Improved anonymity preserving three-party mutual authentication key exchange protocol based on chaotic maps

Kyong-Sok Pak*, Mi-Hyang Kim*, Song-Ho Pak*, Chol-Man Ho*

Faculty of Information Science, Kim Il Sung University, Pyongyang, Democratic People’s Republic of Korea

* These authors contributed equally to this work.

ks.pak0228@ryongnamsan.edu.kp

Abstract

Three-party authentication key exchange is a protocol that allows two users to set up a session key for encrypted communication by the help of a trusted remote server. Providing user anonymity and mutual authentication in the authentication key exchange is important security requirements to protect users’ privacy and enhance its security performance. Recently Li proposed a chaotic maps-based authentication key exchange protocol which attempts to provide mutual authentication and user anonymity, but we found that there were some faults in the key exchange phase and password change phase of his scheme. We prove that Li’s scheme does not provide user anonymity and that the user’s privacy information is disclosed, and propose enhanced three-party authentication key exchange protocol that provides user anonymity and we analyse its security properties and verify its validity based on BAN logic and AVISPA tool.

1. Introduction

Authentication key exchange is one of the important issues to ensure the confidentiality of network security as a way of sharing the session key to perform encryption communication between communication parties in public network.

Researchers have done a lot of work on the two-party authentication key exchange (2PAKE) scheme (only two parties participate in key exchange) [1–5] and three-party authentication key exchange (3PAKE) scheme (except two communicating parties, the trusted third party server participates in key exchange) [6–43]. The main focus in authentication key exchange is to propose a clear authentication and a secure key exchange between participants. Key exchange is a process of setting up a session key to encrypt a message exchanged between participants and only two parties must share the key and the security of the key must be guaranteed. Typical encryption methods used for key exchange include secret-key encryption [26–43] and public-key encryption. Public-key encryption methods include in detail modular exponential operation-based schemes [6–13], elliptic curve encryption-based schemes [14, 15, 30–33, 34], chaotic maps-based schemes [16–25] and bilinear pairing-based schemes [34, 36]. User authentication is a key issue in authentication key exchange as a process of verifying
whether a user is legal or not, where it is important to use authentication factor. Authentication factors include knowledge-based factors (e.g., registered passwords), ownership-based factors (e.g., smart card), biometric infrastructures (fingerprints, irises, etc.) [33]. According to the number of authentication factors, it is classified into single factor authentication [6–13, 16, 20, 34], two-factor authentication [14, 15, 26, 29, 38–42, 44], three-factor authentication [1, 3, 4, 24, 25, 28, 30, 32, 33, 37].

Recently, with the introduction of technologies such as peer-to-peer, cloud computing, wireless sensor network, and Internet of Things (IoT), researchers are further investigating 3PAKE.

1.1 Related work

Password-based authentication key exchange is a traditional method, and many researchers have proposed password-based authentication key exchange methods [6–13, 16, 20, 34]. However, several security disadvantages have been revealed in the authentication key exchange scheme using only passwords.

Tallapally [6] proposed a simple 3PAKE protocol based on password in wireless communication networks, however, Farash [7] has revealed that Lu’s scheme cannot detect online and offline password guessing attacks, and he has improved their scheme, but his scheme was also found to be vulnerable to offline password guessing attacks by Lu [8]. Lu proposed an improved scheme, but his scheme was still vulnerable to offline password guessing attacks [9].

Youn [10] proposed a 3PAKE protocol based on password and exponential operation. However, Heydari [11] pointed out that Youn’s scheme is vulnerable to user impersonate attack. Heydari proposed a modified 3PAKE protocol that overcame the limitations of the Youn’s scheme. However, his scheme also does not provide user anonymity because user’s identity is disclosed in the key exchange phase. Lin et al. [12] proposed verifier-based 3PAKE with low computational cost and transfer cost based on password and modulator exponential operation. However, Chiou [13] pointed out that Lin’s scheme does not provide anonymity and untraceability and is not computationally efficient, and proposed 3PAKE that provides anonymity and untraceability by implementing message encryption with long term key. However, since his scheme also performs key exchange [45] based on modular exponential operation, the computation is still not efficient.

Researchers used the Elliptic Curve Cryptography (ECC) [46] and Chebyshev Chaotic Maps (CCM) [47, 48] much more efficient compared to modular exponential operations. ECC encryption is fast because of its much smaller key length at the same encryption intensity compared to modular exponential operations. Chebyshev Chaotic Maps has a lower public parameter for encryption compared to ECC and is simple to implement and is convenient to apply in portable terminal-system environments.

Wu [14] proposed a key agreement scheme for mobile user roaming service in global mobility networks based on ECC. In his scheme user’s dynamic identity is updated in each session. However, Gupta [15] pointed out that Wu’s scheme fails to support untraceability and it has inefficient typo-detection.

Xie [16] proposed a 3PAKE protocol based on chaotic maps with user password. However, Lee [17] found that Xie’s scheme is vulnerable to offline password guessing attacks and does not provide user anonymity, and proposed a 3PAKE protocol that does not use passwords that overcome their shortcomings. In Lee’s scheme, user privacy is generated by combining the server’s secret key and the user’s identification, it is used to authenticate the corresponding user. However, Xie [18] found that Lee’s scheme is vulnerable to user impersonate attacks, and
Jabbari [19] showed that Lee’s scheme is vulnerable to internal user impersonate attacks and does not provide anonymity.

Farash [20] proposed 3PAKE based on Chebyshev chaotic maps with user password. In his scheme user authentication verifier is generated by combining server privacy with user’s identifier and user’s password. However, Xie [21] and Li [22] found that Farashi’s scheme is capable of user impersonate attacks and offline password guessing attacks. Xie proposed an updated scheme based on chaotic maps overcoming the disadvantages of Farashi’s scheme. However, his scheme was also found by Lu [23] that offline password guessing attacks and user impersonate attacks are possible and user anonymity is not provided. Lu’s scheme encrypts a message with a secret key generated from the server’s public key based on the chaotic map to provide anonymity. However, his scheme has defects in protocol design [24].

To overcome the disadvantages of user authentication using passwords, researchers proposed 3PAKE protocols that combine smart card and biometric with user’s password to authenticate user.

In 2014, Xue [26] analysed Li [27] scheme that proposed a dynamic identifier-based 3PAKE in a multi-server environment and demonstrated that his scheme is vulnerable to attacks such as denial-of-service, internal attack, smart card attack, eavesdropping attack, masquerade attack. He also proposed an authentication key exchange scheme between a client and a service provider based on pseudo dynamic user identity using smart cards and user’s passwords in a multi-server environment. However, Gupta [28] found that his scheme is vulnerable to known password attacks, stolen smart card attacks and user impersonate attacks. In addition, Amin [29] also pointed out that Xue’s scheme does not provide anonymity, is vulnerable to offline password guessing attacks, privileged insider attacks, session key disclose attacks, and user impersonate attacks and has some defects in the authentication phase. Gupta proposed a hash function-based 3PAKE in a multi-service environment with user passwords and smart cards, but Tomar [30] demonstrated that his scheme is vulnerable to DoS attack, stolen smart card attack, user impersonate attacks, and does not provide perfect forward security.

Challa [32] also proposed a signature-based 3PAKE in IoT with password, smart card and biometric. However, Jia [33] pointed out that Challa’s scheme does not provide anonymity and untraceability and is vulnerable to user impersonate attacks, stolen smart card attacks, offline password guessing attacks, and attacks in the password change phase. Jia proposed a signature-based 3PAKE protocol that provides anonymity by updating Challa’s scheme. To provide anonymity, Jia used XOR operations and applied signature based on elliptic curve cryptography. Jia [34] proposed a 3PAKE scheme in fog-driven IoT based on Bilinear Pairing, whose scheme indicated by Ma [35] that it is computationally expensive because of Bilinear Pairing, and proposed a scheme that does not use Bilinear Pairing. Reddicherla [36] also proposed authentication key exchange scheme in Heterogeneous network based on Bilinear Pairing, but it is also not efficient because of the high computational cost.

Researchers also proposed key exchange scheme based on secret-key encryption without public-key encryption to implement 3PAKE in a portable terminal environment with narrow bandwidth and limited storage capacity, such as wireless communication environment or IoT. Key exchange based on secret-key encryption is much more advantageous in terms of computational cost because it does not use high-computational public-key encryption.

Chuang [37] proposed a 3PAKE scheme that provides anonymity with password, smart card and biometric in a multi-service environment. He proposed an authentication key exchange scheme that provides anonymity without public-key encryption in protocol design. However, his scheme was found by Amin [29] to be vulnerable to user impersonate attacks and vulnerable to session key disclose attacks. Amin analysed the disadvantages of the scheme
proposed by Xue [26] and Chuang [37] and proposed an improved lightweight authentication scheme. His scheme provided anonymity using smart cards and passwords without public key encryption.

In 2018, Wei [38] also proposed a 3PAKE protocol that provides anonymity without public-key cryptography to reduce computational cost. In 2019, Yang [39] proposed a lightweight 3PAKE protocol that provides perfect forward security using only XOR and hash functions in a WSN environment.

1.2 Motivation and our contribution

The authentication scheme with password, smart card and biometric is effective in systems that require high security performance. However, most schemes using smart cards are vulnerable to stolen smart card attacks [26, 28, 32], and most schemes are vulnerable to some known attacks.

It is still a challenge for researchers to design protocols that are secure against various attacks in various environments while providing anonymity and untraceability. Many schemes attempted to provide anonymity and traceability, but failed [12, 14, 17, 20, 23, 32, 43].

Recently, in 2018, Li [38] proposed a chaotic maps-based 3-PAKE that provides anonymity with password and smart cards. In his scheme, users share user’s credentials related to user’s identity, user’s password and server’s secret with the server, and chaotic maps is used for exchanging session key. He also used modulo square operations and square root operations based on Chinese Remainder Theorem to encrypt the message providing anonymity and untraceability. However, there are drawbacks in his protocol.

We have analysed the disadvantages of the Li’s scheme and demonstrated that the user’s authentication verifier is disclosed by an internal attacker, providing anonymity is failed and that the password modification is not successful by blocking attacks in the password change phase. We design an enhanced 3PAKE protocol that overcomes several security disadvantages of Li’s scheme, and is resistant to various attacks. In this paper, we propose a strong mutual authentication between server and users to overcome insider attacks, and a re-registration phase that allows users to re-register without altering their identity. Then, we analyse the security properties of our scheme and verify its validity using Ban-Logic [49] and AVISPA [50] tools and show the results of comparative analysis with previous works.

