\cdot), E_K(\cdot), D_K(\cdot)\}$, and the information that can be used for guessing password is $G_A = H(ID_A||pw_A||h(bm_A))\oplus X_A$ and $F_A = H(ID_A||pw_A||h(bm_A)\ ||X_A)$. Suppose that an attacker steals user $A$'s smart card $SC_A$ and knows his identifier $ID_A$. Then the

```
role session(A,B,S:agent, XA,XB,KS:symmetric_key, H:hash_func)
def=
local AB, BS, SA: channel(dy)
composition
        alice(A, B, S,    XA, KS,    H,    AB, SA)
        /\ bob(A, B, S, XA, KS,    H,    BS, AB)
        /\ server(A, B, S, XA, XB, KS,    H,    SA, BS)
end role

role environment()
def=
const a, b, s: agent, xa,xb,xi,ks: symmetric_key,
        h: hash_func,
        p: text,
        auth_s_a_xa, auth_s_b_xb,
        auth_a_s_kas, auth_b_s_kbs,
        auth_a_b_zba, auth_b_a_zab,
        sec_ida, sec_idb, sec_xa, sec_xb,
        sec_kab, sec_ss : protocol_id
intruder_knowledge = {a, b, s, xi, h, ks, p}
composition
        session(a, b, s, xa,xb,ks, h)
        /\ session(a, i, s, xa,xi,ks, h)
        /\ session(i, b, s, xi,xb,ks, h)
        /\ session(a, b, i, xa,xb,ks, h)
end role

goal
secrecy_of sec_ida
secrecy_of sec_idb
secrecy_of sec_xa
secrecy_of sec_xb
secrecy_of sec_kab
authentication_on auth_a_s_kas
authentication_on auth_a_b_zba
authentication_on auth_b_s_kbs
authentication_on auth_b_a_zab
authentication_on auth_s_a_xa
authentication_on auth_s_b_xb
end goal
environment()
```

**Fig 6. Role specification in HLPSL for the session, environment and goal.**

https://doi.org/10.1371/journal.pone.0213976.g006

attacker must compute $PW_A^* = H(ID_A||pw_A^*||h(bm_A))$, $X_A^* = G_A \oplus PW_A^*$ and $F_A^* = H(ID_A||pw_A^*||h(bm_A)||X_A^*)$ by using $ID_A$ and any password $pw_A^*$ to compare $F_A^*$ and $F_A$ stored in $SC_A$. However, $PW_A^*$ cannot be calculated without knowing $h(bm_A)$ which is related $A$'s biometrics. Therefore, the attacker cannot guess the user's password.