2. Preliminaries

This section describes Chebyshev chaotic maps and Bio-hashing functions.

2.1 Chebyshev polynomials

Chebyshev polynomial $T_n(\alpha)$ is defined as follows [47].

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

Chebyshev polynomials satisfy the following recursive relationship [47].

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

$$T_0(\alpha) = 1, T_1(\alpha) = \alpha$$
2.2 The property of Chebyshev polynomials

Chebyshev polynomials have the following two properties [47, 48].

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

Semi-group property: For \( u, v \in \mathbb{N} \) and any \( \alpha \in [-1,1] \), \( T_u(T_v(\alpha)) = T_{uv}(\alpha) = T_v(T_u(\alpha)) \).

2.3 Enhanced Chebyshev polynomials

The semi-group property holds for Chebyshev polynomials on the interval \((-1, +1)\), which can enhance the property as follows [48]:

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

\[
T_u(T_v(\alpha)) = T_{uv}(\alpha) = T_v(T_u(\alpha)) \mod p(u, v \in \mathbb{N}).
\]

2.4 Computational problems based on Chebyshev polynomials

CDLP (Chaotic maps-based Discrete Logarithm problem): For given two real numbers \( \alpha \) and \( \beta \), it is infeasible to find the integer \( n \) by any polynomial time bounded algorithm, where \( \beta = T_n(\alpha) \mod p \) [48].

CDHP (Chaotic maps-based Diffie-Hellman problem): For given three elements \( \alpha, T_m(\alpha) \mod p \) and \( T_n(\alpha) \mod p \), it is infeasible to compute the value \( T_{mn}(\alpha) \mod p \) by any polynomial time bounded algorithm [48].

2.5 Bio-hashing and Fuzzy Extractor function

Biometric indicators have an advantage over traditional user identification methods, because these have some inherent attributes that cannot be easily shared and every person has unique biometric attributes [51]. Generally, imprint biometric characteristics (face, fingerprint, palmprint etc.) may not be exactly same at each time since it might be change at some environment. To solve this problem, Lumini et al. [52] proposed and updated Bio-hashing, which is used to map a user’s biometric features to a user-specific random vectors. Recently many researchers [3, 24] have proposed authentication key exchange schemes based on Bio-Hashing.

Dodis et al. [53] proposed a scheme based on Fuzzy Extractor, which consists of two functions (\( \text{Rep}, \text{Gen} \)). The function \( \text{Gen} \) extracts biometric input \( B \) and outputs a nearly random binary string \( R \) and an auxiliary binary string \( P \). Then function \( \text{Rep} \) recovers \( R \) with the corresponding auxiliary string \( P \) and biometric \( B' \). If \( \text{dist} (B, B') \leq t \) and \( \text{Gen}(B) \rightarrow <R, P> \), then we have \( \text{Rep} (B', P) = R \). Fuzzy Extractor is also used in many authentication schemes [1, 3, 4, 30, 32, 33].

2.6 Adversary model

For the security analysis of authentication protocols, several adversary models have been proposed, such as Dolev-Yao adversary model [54], side-channel technology [55], password guessing attack [56], and insider attacker model [4, 33].

The Dolev-Yao attacker model defines the ability of an attacker in the public channel, and the side-channel technology enables an attacker to extract data stored in a smart card based on reverse engineering and power analysis [57, 58]. Also, the password guessing attack enables an attacker to guess a password from the information related to the user’s password under the
premise that the entropy of the password is low. An insider attacker is a legitimate user in the system and performs malicious actions.

In this subsection, the adversary model for security analysis of the previous work and the proposed scheme is described as follows.

1. An adversary can eavesdrop, modify, remove, block, and retransmit all messages sent on the public channel [54] and cannot access messages sent on the secure channel.

2. An adversary can extract all stored data from a lost or stolen smart card based on side channel technology [55, 57, 58].

3. An attacker can easily guess the user identity or password after obtaining information from an intelligent card or public channel according to [56].

4. An adversary can be a legitimate but malicious user or server in the system [4, 33].

3. Review of Li et al.’s scheme

This section shows that the scheme proposed by Li et al. [22] has some deficiencies. Li designed three-party password-based authentication key exchange protocol based on chaotic maps providing user anonymity. In his scheme, the information related to user’s password is registered with the server side. Also a modular squaring operation and a square root modulo based on the Chinese Remainder Theorem is used for user anonymity. However, his scheme has some faults in the session key exchange phase and the password change phase. Below is a brief description of the scheme proposed by Li et al. and its deficiencies.

3.1 Li et al.’s scheme

Notations used in his paper. Table 1 shows some notations used to describe Li et al.’s schemes.

System initialization. The server selects one private key $k$ and two secret large primes $(u, v)$, and publishes $\{p, \alpha, h(), N\}$.

Registration. Step 1: The user submits $\{U_i, h(\text{rm}_i, PW_i)\}$ to $S$, where $\text{rm}_i \in [1, p+1]$ is random number and $PW_i = T_{pw}(\alpha)$.

Table 1. Notations in Li et al.’s scheme.

| Notation | Description |
|----------|-------------|
| $S$ and $A, B$ | Trusted remote server and two users |
| $U_i$ | Identifier of user $i$ |
| $pw_i$ | Password of user $i$ |
| $k$ | Secret key of $S$ |
| $p$ | A large prime number chosen by $S$ |
| $u, v$ and $N = u \cdot v$ | Two large primes maintained by $S$ |
| $p$ | A large prime number chosen by $S$ |
| $T_n(\cdot)$ | Chebyshev polynomials of degree $n$ |
| $h(\cdot)$ | One-way hash function |
| $\text{SQR} (\cdot), \text{SQRT} (\cdot)$ | Modular squaring operation and square root modulo operation |
| $\oplus$ | XOR operator |

https://doi.org/10.1371/journal.pone.0273664.t001
Step 2: Upon receiving the message from the user, $S$ computes $VP_s = h(U_s, k) + h(rm_s, PW_s)$ mod $p$ and stores $(U_s, VP_s)$ in its database.

Step 3: The user stores $r_m$ into his end-user device.

**Authenticating and key exchange.**

**Step 1** The user $A$ chooses a random number $r_A \in [1, p + 1]$ and computes $R_A = T_{ra}(\alpha) \mod p$, $PW_A = T_{pwa}(\alpha) \mod p$ and $h(rm_A, PW_A)$, where $rm_A$ is retrieved from his end-user device. Then $A$ sends $M_1 = \text{SQR}(U_A, U_B, h(rm_A, PW_A), R_A) \mod N$ to $S$.

**Step 2** Upon receiving $M_1$ from $A$, $S$ obtains $(U_A, U_B, h(rm_A, PW_A)^*, R_A) = \text{SQR}(M_1)$ by using the Chinese Remainder Theorem with $u$ and $v$. Next, $S$ retrieves the stored $VP_A = h(U_A, k) + h(rm_A, PW_A)$ corresponding to $U_A$. If $VP_A = h(U_A, k) = h(rm_A, PW_A)^*$, then $S$ continues next step. $S$ chooses two random numbers $r_{s1}, r_{s2} \in [1, p + 1]$ and computes $R_{s1} = T_{r_{s1}}(\alpha) - h(rm_A, PW_A) \mod p$, $R_{s2} = T_{r_{s2}}(\alpha) - h(rm_B, PW_B) \mod p$ and $\mu_A = U_A + T_{r_{s2}}(\alpha) \mod p$. Then $S$ sends $M_2 = [\mu_A, R_A, R_{s2}]$ to $B$.

**Step 3** Upon receiving $M_2$ from $S$, $B$ computes $PW_B = T_{pwb}(\alpha) \mod p$, $h(rm_B, PW_B)$, $R_{s2}(\alpha) = R_{s2} + h(rm_B, PW_B)$ and $U_A = \mu_A + T_{r_{s2}}(\alpha)$. Then $B$ chooses a random number $r_B \in [1, p + 1]$ and computes $K_B = T_{rb}(\alpha) \mod p$, $K_{BS} = T_{rb}(T_{s2}(\alpha)) \mod p = T_{rb}(\alpha) \mod p$, $K_{BA} = T_{rb}(R_A) \mod p = T_{rb}(\alpha) \mod p$, $V_{BA} = h(0, U_B, U_A, R_B, R_A, K_{BA})$ and $V_{BS} = h(0, U_B, U_A, R_B, R_{s2}, V_{BA}, K_{BS})$. Then $B$ sends $M_3 = \text{SQR}(U_B, U_A, R_B, V_{BS}, V_{BA}) \mod N$ to $S$.

**Step 4** Upon receiving $M_3$ from $B$, $S$ obtains $(U_B, U_A, R_B, V_{BS}^*, V_{BA}^*) = \text{SQR}(M_3)$. $S$ computes $K_{SB} = T_{r_{s2}}(R_B) = T_{rb}(\alpha) \mod p$ and $V_{BS} = h(0, U_B, U_A, R_B, R_{s2}, V_{BA}, K_{BS})$. If $V_{BS} = V_{BS}^*$, $B$ is authenticated by $S$ and then $S$ computes $K_{SA} = T_{r_{s1}}(\alpha) \mod p$ and $V_{SA} = h(0, U_A, U_B, R_{s1}, R_B, V_{BA}, K_{SA})$. Then $S$ sends $M_4 = [R_{s1}, R_B, V_{BA}, V_{SA}]$ to $A$.

**Step 5** Upon receiving $M_4$ from $S$, $A$ computes $K_{AS} = T_{r_{s1}}(\alpha) - h(rm_A, PW_A) \mod p$ and $V_{SA} = h(0, U_A, U_B, R_{s1}, R_B, V_{BA}, K_{SA})$. If $V_{SA}$ equals received $V_{SA}^*$, $S$ is authenticated by $A$. Next, $A$ computes $K_{AB} = T_{ra}(\alpha) \mod p = T_{ra}(R_s) \mod p$ and $V_{BA} = h(0, U_B, U_A, R_B, R_A, K_{BA})$. If $V_{BA} = V_{BA}^*$, $B$ is authenticated by $A$. $A$ computes $V_{AB} = h(1, U_A, U_B, R_A, K_{AB})$ and $V_{AS} = h(1, U_A, U_B, R_A, V_{BA}, R_{s1}, V_{AB}, K_{AS})$. Then $A$ sends $M_5 = [V_{AS}, V_{AB}]$ to $S$.