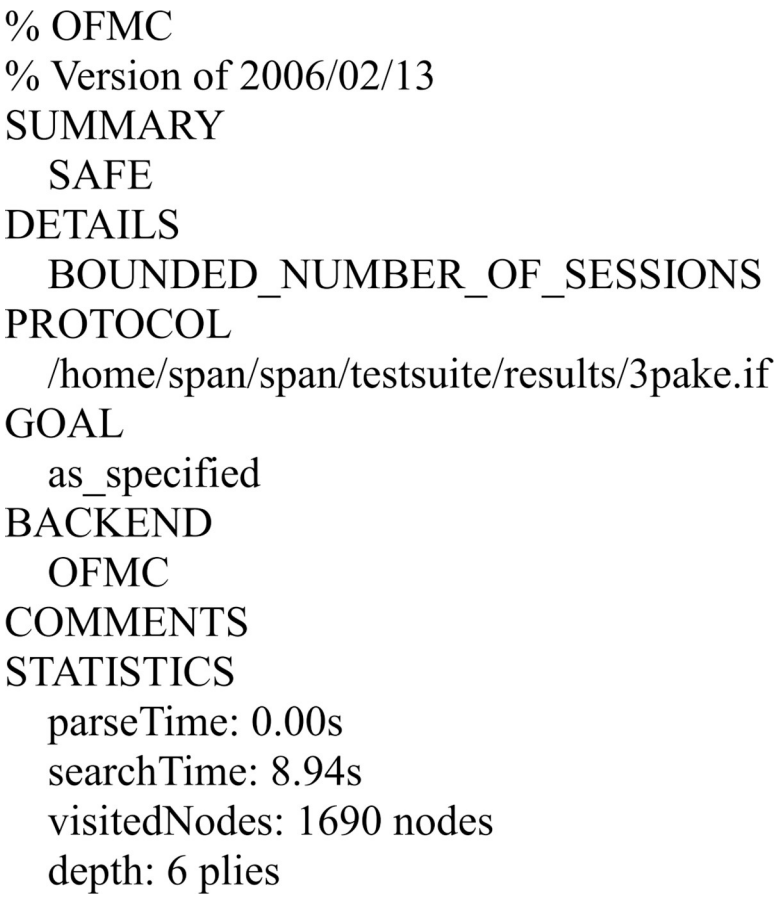


**Fig 7. The result of the analysis using OFMC back-end.**

https://doi.org/10.1371/journal.pone.0213976.g007

**Privileged insider attack.** The proposed scheme is secure against the privileged-insider attack. In the registration phase of the proposed scheme, only the user's identifier is transmitted to the server through a secure channel and the user's password is not transmitted to the server. Therefore, the privilege insider of the server cannot know the user's password. Therefore, the proposed scheme is secure against this attack.

**Stolen verifier attack.** The proposed scheme is secure against stolen verifier attack. In the proposed scheme, there is no user registration table to authenticate user in the server. Therefore, the proposed scheme is secure against stolen verifier attack.

**User impersonate attack.** The proposed scheme is secure against the user impersonate attack and the forgery attack.

In order to impersonate as user $A$, the attacker $C$ changes $K_A$ to $K_C$, and sends a message $\{M_{AS}^*$ $(= E_{KCS}(ID_A, ID_B, Z_{AS}^*))$, $K_C\}$ to the server. The server receiving the message from attacker $C$ computes $K_{SC}$ from $K_C$ and decrypts $M_{AS}^*$ using $K_{SC}$ to obtain $ID_A$, $ID_B$ and $Z_{AS}^*$. Next, server computes $X_A = H(ID_A||s)$ and $Z_{AS} = H(ID_A||ID_B||K_A||X_A)$, and compares it with $Z_{AS}^*$. Therefore, the attacker has to know $X_A = H(ID_A||s)$ or $s$.

However, since $s$ is a secret key of the server and $X_A$ is a secret data that only user $A$ has, the attacker $C$ cannot know it, and thus the impersonate attack is impossible. Also, even if an attacker attempts to impersonate as the user $B$, he does not know $X_B$ or $s$, so he cannot achieve the attack as before.

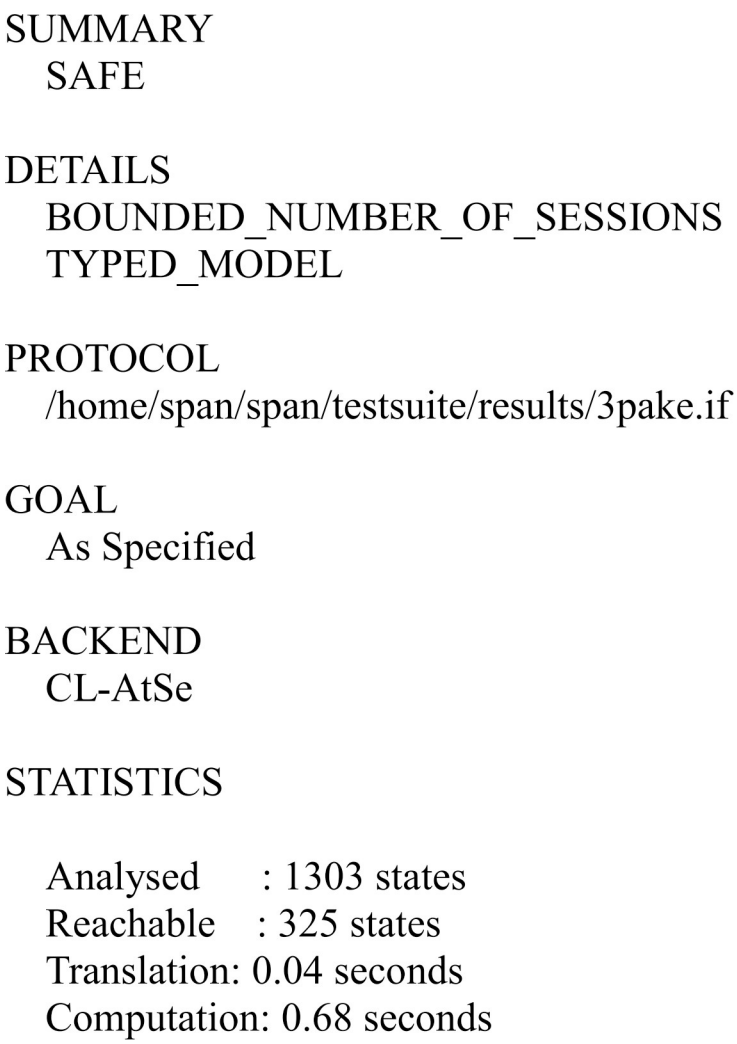


**Fig 8. The result of the analysis using CL-AtSe back-end.**

https://doi.org/10.1371/journal.pone.0213976.g008

**Man-in-the-middle attack.** As above, since an attacker $C$ cannot know $X_A = H(ID_A||s)$, $X_B = H(ID_B||s)$ or $s$, so he cannot modify the sender's message or cannot change $K_A$ and $K_B$, and cannot achieve the man-in-the-middle attack.

**Replay attack.** If an attacker $C$ sends the previous message $\{M_{AS}^*, T_a^*(x)\}$ of the user $A$, according to *CDLP* and *CDHP* assumptions, he cannot know $a^*$, so he does not calculate $Z_{AB}$ in the fourth message of the proposed scheme.

If an attacker $C$ sends the previous message $\{M_{BS}^*, T_b^*(x)\}$ of the user $B$, $Z_{BS}^*$ is calculated as $Z_{BS}^* = H(ID_B||R_A^*||R_B^*||X_B)$. Since $Z_{BS}$ is related to $R_A$ and the server verifies the correctness of $Z_{BS}$, it is impossible for the attacker $C$ to achieve the replay attack.

**Perfect forward security of session key.** In the proposed scheme, the session key $K_{AB}$ is calculated as $K_{AB} = T_a(K_B) = T_{ab}(x)\ mod\ p$. It contains the random numbers $a$ and $b$ that are generated for each session.

Therefore, the proposed scheme provides the perfect forward secrecy of session key.

**Known key security.** In the proposed scheme, the session key $K_{AB}$ is calculated as $K_{AB} = T_a(K_B) = T_{ab}(x)\ mod\ p$. It contains the random numbers $a$ and $b$ that are generated for

**Table 2. Comparison of the computational cost between the proposed scheme and other 3PAKE scheme.**

| | Xie et al.[23] | Lu et al.[31] | Li et al.[38] | Amin et al.[39] | Islam et al.[41] | proposed |
|---|---|---|---|---|---|---|
| A | 3te + 2ts + 4th | 3tc + 4ts + 4th | 4tc + 1tm + 5th | 8th | 2tc + 4ts + 2th | 3tc + 2ts + 6th |
| B | 3te + 2ts + 5th | 2tc + 3ts + 5th | 4tc + 1tm + 5th | 9th | 2tc + 4ts + 2th | 3tc + 2ts + 6th |
| S | 2te + 4ts + 7th | 5tc + 5ts+ 7th | 4tc + 2tq + 5th | 10th | 4ts + 3th | 2tc + 4ts + 6th |
| Total | 8te + 8ts+16th | 10tc + 12ts+ 16th | 12tc+2tm+2tq+15th | 27th | 4tc + 12ts + 7th | 8tc + 8ts + 18th |
| Round | 4 | 5 | 6 | 4 | 4 | 4 |
| Messages | 5 | 7 | 6 | 6 | 8 | 4 |

https://doi.org/10.1371/journal.pone.0213976.t002

each session. Even if an attacker knows previous session key, he cannot calculate a new session key.

## 6. Performance comparisons

This section compares the computational cost and security performance of the proposed scheme with the recent similar 3PAKE techniques [23, 31, 38, 49, 50], of which three [23, 31, 38] attempted to provide user anonymity and others [49, 50] use smart card. The notations used for comparison of computational cost are as follows.

$t_c$: time needed for Chebyshev polynomial operation

$t_e$: time needed for a scalar multiplication on elliptic curve

$t_s$: time needed for symmetric encryption/decryption operation

$t_m$: time needed for a modular squaring operation

$t_q$: time needed for a square root modulo N operation

$t_h$: time needed for one-way hash function operation

Table 2 shows the comparison of the computational cost of the six schemes, including the proposed scheme.

Table 3 shows the comparative evaluation of the security function between the proposed scheme and other 3PAKE schemes.