**Step 6** After receiving $M_5$ from $A$, $S$ verifies if computed $h(1, U_A, U_B, R_A, R_{s1}, V_{AB}, K_{SA})$ equals received $V_{AS}$. If it holds, $A$ is authenticated by $S$. Then $S$ computes $V_{SB} = h(1, U_A, U_B, R_A, V_{BA}, K_{SB})$ and sends $M_6 = [V_{AS}, V_{SB}]$ to $B$.

**Step 7** After receiving $M_6$ from $B$, $S$ verifies if computed $h(1, U_A, U_B, R_A, V_{SB}, K_{SB})$ and $h(1, U_A, U_B, R_B, V_{BA}, R_{s1}, K_{AB})$ equals received $V_{SB}$. If it does, $B$ and $S$ are authenticated by $B$. Finally, $A$ computes the session key $SK_{AB} = h(2, U_A, U_B, R_A, R_B, K_{AB})$ and $B$ computes the session key $SK_{BA} = h(2, U_B, U_A, R_B, R_A, K_{BA})$.

**Password change phase.**

**Step 1** The user $A$ chooses a random number $r_A \in [1, p + 1]$ and computes $R_A = T_{ra}(\alpha) \mod p$, $PW_A = T_{pwa}(\alpha) \mod p$ and $h(rm_A, PW_A)$, Then $A$ sends $C_1 = \text{SQR}(U_A, h(rm_A, PW_A), R_A) \mod p$ to $S$.

**Step 2** Upon receiving $C_1$ from $A$, $S$ obtains $(U_A, h(rm_A, PW_A)^*, R_A) = \text{SQR}(C_1)$ by using the Chinese Remainder Theorem with $u$ and $v$. Next, $S$ verifies the received $h(rm_A, PW_A)^*$ with the stored $VP_A = h(U_A, k) + h(rm_A, PW_A)$ mod $p$ corresponding to $U_A$. If $VP_A = h(U_A, s) = h(rm_A, PW_A)^*$, $S$ accepts $A$’s request message $C_1$. Then $S$ chooses a random number $r_s \in [1, p + 1]$ and computes $R_s = T_{r_s}(\alpha) - h(rm_A, PW_A) \mod p$, $K_{SA} = T_{r_s}(R_A) \mod p$ and $V_{SA} = h(0, U_A, R_s, R_A, K_{SA})$. Then $S$ sends $C_2 = [R_s, V_{SA}]$ to $A$.

**Step 3** Upon receiving $C_2$ from $S$, $A$ computes $K_{SA} = T_{r_s}(R_s + h(rm_A, PW_A)) = T_{r_s}(\alpha) \mod p$ and verifies if computed $V_{SA} = h(0, U_A, R_s, R_A, K_{SA})$ equals received $V_{SA}^*$. If it holds, $S$ is authenticated by $A$. Next, $A$ selects a new password $pw_A^*$ and a new random number $rm_A^*$
and computes $V_{AS} = h(1, U_A, R_A, R_S, K_{AS}), P_{WA} = T_{pwA^*}(\alpha) \mod p$, and $h(rmA^*, P_{WA}^*)$. Then A sends $C_3 = \text{SQR}(V_{AS}, U_A, h(rmA^*, P_{WA}^*)) \mod p$ to $S$.

**Step 4** Upon receiving $C_3$ from A, S verifies if computed $V_{AS} = h(1, U_A, R_A, R_S, K_{AS})$ equals received $V_{AS}$. If it holds, S accepts A’s password change request, computes $R_1 = h(1, U_A, h(rmA^*, P_{WA}^*), K_{SA})$ and $V_{PA} = h(U_A, k) + h(rmA^*, P_{WA}^*) \mod p$ and replaces $V_P$ with $V_{PA}$. Then S sends $C_4 = \{\text{Accept}, R_1\}$ to A. Otherwise, S rejects A’s password change request, computes $R_2 = h(0, U_A, h(rmA^*, P_{WA}^*), K_{SA})$ and sends $C_5 = \{\text{Reject}, R_2\}$ to A. If the message is $C_4$, A verifies if computed $R_1 = h(1, U_A, h(rmA^*, P_{WA}^*), K_{SA})$ equals received $R_1^*$. If it holds, A confirms $p_{WA}^*$ as the new password and replaces $rm_A$ with $rm_A^*$ in his end-user device. Otherwise, A returns to **Step 1** and follows the process. If the message is $C_5$, A returns to **Step 1** with another new password and follows the process.

### 3.2 Faults of Li et al.’s scheme

Many attack models [54–58] have been proposed by researchers and based on them, cryptographic protocols [25, 33, 34, 38, 59, 60] have been analysed. Based on the adversary model presented in Section 2.6, we analyse Li et al.’s scheme. According to the adversary model, the adversary can eavesdrop and block all message sent on the public channel and he can be a legitimate user in the system. In this paper, we call such an adversary an insider adversary.

**Verifier disclosure attacks.** Li et al.’s scheme has a faults that user’s authentication verifier is disclosed to the insider adversary in the authentication and key exchange phase. In his scheme, $h(rmA, P_{W_1})$ is user’s authentication verifier, where $rm_A$ is stored into user’s end-device, $P_{W_1} = T_{pw}(\alpha) \mod p$ and $p_{W_1}$ is user’s password. However, $h(rmA, P_{W_1})$ is disclosed to the insider adversary.

The details of verifier disclosure attack in his scheme are described as follows.

**Step 1.** In order to exchange a session key with a legal user $A$, an insider adversary $C$ chooses a random number $rC \in [1, p + 1]$ and computes $R_C = T_{rc}(\alpha) \mod p, PW_C = T_{pwC}(\alpha) \mod p$ and $h(rmc, PW_C)$, where $rmC$ is retrieved from his end-user device. Then C sends $M_1 = \text{SQR}(U_C, U_A, h(rmc, PW_C), R_C) \mod N$ to $S$.

**Step 2.** Upon receiving $M_1$ from C, S obtains $(U_C, U_A, h(rmc, PW_C), R_C) = \text{SQR}(M_1)$. Next, S retrieves the stored $VP_C = h(U_C, k) + h(rmc, PW_C)$ corresponding to $U_C$. If $VP_C = h(U_C, k)$, then S continues next step. S chooses two random numbers $rS_1, rS_2 \in [1, p + 1]$ and computes $R_S = h(rmc, PW_C) \mod p, R_{S_2} = T_{rS_2}(\alpha) - h(rmA, PW_A) \mod p$ and $\mu_C = U_C \oplus T_{rS_2}(\alpha) \mod p$. Then S sends $M_2 = \{\mu_C, R_C, R_{S_2}\}$ to $A$.

**Step 3.** At this time, C intercepts $M_2 = \{\mu_C^*, R_C^*, R_{S_2}^*\}$ and computes as follows:

$U_C \oplus \mu_C^* = U_C \oplus U_C \oplus T_{rS_2}(\alpha) = T_{rS_2}(\alpha)$ (Because of $U_C$ is C’s identifier, C knows it.

Through checking for $R_C^*$ of $M_2$, C can verify that $M_2$ is a message generated at the server according to $M_1$.

$h(rmA, PW_A) = T_{rS_2}(\alpha) - R_{S_2}^* = T_{rS_2}(\alpha) - (T_{rS_2}(\alpha) - h(rmA, PW_A))$

As the result, C can obtain A’s authentication verifier $h(rmA, PW_A)$.

In this way, C can obtain all of legal users’ authentication verifier.

If an insider adversary wants to get an authentication verifier of user $A$, it is necessary to generate a message $M_1$ for exchanging session key with user $A$ according to the designed protocol, send it to the server, intercept the message $M_2$ from the server, and then compute it according to the procedure shown above.

**User impersonate attacks.** As shown above, since an insider adversary $C$ can obtain any of legal users’ authentication verifier through verifier disclosure attack, he can impersonate as any legal user.
If an insider adversary C wants to impersonate as a legal user A and communicate with B, he obtains the user A’s authentication verifier \( V_A = h(rm_A, PW_A) \) through the verifier disclosure attack as shown above before the authentication and key exchange phases.

In the authentication and key exchange phase, C works as follows

**Step 1.** C chooses a random number \( r_C \in \{1, p + 1\} \) and computes \( R_C = T_C(\alpha) \mod p \). After that, he can sufficiently make the message \( M_1 = SQR(U_A, U_B, h(rm_A, PW_A), R_C) \mod N \) by using the user A’s authentication verifier \( V_A = h(rm_A, PW_A) \). Then C impersonates A to sends \( M_1 \) to \( S \).

The process in steps 2, 3 and 4 is performed according to the protocol, B computes \( K_{BA}^* = T_{IB}(R_C) \mod p \) and \( V_{BA}^* = h(0, U_B, A, B, R_C, K_{BA}^*) \), S computes \( K_{SA}^* = T_{IS1}(C) \mod p \) and \( V_{SA}^* = h(0, U_A, U_B, S, R_C, R_B, V_{BA}^*, K_{SA}^*) \).

**Step 5:** C intercepts \( M_4 \) from \( S \) to A, C computes \( K_{AS}^* = T_C(S + h(rm_A, PW_A)) = T_{CGS1}(\alpha) \mod p \) and \( K_{AB}^* = T_{BC}(R_C) \mod p \).\( V_{AS}^* = h(1, U_A, U_B, R_C, R_B, K_{AB}^*) \) and \( V_{AS}^* = h(1, U_A, U_B, R_C, R_B, V_{BA}^*, K_{AS}^*) \). Then A sends \( M_5 = \{V_{AS}^*, V_{AB}^*\} \) to \( S \).

As the result, C can successfully impersonate as the user A.