As shown in Table 2 and Table 3, the proposed scheme outperforms the other schemes in terms of the security functions presented. Xie's scheme provides user anonymity, but his scheme is vulnerable to the privileged insider attack. Lu et al.'s scheme attempted to provide user anonymity, but did not achieve it. There are weaknesses at the session key establishment phase and the password change phase of his scheme. Li's scheme provides user anonymity, but

**Table 3. Comparative evaluation of the security function between the proposed scheme and other 3PAKE schemes.**

| | Xie et al.[23] | Lu et al.[31] | Li et al.[38] | Amin et al.[39] | Islam et al.[41] | proposed |
|---|---|---|---|---|---|---|
| Provision of User anonymity | Yes | No | Yes | No | No | Yes |
| Protection of Privileged insider attack | No | Yes | Yes | No | Yes | Yes |
| Protection of password guessing attack | Yes | Yes | Yes | Yes | Yes | Yes |
| Protection of User impersonate attack | Yes | Yes | Yes | Yes | Yes | Yes |
| Provision of Password change phase | No | Yes | Yes | Yes | Yes | Yes |
| Secrecy of Password change phase | - | No | Yes | Yes | Yes | Yes |
| Password change without server's help | - | No | No | Yes | Yes | Yes |
| Without timestamp | Yes | Yes | Yes | Yes | No | Yes |
| Using smart card | No | No | No | Yes | Yes | Yes |

https://doi.org/10.1371/journal.pone.0213976.t003

in his scheme there are more rounds, messages and computational cost than our proposed scheme. Amin's and Islam's scheme are superior to our proposed scheme in terms of computational cost, but do not provide user anonymity for key exchange.

## 7. Conclusion

In this paper, we analyse the Lu et al.'s scheme and point out its weakness, and propose a round-effective 3PAKE protocol based on chaotic maps using smart card to provide with user anonymity. In the proposed scheme, there is no information related to the user's password at the server side and users share the secret key with the server, which is derived by the server's secret key and his identifier. The proposed scheme is more efficient than other schemes in terms of number of rounds and computational cost, and it is formally analysed based on BAN logic and AVISPA tool, and can protect against various attacks as shown through informal security analysis. The proposed scheme is suitable for authentication and key agreement in a wireless network environment.

## Author Contributions

**Conceptualization:** Kyongsok Pak, Songho Pak, Myongsuk Pak.

**Formal analysis:** Myongsuk Pak.

**Methodology:** Cholman Ho, Choljin Hwang.

**Project administration:** Songho Pak.

## References

1. Bellovin SM, Merritt M. Encrypted key exchange: password-based protocols Secure Against dictionary attacks. IEEE Security and Privacy Magazine. 1992;72–13
2. Zhu H, Hao X. A provable authenticated key agreement protocol with privacy protection using smart card based on chaotic maps. Nonlinear Dyn. 2015; 81(1–2):311–11
3. Maitra T, Obaidat MS, Islam SH, Giri D, Amin R. Security analysis and design of an efficient ECC-based two-factor password authentication scheme. Secur Commun Netw. 2016; 9(17):4166–16
4. Wang C, Zhang X, Zheng Z. Cryptanalysis and Improvement of a Biometric-Based Multi-Server Authentication and Key Agreement Scheme. Plos One. 2016; 11(2):e0149173 https://doi.org/10.1371/journal.pone.0149173 PMID: 26866606
5. Guo H, Wang P, Zhang X, Huang Y, Ma F. A robust anonymous biometric-based authenticated key agreement scheme for multi-server environments. Plos One. 2017; 12(11):e0187403 https://doi.org/10.1371/journal.pone.0187403 PMID: 29121050
6. Yang L, Zheng Z. Cryptanalysis and improvement of a biometrics-based authentication and key agreement scheme for multi-server environments. Plos One. 2018; 13(3):e0194093 https://doi.org/10.1371/journal.pone.0194093 PMID: 29534085
7. Huang HF. A simple three-party password-based key exchange protocol. Int J Commun Syst. 2009; 22 (7):857–6
8. Chang TY, Hwang MS, Yang WP. A communication-efficient three-party password authenticated key exchange protocol. Inform Sciences. 2011; 181(1):217–10
9. Lee TF, Hwang T. Simple password-based three-party authenticated key exchange without server public keys. Inform Sciences. 2010; 180(9):1702–13
10. Yoon EJ, Yoo KY. Cryptanalysis of a simple three-party password-based key exchange protocol. Int J Commun Syst. 2011; 24(4):532–11
11. Pu Q, Wang J, Wu S, Fu J. Secure verifier-based three-party password authenticated key exchange. Peer Peer Netw Appl. 2013; 6(1):15–11
12. Tso R. Security analysis and improvements of a communication-efficient three-party password authenticated key exchange protocol. J Supercomput. 2013; 66(2):863–12
13. Youn TY, Kang E, Lee C. Efficient three-party key exchange protocols with round efficiency. Telecommun Syst. 2013; 52(2):1367–10