**Failure of user anonymity.** An insider adversary C can obtain all of legal users’ authentication verifier through verifier disclosure attack as shown above. That is, C knows the authentication identifier \( V_i \) of any user \( U_i \). When a legal user A exchanges a session key with a legal user B, an insider adversary C can intercept \( M_2 = \{V_{A_i}, R_{A_i}, R_{S_i}\} \) that \( S \) sends to \( B \) and then computes as follows: For each authentication verifier \( V_i \) of user \( U_i \), C repeat the following calculation until \( U_A^* \) and \( U_i \) are equal.

\[
T_{iS2} \ast (x) = R_{S2} + V_i = R_{S2} + h(rm_i, PW_i) = (T_{iS2}(x) - h(rm_i, PW_i)) + h(rm_i, PW_i)
\]

\[
U_A^* = \mu_A \oplus T_{iS2} \ast (x) \mod p = U_A \oplus T_{iS2}(x) \oplus T_{iS2} \ast (x)
\]

If \( U_A^* \) and \( U_i \) are equal, C can know that current user’s identifier is \( U_i \).

As the result, C can know user A’s identifier \( U_A \).

**Weaknesses of password change phase.** In the Li et al.’s scheme, the information related to the user’s password is registered with the remote server and users can change their password in the password change phase. In the registration phase, the information related to the user \( U_i \)’s password stored on the server is \( VP_i = h(U_i, k) + h(rm_i, PW_i) \) (where \( h(rm_i, PW_i) \) is user authentication verifier) and this information is replaced with \( VP_i = h(U_i, k) + h(rm_i, PW_i) \) in the password change phase. However, an attacker can block the message \( C_5(\text{user’s request}) \) and \( C_4 \) or \( C_3(\text{server’s response}) \) in Step 4 of the password change phase, then the user cannot know whether his password is successfully changed or not. In this case, if the scheme decides that does not change user’s password, the attacker blocks the message \( C_4 \) or \( C_5 \), if the scheme decides that changes user’s password, the attacker blocks the message \( C_3 \). As a result, the user’s authentication verifier is different with the server’s one, the user cannot login to the server no more.

### 4. Proposed scheme

This section describes an enhanced 3PAKE protocol using smart card that overcomes the limitations of the Li et al.’s scheme. The proposed scheme has five phases: system initialization phase, registration phase, authentication and session key exchange phase, password change phase, and renew registration phase. Table 2 shows some notations used to describe the proposed schemes.
4.1 System initialization phase

1. \( S \) selects his secret key \( k_s \in [1, p+1] \) and computes public key \( P_s = T_{k_s}(\alpha) \).

2. \( S \) selects a large prime number \( p, \alpha \in Z_p \), \( H(\cdot) \) and \( E_K(\cdot)/D_K(\cdot) \).

\( S \) publishes \( \{ p, \alpha, P_s, T_n(\cdot), H(\cdot), E_K(\cdot), D_K(\cdot) \} \) as system’s parameters.

4.2 User registration phase

Fig 1 shows user registration process.

User \( A \) sends his identifier \( U_A \) to \( S \) via secure channel. \( S \) retrieves \( U_A \) in the user registration table to check whether user \( A \) has already been registered. If \( U_A \) does not exist in the user table

\[
\text{User } A/SC_A \quad \text{Server } S
\]

\[
\{U_A\} \quad \text{\textarrow} \quad X_A = H(U_A||N_a||k_s)
\]

\[
SC_A \leftarrow \{p, \alpha, P_s, X_A, T_n(\cdot), H(\cdot), E_K(\cdot), D_K(\cdot)\}
\]

\[
\text{Input } \quad pw_A, bm_A
\]

\[
G_A = H(U_A||pw_A||h(bm_A)) \oplus X_A
\]

\[
F_A = H(U_A||pw_A||h(bm_A)||X_A)
\]

\[
SC_A \leftarrow \{p, \alpha, P_s, G_A, F_A, T_n(\cdot), H(\cdot), E_K(\cdot), D_K(\cdot)\}
\]

Fig 1. User registration phase.

https://doi.org/10.1371/journal.pone.0273664.g001
Fig 2 shows the authentication and session key exchange steps of the proposed scheme.

### Step 1
User A connects his smart card $SC_A$ to the user end-device and inputs his identifier $U_A$, password $pw_A$ and biometrics $bm_A$. $SC_A$ computes:

$$X_A^* = G_A \otimes H(U_A \| pw_A \| h(bm_A)), F_A^* = H(U_A \| pw_A \| h(bm_A) \| X_A^*)$$

If $F_A \neq F_A^*$, $SC_A$ aborts the process. Otherwise $SC_A$ selects any $a \in [1, p+1]$ and computes:

$$T_a = T_a(x) \mod p, K_{AS} = T_a(P_s) = T_{aks}(x) \mod p$$

$$V_A = H(U_A \| U_b \| T_A \| X_A^*), M_{AS} = E_{KAS}(U_A, U_b, V_A)$$

A sends $M_1 = \{M_{AS}, T_a\}$ to S.

### Step 2
After receiving $M_1 = \{M_{AS}, T_a\}$ from A, S computes:

$$K_{AS} = T_{ks}(T_a) = T_{ksa}(x) \mod p, \{U_A^*, U_b^*, V_A^*\} = D_{KAS}(M_{AS}), X_A = H(U_A^* \| N_s \| k_s), V_A = H(U_A^* \| U_b^* \| T_A^* \| X_A^*)$$

S checks whether $V_A$ and $V_A^*$ are same. If $V_A \neq V_A^*$, S aborts the process. S chooses a random number $N_s \in [1, p+1]$ and computes $V_{SA} = H(N_s^* \| U_A^* \| U_b^* \| T_A^* \| X_A^*)$.

S sends $M_2 = \{V_{SA}, N_s\}$ to A.

### Step 3
After receiving $M_2 = \{V_{SA}^*, N_s^*\}$ from S, A computes:

$$V_{SA} = H(N_s^* \| U_A^* \| U_b^* \| T_A^* \| X_A^*)$$

A checks whether $V_{SA}$ and $V_{SA}^*$ are same. If $V_{SA} \neq V_{SA}^*$, A aborts the process. A computes $V_{AS} = H(U_A^* \| U_b^* \| T_A^* \| N_s^* \| X_A^*)$ and sends $M_3 = \{V_{AS}, N_s\}$ to B.

### Step 4
After receiving $M_3 = \{V_{AS}^*, N_s^*\}$ from A, B connects his smart card $SC_B$ to the user end-device and inputs his identifier $U_b$, password $pw_B$ and biometrics $bm_B$. $SC_B$ computes:

$$X_B^* = G_B \otimes H(U_B \| pw_B \| h(bm_B)), F_B^* = H(U_B \| pw_B \| h(bm_B) \| X_B^*)$$

If $F_B \neq F_B^*$, $SC_B$ aborts the process. Otherwise $SC_B$ computes:

$$T_B = T_b(x) \mod p, K_{BS} = T_b(P_s) = T_{bsk}(x) \mod p$$

$$V_B = H(U_B \| T_B \| N_s^* \| X_B^*), M_{BS} = E_{KBS}(U_B, N_s^*, V_B)$$

B sends $M_4 = \{M_{BS}, T_B, V_{AS}\}$ to S.

### Step 5
After receiving $M_4 = \{M_{BS}, T_B^*, V_{AS}^*\}$ from B, S computes:

$$K_{AS} = T_{ks}(T_B^*) = T_{ksa}(x) \mod p, \{U_B^*, N_s^*, V_B^*\} = D_{KBS}(M_{BS})$$

$$X_B = H(U_B^* \| N_s \| k_s), V_B = H(U_B^* \| T_B^* \| N_s^* \| X_B^*), V_{AS} = H(U_A \| U_B \| T_A \| N_s^* \| X_A^*)$$
Fig 2. Authentication and session key exchange phase of the proposed scheme.

https://doi.org/10.1371/journal.pone.0273664.g002
If $V_B \neq V_B^*$ or $V_{AS} \neq V_{AS}^*$, $S$ aborts the process. Otherwise $S$ authenticates $A$ and $B$, and chooses a random number $R_S$, then computes

$$V_{SB} = H(U_A \parallel U_B \parallel T_A \parallel T_B \parallel R_S \parallel X_B), M_{SB} = E_{KBS}(U_A, T_A, R_S, V_{SB}).$$

$$V_{SAB} = H(U_A \parallel U_B \parallel T_A \parallel T_B \parallel R_S \parallel X_A), M_{SA} = E_{KAS}(U_B, T_B, R_S, V_{SAB})$$

$S$ sends $M_S = \{M_{SB}, M_{SA}\}$ to $B$.

**Step 6.** After receiving $M_S = \{M_{SB}, M_{SA}\}$ from $S$, $B$ computes

$$\{U_A^*, T_A^*, R_S^*, V_{SB}^*\} = D_{KBS}(M_{SB}), V_{SB} = H(U_A^* \parallel U_B \parallel T_A^* \parallel T_B \parallel R_S^* \parallel X_B)$$

If $V_{SB} \neq V_{SB}^*$, $B$ aborts the process. Otherwise $B$ authenticates $S$ and $A$, and then computes

$$T_{BA} = T_A(T_A^*), T_{AB} = T_B(T_B^*), K_{BA} = H(T_{BA} \parallel R_S)$$

$$V_{BA} = H(U_A \parallel U_B \parallel T_A^* \parallel T_B \parallel R_S^* \parallel K_{BA})$$

$B$ sends $M_B = \{V_{BA}, M_{SA}\}$ to $A$.

**Step 7.** After receiving $M_6 = \{V_{BA}^*, M_{SA}\}$ from $B$, $A$ computes

$$\{U_B^*, T_B^*, R_S^*, V_{SAB}^*\} = D_{KAS}(M_{SA}), V_{SAB} = H(U_A \parallel U_B^* \parallel T_A \parallel T_B^* \parallel R_S^* \parallel X_A)$$

$$T_{AB} = T_A(T_B^*), T_{AB} = T_B(T_A^*), K_{AB} = H(T_{AB} \parallel R_S)$$

$$V_{AB} = H(U_A \parallel U_B^* \parallel T_A \parallel T_B^* \parallel R_S \parallel K_{AB})$$

If $V_{SAB} \neq V_{SAB}^*$ or $V_{BA} \neq V_{BA}^*$, $A$ aborts the process. Otherwise $A$ authenticates $B$ and $S$. $A$ sets $K_{AB}$ as a session key. Then $A$ computes $V_{AB} = H(U_B^* \parallel U_A \parallel T_B^* \parallel T_A \parallel R_S \parallel K_{AB})$.