14. Farash MS, Attari MA. An efficient client-client password-based authentication scheme with provable security. J Supercomput. 2014; 70(2):1002–21
15. Heydari M, Sadough SMS, Farash MS, Chaudhry SA, Mahmood K. An efficient password-based authenticated key exchange protocol with provable security for mobile client-client networks. Wireless Pers Commun. 2016; 88(2):337–20
16. Zhao J, Gu D. Provably secure three-party password-based authenticated key exchange protocol. Inform Sciences. 2012; 184(1):310–14
17. Wu S, Chen K, Zhu Y. Enhancements of a three-party password-based authenticated key exchange protocol. Int Arab J Inf Techn. 2013; 10(3):215–7
18. Xiong H, Chen Y, Guan Z, Chen Z. Finding and fixing vulnerabilities in several three-party password authenticated key exchange protocols without server public keys. Inform Sciences. 2013; 235():329–12
19. Wu S, Pu Q, Wang S, He D. Cryptanalysis of a communication-efficient three-party password authenticated key exchange protocol. Inform Sciences. 2012; 215():83–14
20. Wu S, Chen K, Pu Q, Zhu Y. Cryptanalysis and enhancements of efficient three-party password-based key exchange scheme. Int J Commun Syst. 2013; 26(5):674–13
21. Tan Z. A communication and computation-efficient three-party authenticated key agreement protocol. Secur Commun Netw. 2013; 6(7):854–10
22. Wang ZH, Huo ZQ, Shi W. Security analysis and enhancements of a three-party authenticated key agreement protocol. Acta Scientiarum Technology. 2015; 37(3):329–8
23. Xie Q, Hu B, Dong N, Wong DS. Anonymous three-party password authenticated key exchange scheme for telecare medical information systems. Plos One. 2014; 9(7):e102747 https://doi.org/10.1371/journal.pone.0102747 PMID: 25047235
24. Lou DC, Huang HF. Efficient three-party password-based key exchange scheme. Int J Commun Syst. 2011; 24(4):504–9
25. Liu T, Pu Q, Zhao Y, Wu S. Ecc-based password-authenticated key exchange in the three-party setting. Arab J Sci Eng. 2013; 38(8):2069–9
26. Marcos A, Simplicio JR, Sakuragui RM. Cryptanalysis of an efficient three-party password-based key exchange scheme. Int J Commun Syst. 2012; 25(11):1443–7
27. Farash MS, Attari MA. An efficient and provably secure three-party password-based authenticated key exchange protocol based on chebyshev chaotic maps. Nonlinear Dyn. 2014; 77(1–2):399–13
28. Lee CC, Li CT, Chiu ST, Lai YM. A new three-party-authenticated key agreement scheme based on chaotic maps without password table. Nonlinear Dyn. 2015; 79(4):2485–11
29. Li X, Niu J, Kumari S, Khan MK, Liao J, Liang W. Design and analysis of a chaotic maps-based three-party authenticated key agreement protocol. Nonlinear Dyn. 2015; 80(3):1209–12
30. Xie Q, Zhao J, Yu X. Chaotic maps-based three-party password-authenticated key agreement scheme. Nonlinear Dyn. 2013; 74(4):1021–7
31. Lu Y, Li L, Zhang H, Yang Y. An extended chaotic maps-based three-party password-authenticated key agreement with user anonymity. Plos One. 2016; 11(4):e0153870 https://doi.org/10.1371/journal.pone.0153870 PMID: 27101305
32. Lee CC, Li CT, Hsu CW. A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps. Nonlinear Dyn. 2013; 73(1–2):125–8
33. Farash MS, Attari MA, Kumari S. Cryptanalysis and improvement of a three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps. Int J Commun Syst. 2017; 30(1)
34. Hu X, Zhang Z. Cryptanalysis and enhancement of a chaotic maps-based three-party password authenticated key exchange protocol. Nonlinear Dyn. 2014; 78(2):1293–8
35. Lai H, Orgun MA, Xiao J, Pieprzyk J, Xue L, Yang Y. Provably secure three-party key agreement protocol using chebyshev chaotic maps in the standard model. Nonlinear Dyn. 2014; 77(4):1427–13
36. Lee TF, Lin CY, Lin CL, Hwang T. Provably secure extended chaotic map-based three-party key agreement protocols using password authentication. Nonlinear Dyn. 2015; 82(1–2):29–10
37. Lee TF. Efficient three-party authenticated key agreements based on chebyshev chaotic map-based diffie-hellman assumption. Nonlinear Dyn. 2015; 81(4):2071–8
38. Li CT, Chen CL, Lee CC, Weng CY, Chen CM. A novel three-party password-based authenticated key exchange protocol with user anonymity based on chaotic maps. Soft Comput. 2017:1–12
39. Diffie W, Hellman M. New directions in cryptography. IEEE T Inform Theory. 1976; 22(6):644–11
40. Koblitz N. Elliptic curve cryptosystems. Math Comput. 1987; 48(177):203–7