$A$ sends $M_7 = \{V_{AB}^*\}$ to $B$.

**Step 8.** After receiving $M_7 = \{V_{AB}^*\}$ from $A$, $B$ computes

$$V_{AB} = H(U_B \parallel U_A \parallel T_B \parallel T_A \parallel R_S \parallel K_{BA})$$

If $V_{AB} \neq V_{AB}^*$, $B$ aborts the process. Otherwise $B$ authenticates $A$ and sets $K_{BA}$ as a session key.

### 4.4 Password change phase

User $A$ connects his smart card $SC_A$ to the user end-device and inputs his identifier $U_A$, password and biometrics $bm_A$. $SC_A$ computes $X_A^* = G_A \oplus H(U_A \parallel \text{npw}_A \parallel \text{h}(bm_A))$ and $F_A^* = H(U_A \parallel \text{npw}_A \parallel \text{h}(bm_A) || X_A)$, 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 $\text{npw}_A$. $SC_A$ computes $G_A^{\text{new}} = H(U_A \parallel \text{npw}_A \parallel \text{h}(bm_A)) \oplus X_A, F_A^{\text{new}} = H(U_A \parallel \text{npw}_A \parallel \text{h}(bm_A) || X_A)$ and replaces $\langle G_A, F_A \rangle$ of his memory with $\langle G_A^{\text{new}}, F_A^{\text{new}} \rangle$.

### 4.5 Re-registration phase

When a user registered with the server has lost or stolen his smart card, he needs to re-register with the server. But, some schemes [19, 22, 23] have not the re-registration phase or cannot re-register without changing his identifier, because the user’s secret consists of user’s identifier and server’s secret.

In the proposed scheme, as the user’s secret $X_A$ consists of user’s identifier, random number and server’s secret, users can re-register with the remote server without changing his identifier.
If a user wants to re-register with the server, he should only send his identifier to the server and register with the server following the proposed registration phase scheme.

User A sends his identifier \( U_A \) to \( S \) via secure channel. \( S \) retrieves \( U_A \) in the user registration table to check whether User A has already been registered. If User A exists in the user registration table, \( S \) chooses a random number \( N_{a_{\text{new}}} \), computes \( X_{a_{\text{new}}} = H(U_A || N_{a_{\text{new}}} || k_a) \) and stores \( \{ p, \alpha, P_a, X_{a_{\text{new}}}, T_a(\cdot), H(\cdot), E_{K(\cdot)}, D_{K(\cdot)} \} \) in \( SC_A \) and delivers it to User A via secure channel. User A stores a tuple \( \{ U_A, N_{a_{\text{new}}} \} \) into his user-register table.

User A, which receives \( SC_A \), computes \( U_A \) and selects password \( pw_A \) and biometric \( bm_A \). The \( SC_A \) that receives the user input and computes \( G_A = H(U_A || pw_A || h(bm_A)) \oplus X_{a_{\text{new}}}, F_A = H(U_A || pw_A || h(bm_A)) \oplus \) \( (bm_A)) || X_{a_{\text{new}}} \) and stores \( \{ p, \alpha, P_a, GX_A, GK_A, F_A, T_a(\cdot), H(\cdot), E_{K(\cdot)}, D_{K(\cdot)} \} \) in his memory.

### 5. Security analysis of the proposed scheme

In this section, we present an informal analysis and formal verification of the proposed scheme.

For formal analysis, we first use the BAN logic [49] to verify the mutual authentication property and the validation of the established session key of the proposed scheme, and we next use AVISPA (Automated validation of internet security protocol and application) toolkit [50] to verify the resistance of the proposed scheme against the passive and active attacks including man-in-the-middle and replay attacks.

Last, we demonstrate the proposed scheme can resist various kinds of attacks and provides various security properties through informal security analysis.

#### 5.1 Authentication proof based on BAN logic

**Notations and rules.** Table 3 shows some notations and rules defined in BAN logic [49].

**Goals.** We establish the following goals to prove that our scheme provides strong mutual authentication and the established session key is secure.

\[ \text{Goal} 1: S \equiv A \equiv U_A \]

| Notation | Description |
|----------|-------------|
| \( P \equiv X \) | \( P \) believes \( X \) |
| \( P \models X \) | \( P \) sees \( X \) |
| \( P \models \sim X \) | \( P \) once said \( X \) |
| \( P \models X \) | \( P \) has jurisdiction over \( X \) |
| \( \pi(X) \) | \( X \) is fresh |
| \( P \models \pi \rightarrow \pi Q \) | \( K \) 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 \( Y \) |
| \( \alpha \models \pi(X) \models \pi Y \) | Message-meaning rule |
| \( R_1 : \pi \models \pi(X) \) | Nonce-verification rule |
| \( R_2 : \pi(X) \models \pi X \) | Jurisdiction rule |
| \( R_3 : \pi \models \pi(X) \) | Freshness rule |
| \( R_4 : \pi \models \pi(X) \) | Belief rule |
| \( R_5 : \pi \models \pi(X) \) | See rule |
| \( R_6 : \pi \models \pi(X) \) | Hash function rule |

Table 3. Shows some notations and rules of BAN logic.
Goals: $S = B = U_B$

Goal: $A = S = R_S$

Goal: $B = S = R_S$

Goal: $A = B = U_B$

Goal: $B = U_A$

Goal: $A \equiv B \rightarrow B$

Goal: $B \equiv A \rightarrow B$

Goal: $A \equiv B \rightarrow A$

Goal: $B \equiv A \rightarrow A$

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

$M_{AS} : A \rightarrow S : \{ U_A, U_B', V_A = H(U_A || U_B') || T_A || A \xrightarrow{x_A} S \} \rightarrow S_{AS}, T_A$

$M_2 : S \rightarrow A : V_{SA} = H(N_S || U_A || U_B || T_A || A \xrightarrow{x_A} S) \rightarrow x_A, N_S$

$M_3 : A \rightarrow S : V_{AS} = H(U_A || U_B || T_A || N_S || A \xrightarrow{x_A} S) \rightarrow x_A, N_S$

$M_{BS} : B \rightarrow S : \{ U_B, N_S, V_B = H(U_B || T_B || N_S || B \xrightarrow{x_B} S) \} \rightarrow S_{BS}, T_B$

$M_{SB} : S \rightarrow B : \{ U_A, T_A, V_{SB} = H(U_A || T_A || B \xrightarrow{x_B} S) \} \rightarrow K_{BS}, T_A$

$M_{SA} : S \rightarrow A : \{ U_B, T_B, V_{SAB} = H(U_B || T_B || B \xrightarrow{x_B} S) \} \rightarrow K_{AS}$

$M_{CB} : B \rightarrow A : V_{BA} = H(U_B || U_A || T_A || T_B || R_S || K_{AB})$

$M_{CA} : A \rightarrow B : V_{AB} = H(U_B || U_A || T_A || T_B || R_S || K_{AB})$

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

$A_{A1} : A \equiv a$

$A_{A2} : A \equiv \#(a)$

$A_{A3} : A \equiv K_{AS}$

$A_{A4} : A \equiv x_A \rightarrow S$

$A_{A5} : A \equiv U_A$

$A_{A6} : A \equiv U_B$

$A_{A7} : A \equiv S \rightarrow T_B$

$A_{A8} : A \equiv S \rightarrow R_S$

$A_{A9} : B \equiv b$
$A_{B2}$: $B|\equiv \#(b)$

$A_{B3}$: $B|\equiv K_{BS}$

$A_{A4}$: $B|\equiv B^x_a \rightarrow S$

$A_{B5}$: $B|\equiv U_B$

$A_{B6}$: $B|\equiv S|\Rightarrow T_A$

$A_{B7}$: $B|\equiv S|\Rightarrow R_S$

$A_{A1}$: $S|\equiv N_S$

$A_{A2}$: $S|\equiv \#(N_S)$

$A_{A3}$: $S|\equiv A^x_a \rightarrow S$

$A_{A4}$: $S|\equiv B^x_b \rightarrow S$

**Analysis.** According to $M_{A2}$ and $A_{A4}$, we apply the Message-meaning rule ($R_1$) and Hash function rule ($R_3$), we can obtain:

$S| \equiv K_{A5}$, $S |\left\langle \{ U_A, U_B, V_A \rightarrow H(U_A || U_B || T_A || A^x_a \rightarrow S) > x_a \right\rangle_{K_{A5}}$

\[ S \equiv \left\langle \{ U_A, U_B, V_A \rightarrow H(U_A || U_B || T_A || A^x_a \rightarrow S) > x_a \right\rangle_{K_{A5}} \]

$S| \equiv A^x_a \rightarrow S$, $S |\left\langle < H(U_A || U_B || T_A || A^x_a \rightarrow S) > x_a \right\rangle_{A^x_a \rightarrow S}$

\[ S \equiv A^x_a \rightarrow S, S \equiv < H(U_A || U_B || T_A || A^x_a \rightarrow S) > x_a \right\rangle_{A^x_a \rightarrow S} \]

$A| \equiv A \sim H(N_S || U_A || U_B || T_A || A^x_a \rightarrow S)$

\[ A \equiv A \sim H(N_S || U_A || U_B || T_A || A^x_a \rightarrow S) \]

According to $M_{A2}$ and $A_{A4}$, we apply the Message-meaning rule ($R_1$) and Hash function rule ($R_3$), we can obtain:

$A| \equiv A^x_a \rightarrow S$, $A |\left\langle < H(N_S || U_A || U_B || T_A || A^x_a \rightarrow S) > x_a \right\rangle_{A^x_a \rightarrow S}$

\[ A \equiv A^x_a \rightarrow S, A \equiv < H(N_S || U_A || U_B || T_A || A^x_a \rightarrow S) > x_a \right\rangle_{A^x_a \rightarrow S} \]

$A| \equiv S| \sim H(N_S || U_A || U_B || T_A || A^x_a \rightarrow S)$

\[ A \equiv S| \sim H(N_S || U_A || U_B || T_A || A^x_a \rightarrow S) \]

According to $M_{A2}$, Freshness rule($R_4$) and $A_{A2}$, we can obtain:

$A| \equiv \#(a), T_A = T_a(a)$

\[ A| \equiv \#(T_a) \]

$A| \equiv \#(T_a)$

\[ A \equiv \#(T_a) \]

$A| \equiv \#(T_a)$

\[ A \equiv \#(T_a) \]

$A| \equiv \#(N_S, U_A, U_B, T_A)$

\[ A \equiv \#(N_S, U_A, U_B, T_A) \]
According to $S_2$ and $S_3$, we apply the Nonce-verification rule ($R_2$) and Belief rule($R_3$), we can obtain:

$$S_1: \quad \frac{A \equiv \#(N_5, U_\alpha, U_\beta, T_\alpha), A \equiv S \sim (N_5, U_\alpha, U_\beta, T_\alpha)}{A \equiv S \equiv (N_5, U_\alpha, U_\beta, T_\alpha)}$$

$$A \equiv S \equiv N_5$$

According to $M_3$, $A_{S3}$ and $S_1$, we apply the Message-meaning rule ($R_1$) and Hash function rule($R_6$), we can obtain:

$$S_5: \quad \frac{S \equiv A \leftarrow x_A S, S \Leftarrow < H(U_\alpha || U_\beta || T_\alpha || N_5 || A \leftarrow x_A S) > x_A S}{S \equiv A \sim H(U_\alpha || U_\beta || T_\alpha || N_5 || A \leftarrow x_A S), S \Leftarrow \{N_5, U_\alpha, U_\beta, T_\alpha\}}$$

$$S \equiv A \sim (N_5, U_\alpha, U_\beta, T_\alpha)$$

According to $M_3$, Freshness rule($R_4$) and $A_{S2}$, we can obtain:

$$S_6: \quad \frac{S \equiv \#(N_5)}{S \equiv \#H(N_5 || U_\alpha || U_\beta || T_\alpha || A \leftarrow x_A S)}$$

$$S \equiv \#H(N_5 || U_\alpha || U_\beta || T_\alpha || A \leftarrow x_A S)}$$

$$S \equiv \#(N_5, U_\alpha, U_\beta, T_\alpha)$$

According to $S_2$ and $S_6$, we apply the Nonce-verification rule ($R_2$) and Belief rule($R_3$), we can obtain:

$$S_2: \quad \frac{S \equiv \#(N_5, U_\alpha, U_\beta, T_\alpha), S \equiv A \sim (N_5, U_\alpha, U_\beta, T_\alpha)}{S \equiv A \equiv (N_5, U_\alpha, U_\beta, T_\alpha)}$$

$$: \text{Goal}_1$$

$$S \equiv A \equiv T_\alpha, S \equiv A \equiv U_A$$

According to $M_{BS}$ and $A_{S4}$, we apply the Message-meaning rule ($R_1$) and Hash function rule ($R_6$), we can obtain:

$$S_9: \quad \frac{S \Leftarrow K_{BS}, S \Leftarrow \{U_B, N_5, V_B = < H(U_B || T_B || N_5 || B \leftarrow x_B S) > x_B S \}_{K_{BS}}}{S \Leftarrow \{U_B, N_5, < H(U_B || T_B || N_5 || B \leftarrow x_B S) > x_B S \}_{K_{BS}}, S \Leftarrow T_B}$$

$$S \equiv B \leftarrow x_B S, S \Leftarrow < H(U_B || T_B || N_5 || B \leftarrow x_B S) > x_B S$$

$$S \equiv B \sim H(U_B || T_B || N_5 || B \leftarrow x_B S)$$

$$S \equiv B \sim H(U_B || T_B || N_5 || B \leftarrow x_B S), S \Leftarrow \{U_B, T_B, N_5\}$$

$$S \equiv B \sim (U_B, T_B, N_5)$$
According to $M_{BS}$, Freshness rule($R_4$) and $A_{S2}$, we can obtain:

$$S_9: \quad S| \equiv \#(N_1)$$

$$S| \equiv \#H(U_B||T_B||N_1||B^{x_9}S)$$

According to $S_9$ and $S_9$, we apply the Nonce-verification rule ($R_2$) and Belief rule($R_3$), we can obtain:

$$S_{10}: \quad S| \equiv \#(U_B, T_B, N_1), S| \equiv B| \equiv (U_B, T_B, N_1)$$

According to $S_1$ and $S_9$, we apply the Jurisdiction rule ($R_3$) and Hash function rule($R_8$), we can obtain:

$$B \iff K_{BS}, B \iff \{U_A, T_A, R_3, < H(U_A||U_B||T_A||T_B||R_3||B^{x_9}S) > \}_{x_9}$$

According to $M_{SB}$ and $A_{B4}$, we apply the Message-meaning rule ($R_1$) and Hash function rule($R_8$), we can obtain:

$$B| \equiv S| \sim H(U_A||U_B||T_A||T_B||R_3||B^{x_9}S), B \iff \{U_A, U_B, T_A, T_B, R_3\}, B| \equiv \#S| \sim (U_A, U_B, T_A, T_B, R_3)$$

According to $M_{SB}$, Freshness rule($R_4$) and $A_{R2}$, we can obtain:

$$S_{12}: \quad B| \equiv \#(b), T_B = T_b(x), B| \equiv \#(T_B)$$

$$B| \equiv \#H(U_A||U_B||T_A||T_B||R_3||B^{x_9}S)$$

According to $S_{11}$ and $S_{12}$, we apply the Nonce-verification rule ($R_2$) and Belief rule($R_3$), we can obtain:

$$S_{13}: \quad B| \equiv \#(U_A, U_B, T_A, T_B, R_3), B| \equiv S| \sim (U_A, U_B, T_A, T_B, R_3)$$

According to $S_{13}$ and $A_{B6}$, we apply the Jurisdiction rule ($R_3$), we can obtain:

$$S_{14}: \quad B| \equiv S| \Rightarrow U_A, B| \equiv S| \equiv U_A$$

: Goal$_i$
According to $S_{13}$, $A_{B'}$ and $A_{B''}$, we apply the Jurisdiction rule ($R_3$), we can obtain:

$$
S_{15} : \begin{align*}
B | \equiv S | \Rightarrow T_A, B | \equiv S | \Rightarrow T_A, B | \equiv S | \Rightarrow R_3, B | \equiv S | \Rightarrow R_3
\end{align*}
$$

According to $S_{15}$, $A_{B'}$ and $K_{AB} = H(T_{AB} \parallel R_3)$, we apply the Belief rule($R_3$), we can obtain:

$$
S_{16} : \begin{align*}
B | \equiv T_A, B | \equiv b, \quad B | \equiv T_{AB}, B | \equiv R_3, B | \equiv R_3 = \begin{array}{c} \text{Goal}_b \\ A | \equiv R_3 \end{array}
\end{align*}
$$

According to $M_{SA}$ and $A_{A''}$, we apply the Message-meaning rule ($R_4$) and Hash function rule($R_4$), we can obtain:

$$
S_{17} : \begin{align*}
A | \equiv K_{A''}, A | \equiv \{ U_B, T_B, R_3 \}, \quad < H(U_A \parallel U_B \parallel T_A \parallel T_B \parallel R_3 \parallel A \leftarrow S) ; X_A \leftarrow S >_{A_2} \}
\end{align*}
$$

According to $M_{SA}$, Freshness rule($R_4$) and $A_{A''}$, we can obtain:

$$
S_{18} : \begin{align*}
A | \equiv \#(a) , T_A = T_s(x), \quad A | \equiv \#(T_A), \quad A | \equiv \#(T_{A'')}, \quad A | \equiv \#H(U_A \parallel U_B \parallel T_A \parallel T_B \parallel R_3 \parallel A \leftarrow S)
\end{align*}
$$

According to $S_{17}$ and $S_{18}$, we apply the Nonce-verification rule ($R_2$) and Belief rule($R_3$), we can obtain:

$$
S_{19} : \begin{align*}
A | \equiv \#(U_A, U_B, T_A, T_B, R_3), A | \equiv \#(T_A), \quad A | \equiv \#(T_{A''}), \quad A | \equiv \#(U_A, U_B, T_A, T_B, R_3) ; \quad \text{Goal}_3
\end{align*}
$$

According to $S_{19}$, $A_{A''}$ and $A_{A'}$, we apply the Jurisdiction rule ($R_3$), we can obtain:

$$
S_{20} : \begin{align*}
A | \equiv S | \Rightarrow T_B, A | \equiv S | \Rightarrow T_B, A | \equiv S | \Rightarrow R_3, A | \equiv S | \Rightarrow R_3
\end{align*}
$$

According to $S_{20}$, $A_{A''}$ and $K_{AB} = H(T_{AB} \parallel R_3)$, we apply the Belief rule($R_3$), we can obtain:

$$
S_{21} : \begin{align*}
A | \equiv T_B, A | \equiv a, \quad A | \equiv T_{AB}, A | \equiv R_3, \quad A | \equiv R_3
\end{align*}
$$
According to $M_6$ and $S_{21}$, we apply the Message-meaning rule ($R_1$) and Hash function rule ($R_8$), we can obtain:

$$S_{22} : \begin{align*}
A & \equiv A^{K_{AB}}B, \ A \vdash H(U_A \parallel U_B \parallel T_A \parallel T_B \parallel R_S \parallel A^{K_{AB}}B) > A^{K_{AB}}B \\
A & \equiv B \vdash H(U_A \parallel U_B \parallel T_A \parallel T_B \parallel R_S \parallel A^{K_{AB}}B) \\
A & \equiv B \vdash H(U_A \parallel U_B \parallel T_A \parallel T_B \parallel R_S \parallel A^{K_{AB}}B), \ A \vdash \{U_A, U_B, T_A, T_B, R_S\} \\
A & \equiv B \vdash A^{K_{AB}}B
\end{align*}$$

According to $M_6$, Freshness rule($R_4$) and $A_{A2}$, we can obtain:

$$S_{23} : \begin{align*}
A & \equiv \#(a), T_A = T_a(x) \quad A \equiv \#(T_A) \quad A \equiv \#A^{K_{AB}}B \\
A & \equiv \#A^{K_{AB}}B
\end{align*}$$

According to $S_{22}$ and $S_{23}$, we apply the Nonce-verification rule ($R_2$), we can obtain:

$$S_{24} : \begin{align*}
A & \equiv \#A^{K_{AB}}B, A \equiv B \vdash A^{K_{AB}}B \\
A & \equiv B \equiv A^{K_{AB}}B
\end{align*}$$