41. Gura N, Patel A, Wander A, Eberle H, Shantz SC. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. Lecture Notes in Computer Science. 2004; 4:119–14
42. Zhang L. Cryptanalysis of the public key encryption based on multiple chaotic systems. Chaos Soliton Fract. 2008; 37(3):669–6
43. Mason JC, Handscomb DC. Chebyshev polynomials. London: Chapman & Hall/CRC Press; 2003.
44. Liu L. The Arithmetic Performance Test and Analysis on Finite Fields Chebyshev Polynomials. Journal of Communication University of China. 2012: 19(4):54–5
45. Chatterjee S, Roy S, Das AK, Chattopadhyay S, Kumar N, Vasilakos AV. Secure Biometric-Based Authentication Scheme using Chebyshev Chaotic Map for Multi-Server Environment. IEEE Transactions on Dependable & Secure Computing. 2016; PP(99):1–15
46. Kumari S, Wu F, Das AK, Arshad H, Khan MK. A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps. Future Generation Computer Systems. 2016; 63(C):56–20
47. Li JL, Zhang WG, Kumari S, Choo KKR, Hogrefe D. Security analysis and improvement of a mutual authentication and key agreement solution for wireless sensor networks using chaotic maps. T Emerg Telecommun T. 2018; (15):e3295
48. Xie Q, Hu B, Wu T. Improvement of a chaotic maps-based three-party password-authenticated key exchange protocol without using server's public key and smart card. Nonlinear Dyn. 2015; 79(4):2345–14
49. Amin R, Biswas GP. Cryptanalysis and design of a three-party authenticated key exchange protocol using smart card. Arab J Sci Eng. 2015; 40(11):3135–15
50. Islam SH. Design and analysis of a three party password-based authenticated key exchange protocol using extended chaotic maps. Inform Sciences. 2015; 312:104–27
51. Yang H, Zhang Y, Zhou Y, Fu X, Liu H, Vasilakos AV. Provably secure three-party authenticated key agreement protocol using smart cards. Comput Netw. 2014; 58:29–10
52. 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; 74(1–2):419–9
53. Lai H, Xiao J, Li L, Yang Y. Applying semi-group property of enhanced chebyshev polynomials to anonymous authentication protocol. Math Probl Eng. 2012;2012
54. Odelu V, Das AK, Goswami A. An efficient biometric-based privacy-preserving three-party authentication with key agreement protocol using smart cards. Secur Commun Netw. 2015; 8(18):4136–21
55. Jin ATB, Ling DNC, Goh A. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. Pattern Recogn. 2004; 37(11):2245–11
56. Lumini A, Nanni L. An improved BioHashing for human authentication. Pattern Recogn. 2007; 40 (3):1057–9
57. Mishra D, Das AK, Mukhopadhyay S. A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. Expert Syst Appl. 2014; 41(18):8129–15
58. Amin R, Biswas GP. A Novel User Authentication and Key Agreement Protocol for Accessing Multi-Medical Server Usable in TMIS. J Med Syst. 2015; 39(3):1–17
59. Burrows M, Abadi M, Needham R. A logic of authentication. ACM T Comput Syst. 1989; 23(5):1–13.
60. AVISPA: Automated Validation of Internet Security Protocols and Applications. http://www.avispa-project.org/ (accessed on January 2019)