Goal$_0$

According to $M_7$ and $S_{16}$, we apply the Message-meaning rule ($R_1$) and Hash function rule ($R_8$), we can obtain:

$$S_{25} : \begin{align*}
B & \equiv A^{K_{AB}}B, B \vdash H(U_B \parallel U_A \parallel T_B \parallel T_A \parallel R_S \parallel A^{K_{AB}}B) > A^{K_{AB}}B \\
B & \equiv A \vdash H(U_B \parallel U_A \parallel T_B \parallel T_A \parallel R_S \parallel A^{K_{AB}}B) \\
B & \equiv A \vdash H(U_B \parallel U_A \parallel T_B \parallel T_A \parallel R_S \parallel A^{K_{AB}}B), B \vdash \{U_A, U_B, T_A, T_B, R_S\} \\
B & \equiv A \vdash A^{K_{AB}}B
\end{align*}$$

According to $M_7$, Freshness rule($R_4$) and $A_{B2}$, we can obtain:

$$S_{26} : \begin{align*}
B & \equiv \#(b), T_B = T_b(x) \quad B \equiv \#(T_B) \quad B \equiv \#A^{K_{AB}}B \\
B & \equiv \#A^{K_{AB}}B
\end{align*}$$

According to $S_{25}$ and $S_{26}$, we apply the Nonce-verification rule ($R_2$), we can obtain:

$$S_{27} : \begin{align*}
B & \equiv \#A^{K_{AB}}B, B \equiv A \vdash A^{K_{AB}}B \\
B & \equiv A \equiv A^{K_{AB}}B \quad : \text{Goal$_1$}
\end{align*}$$

5.2 Validation test based on AVISPA

In this section, we simulate the proposed scheme for the formal security analysis using AVISPA. The AVISPA tool provides the role based HLPSL (High-Level Protocol Specification Language) for specification of protocols and security properties and four back-ends: OFMC (On-the-fly Model-Checker), CL-AtSe(Constraint-Logic-based Attack Searcher), SATMC (SAT-based ModelChecker) and TA4SP(Tree Automata-based Protocol Analyzer), which are used to identify active and inactive attacks on the protocol such as Man-In-The-Middle attack and replay attack, and to analyse various security properties of the protocol, such as key security and authentication [25, 50].
In order to verify the security properties of the protocol using AVISPA, it needs to be specified in HLPSL (High Level Protocol Specification Language).

**Specifying the proposed protocol.** There are three participants in the proposed protocol: server S and two users A, B. Figs 3–5 shows the specifications in HLPSL for the role of users A, B, and server S.

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

In our implementation, we verified the six secrecy goals containing the user anonymity and the user’s secret preserving and seven authentication properties for the mutual authentication.

**Analysis of the results.** We have simulated the proposed scheme using FMC and CL-AtSe back-ends of AVISPA. The simulation results of the security verification are 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 provides mutual authentication, 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 section, we demonstrate that the proposed scheme can resist various kinds of attacks and provides various security properties such as mutual authentication, user anonymity, untraceability and so on.

**Mutual authentication.** Mutual authentication is a key feature of the authenticated key agreement protocol. The proposed scheme achieves strong mutual authentication. In the proposed scheme, X is a shared secret between the server S and the user U in the registration phase. Also, Xs is a nonce of the server S, and a, b are secrets generated by user A and user B for generating a session key and these are also used as the nonce.

In the Step 5 of the authentication and key exchange phase, the server S receives the message M3 of user B, which includes Vb = f(Ua||Ta||Ns||Xa) generated by user B and the VAs = f(Ua||Ta||Ns||Xs) generated by user A. S also computes Vb = f(Ua||Ta||Ns||Xa) and authenticates the user A and B through verifying whether Vb = Vb and VAS = VAS. Since XA and XB are the secrets shared between S and the user A, B and NS is a nonce of S, Vb can be generated only by user B, and VAS can be generated only by user A. Thus, S can authenticate the user A and B through checking these values. In the Step 6, the user B receives the message M5 from S, which includes Vsb = f(Ua||Ta||Ns||Xb) generated by the server S. Since Tb = Td(α mod p) includes B’s nonce b and Xb is a secret shared with S, Vsb can be generated only by S. Therefore, B can authenticate S through checking Vsb. In the Step 7, the user A also can authenticate S through verifying Vsb = f(Ua||Ta||Ns||Xa). Likewise, User A authenticates User B through checking VBA = f(Ua||Ta||Ts||Rs||KA) at Stage 7, and User B authenticates User A through checking VAB = f(Ua|| Ta|| Ts|| Rs||KA) at Stage 8.

**User anonymity.** The proposed scheme guarantees user anonymity. The messages (MAs, MBs, Mbs, and Msa) associated with the user’s identifier are encrypted with the secret key, which is only known for each participant. For example, MAs is encrypted with the secret key KAs, which is calculated as follows: KAs = Td(Ta(x)) = Td(Ta(x)), where the random number a is only known for the user A and the secret key s is only known for the server S.
role alice(A,B,S :agent, XA, XB: symmetric_key, PS: public_key,
   H,T: hash_func, SND, RCV: channel(dy))

played_by A

def=

local State: nat,
   UA, UB, NA, NS, RS, TA, TB, TAB, KAS, Tx: text,
   VA, VSA, VAS, VSAB, VBA, VAB, KBA, KAB: text

const
   sec_xa, sec ua, sec_kab : protocol_id,
   auth_a_s_vsa, auth_a_s_vsab, auth_a_b_vba,
   auth_s_a_vas, auth_b_a_vab : protocol_id

init State := 0

transition

1. State = 0∧ RCV(start) =>
   State₁:= 1∧ secret({XA}, sec_xa, {A,S})
   ∧ secret({UA, sec ua, {A, B, S}})
   ∧ NA' := new() ∧ TA' := T(Tx.NA')
   ∧ KAS' := T(PS.TA') ∧ VA' := H(UA.UB.TA'.XA)
   % Send the first message to server
   ∧ SND ({UA.UB.VA'} _KAS'.TA')
   % Receive the reply message from server
   ∧ RCV(H(NS'.UA.UB.TA'.XA).NS') =>
   State₂:= 2∧ VSA' := H(NS'.UA.UB.TA'.XA)
   ∧ request(A, S, auth_a_s_vsa, VSA')
   ∧ VAS' := H(UA.UB.TA'.NS'.XA)
   % Send the authentication request message to bob
   ∧ SND(VAS'.NS')
   ∧ witness(A, S, auth_s_a_vas, VAS')
   % Receive the authentication reply message from bob
   ∧ RCV({UB.TB'.RS'.H(UA.UB.TA'.TB'.RS'.XA)}) _KAS'
       .H(UA.UB.TA'.TB'.RS'.KBA')) =>
   State₃:= 3∧ VSAB' := H(UA.UB.TA'.TB'.RS'.XA)
   ∧ request(A, S, auth_a_s_vsab, VSAB')
   ∧ TAB' := T(NA.TB') ∧ KAB' := H(TAB'.RS')
   ∧ VBA' := H(UA.UB.TA'.TB'.RS'.KAB')
   ∧ request(A, B, auth_a_b_vba, VBA')
   ∧ VAB' := H(UA.UB.TA'.TB'.RS'.KAB')
   % Send the session key exchange message to bob
   ∧ SND(VAB')
   ∧ witness(A, B, auth_b_a_vab, VAB')

end role

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

https://doi.org/10.1371/journal.pone.0273664.g003
role bob(A,B,S :agent, XA, XB: symmetric_key, PS: public_key, 
        H,T: hash_func, SND, RCV : channel(dy))
played_by B
def=
local   State: nat,
        UA, UB, NB, NS, RS, TA, TB, TAB, KAS, KBS, Tx: text,
        VB, VSB, VBA, VAB, KBA, KAB: text
const
        sec_xb, sec_ub, sec_kab : protocol_id,
        auth_b_s_vsb, auth_b_a_vab : protocol_id,
        auth_s_b_vb, auth_a_b_vba : protocol_id
init State := 0
transition
% Receive the authentication request message from alice
1. State = 0 ∧ RCV(H(UA,UB,TA'.NS'.XA).NS') →>
   State':= 1 ∧ secret({XB}, sec_xb, {B,S})
   ∧ secret({UB}, sec_ub, {A,B,S})
   ∧ NB':= new() ∧ TB':= T(Tx,NB')
   ∧ KBS':= T(PS,NB') ∧ VB':= H(UB, TB'.NS'.XB)
% Send the login request message to server
   ∧ SND({UB.NS'.VB'}_KBS'.TB'.H(UA,UB,TA'.NS'.XA))
   ∧ witness(B, S, auth_s_b_vb, VB)
% Receive the login reply message from server
2. State = 1 ∧ RCV({UA,TA'.RS'.H(UA,UB,TA'.TB'.RS'.XB)}_KBS'.
   {UB.TB'.RS'.H(UA,UB,TA'.TB'.RS'.XA)}_KAS') →>
   State':= 2 ∧ VSB':= H(UA,UB,TA'.TB'.RS'.XB)
   ∧ request(B, S, auth_b_s_vsb, VSB')
   ∧ TAB':= T(NB,TA') ∧ KBA':= H(TAB'.RS')
   ∧ VBA':= H(UA,UB,TA'.TB'.RS'.KBA')
% Send the authentication reply message to alice
   ∧ SND({UB.TB'.RS'.H(UA,UB,TA'.TB'.RS'.XA})_KAS'.VBA')
   ∧ witness(B, A, auth_a_b_vba, VBA)
% Receive the session key exchange message from alice
3. State = 2 ∧ RCV(H(UB,UA,TA'.TA'.RS'.KAB')) →>
   State':= 3 ∧ VAB':= H(UB,UA,TA'.TA'.RS'.KBA)
   ∧ secret({KBA}, sec_kab, {A, B})
   ∧ request(B, A, auth_b_a_vab, VAB')
end role

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

https://doi.org/10.1371/journal.pone.0273664.g004
role server(A,B,S :agent, XA, XB: symmetric_key, H,T: hash_func, SND, RCV : channel(dy))
played_by B

def=

local State: nat,
  UA, UB, NA, NB, NS, RS, Tx, KS: text,
  TA, TB, TAB, KAS, KBS, VSA, VAS, VB, VSB, VSAB: text

const
  sec_ks, sec_xa, sec_xb, sec_ua, sec_ub, sec_kab : protocol_id,
  auth_a_s_vsa, auth_s_a_vas, auth_s_b_vb,
  auth_a_s_vsab, auth_b_s_vsb : protocol_id

init State := 0

transition

% Receive the first message from alice
1. State = 0 ∧ RCV({UA,UB,H(UA,UB,TA'.XA)})_T(KS.TA').TA') =|=>
   State':= 1 ∧ secret({KS}, sec_ks, {S})
   ∧ secret({XA}, sec_xa, {A,S})
   ∧ secret({XB}, sec_xb, {B,S})
   ∧ secret({UA}, sec_ua, {A,B,S})
   ∧ secret({UB}, sec_ub, {A,B,S})
   ∧ NS':= new() ∧ VSA':= H(NS'.UA,UB,TA'.XA)

% Send the first reply message to alice
   ∧ SND(VSA'.NS') ∧ witness(S, A, auth_a_s_vsa, VSA')
% Receive the login messages from bob
2. State = 1 ∧ RCV({UB,NS'.H(UB,TB'.NS'.XB)})_T(KS,TB')
   .TB'.H(UA,UB,TA'.NS'.XA)) =|=>
   State':= 2 ∧ VAS':= H(UA,UB,TA'.NS'.XA)
   ∧ request(S, A, auth_s_a_vas, VAS')
   ∧ VB':= H(UB, TB'.NS'.XB)
   ∧ request(S, B, auth_s_b_vb, VB') ∧ RS':= new()
   ∧ VSB':= H(UA,UB,TA'.TB'.RS'.XB)
   ∧ VSAB':= H(UA,UB,TA'.TB'.RS'.XA)

% Send the login reply message to bob
   ∧ SND({UA,TA'.RS'.VSB'})_T(KS,TB')
       .{UB, TB'.RS'.VSAB'}_T(KS.TA'))
   ∧ witness(S, A, auth_a_s_vsab, VSAB')
   ∧ witness(S, B, auth_b_s_vsb, VSB')

end role

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

https://doi.org/10.1371/journal.pone.0273664.g005
% ======================= session =======================
role session(A,B,S:agent, XA, XB:symmetric_key, PS:public_key,
                 H,T:hash_func)
def=
local S1, S2, S3, R1, R2, R3: channel(dy)
composition
    alice(A, B, S, XA, XB, PS, H, T, S1, R1)
    \ bob(A, B, S, XB, XB, PS, H, T, S2, R2)
    \ server(A, B, S, XA, XB, H, T, S3, R3)
end role

% ======================= environment =======================
role environment()
def=
const a, b, s: agent, xa, xb, xi: symmetric_key, ps: public_key,
                 h: hash_func, t: hash_func,
                 sec ua, sec ub, sec xa, sec xb, sec kab, sec ks: protocol_id,
                 auth a s vsa, auth a s vsab, auth a b vba,
                 auth b s vsb, auth b a vab,
                 auth s a vas, auth s b vb: protocol_id
intruder_knowledge = {a, b, s, xi, h, t, ps}
composition
    session(a, b, s, xa, xb, ps, h, t)
    \ session(a, i, s, xa, xi, ps, h, t)
    \ session(i, b, s, xi, xb, ps, h, t)
    \ session(a, b, i, xa, xb, ps, h, t)
end role

% ======================= goal =======================
goal
secrecy_of sec ua
secrecy_of sec ub
secrecy_of sec xa
secrecy_of sec xb
secrecy_of sec kab
secrecy_of sec ks
authentication_on auth a s vsa
authentication_on auth a s vsab
authentication_on auth a b vba
authentication_on auth b s vsb
authentication_on auth b a vab
authentication_on auth s a vas
authentication_on auth s b vb
end goal
environment()

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

https://doi.org/10.1371/journal.pone.0273664.g006
Even if $T_a(x)$ and $T_s(x)$ is exposed, according to CDLP and CDHP assumptions, it is impossible for a third party to calculate $K_{AS}$ or $a, s$. Therefore, a third party except user and server cannot know the user's identifier.

**Untraceability.** The proposed protocol provides untraceability.

As in the user anonymity proof, all messages ($M_{AS}, M_{BS}, M_{SA}$ and $M_{SB}$) containing the user's identity are encrypted as follows.

$$M_{AS} = E_{K_{AS}}(U_A, U_B, V_A), \quad M_{SA} = E_{K_{AS}}(U_B, T_B, R_S, V_{SAB}), \quad K_{AS} = T_{ks}(T_A) = T_{ks}(\alpha) \mod p$$

$$M_{BS} = E_{K_{BS}}(U_B, N_S, V_B), \quad M_{SB} = E_{K_{BS}}(U_A, T_A, R_S, V_{SB}), \quad K_{BS} = T_{kb}(T_B) = T_{bks}(\alpha) \mod p$$

---

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

https://doi.org/10.1371/journal.pone.0273664.g007
Then, the encryption keys $K_{AS}$ and $K_{BS}$ are all computed from the random numbers $a$ and $b$ generated by user $A$ and $B$, so that different messages are exchanged in different sessions. Other messages also contain a random number in different sessions, so that they are presented random bit arrays in each sessions.

Therefore, the proposed protocol provides untraceability.
Off-line password guessing attack

The proposed scheme is secure against the password guessing attack.

In the proposed scheme, user’s password is used for accessing the smart card and the information related to it does not disclose in public channel.

The information stored in the user A’s smart card is \( \{ p, \alpha, P, G, F, T, H, E, D \} \), and the information related to the user’s password is \( G_A = H(U_A||p|w_A|h(bm_A)) \) and \( F_A = H(U_A||pw_A||h(bma_A)||X_A) \). Suppose that an attacker steals user A’s smart card \( SC_A \) and knows his identifier \( U_A \). In order to guess the user A’s password, the attacker must compute \( PR_A^* = H(U_A||pw_A^*||h(bma_A)) \), \( X_A^* = G_A \oplus PR_A^* \) and \( F_A^* = H(U_A||pw_A^*||h(bma_A)||X_A^*) \) with any password \( pw_A^* \) to compare \( F_A^* \) and \( F_A \) stored in \( SC_A \). However, as the attacker cannot know \( h(bm) \), he cannot compute \( PR_A^* \). Therefore, the attacker cannot guess the user’s password.

**Privileged insider attack.** The proposed scheme is secure against the privileged-insider attack. In the proposed scheme, user’s password is not transmitted to the server \( S \) and 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 registration phase of the proposed scheme, the server stores a tuple \( \{ U_A, N_a \} \) into his user register-table, where \( U_A \) is user A’s identifier and \( N_a \) is a random number selected by the server. These are not sensitive to authenticate the user. Therefore, the proposed scheme is secure against stolen verifier attack.

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

The user impersonate attack is only possible in the scheme which can’t provide a certain authentication. For example, if a participant \( X \) can’t authenticate a participant \( Y \), an attacker can impersonate as \( Y \). As shows the above, the proposed scheme achieves certain mutual authentication. In the Step 5, the server \( S \) certainly authenticates \( A \) and \( B \) with his nonce \( N_t \) and the user’s secret \( X_A \) and \( X_B \). If an attacker wants to impersonate as the user A, he must compute \( V_A = H(U_A||U_B||T_A||X_A) \) or \( V_{AS} = H(U_A||U_B||T_A||N_a||X_A) \), but he doesn’t know the user A’s secret \( X_A = H(U_A||N_a||k) \) and could not compute it (Because \( k \) is the server’s secret key), so he cannot compute \( V_A \) or \( V_{AS} \) and cannot impersonate as the user A. As the same, an attacker cannot impersonate as the user B.

**Man-in-the-middle attack.** As shows the above, the proposed scheme achieves certain mutual authentication, so an attacker cannot impersonate as the initiator A and the responder B, and cannot achieve the man-in-the-middle attack.

If an attacker wants to achieve a man-in-the-middle attack, he must exchange a session key \( K_{AB}^* = H(T_{AB}||R_{S}^*) \) with users A and B.

Suppose that an attacker generates a random number \( b^* \) and \( R_{S}^* \) to exchange the session key with user A, computes \( T_{AB}^* = T_b|T_d|a \) (\( a \) mod \( p \)), and then computes \( K_{AB}^* = H(T_{AB}^*||R_{S}^*) \). However, the attacker cannot generate \( V_{SAB} = H(U_A||U_B||T_A||T_B||R_{S}^*||X_A) \) because he does not know the user A’s secret \( X_A \). Therefore, in Step 7, user A can detect the attack via checking for \( V_{SAB} \).

To exchange the session key with user B, an attacker has to compute \( K_{AB}^* \) by generating random numbers \( a^* \) and \( R_{S}^* \) and calculate \( V_{SB} = H(U_A||U_B||T_A||T_B||R_{S}^*||X_B) \). However, since the attacker does not know the user B’s secret \( X_B \), it is impossible to compute the \( V_{SB} \), so in Step 6, user B can detect the attack via checking for \( V_{SB} \).

That is, the proposed scheme is resistant to man-in-the-middle attack.
Replay attack. In the Step 5 of the proposed scheme, the server S certainly authenticates A and B.

If an attacker C intercepts the previous message $M_1 = \{M_{AS}^+, T_A^+\}$ of the user A and retransmits it to the server, the server responses $M_2 = \{V_{SA}, N_3\}$, where $N_3$ is generated by the server. However, in the Step 3, the attacker must compute $M_3 = \{V_{AS}, N_3\}$, but he does not know the user A’s secret $X_A$ and he cannot compute $V_{AS} = H(U_A||U_B||T_A||N_3||X_A)$. Therefore, in the Step 5, the server S can successfully detect the replay attack through checking for $V_{AS}$. Likewise, if an attacker retransmits the user B’s message $M_4$ to server S, in the Step 5, the server can detect the replay attack through checking for $V_B = H(U_B^+||T_B^+||N_S^+||X_B)$ containing the user B’s secret $X_B$.

Known key security. In the proposed scheme, the session key $K_{AB}$ is calculated as $K_{AB} = H(T_{ab}(a) \mod p || R_3)$. It contains the random numbers $a$, $b$ and $R_3$ that are generated by each participant for each session. Even if an attacker knows the previous session key, he cannot calculate a new session key.

6. Performance comparisons

This section, we evaluate the computational cost, communication overhead and security performance of our proposed scheme and other recent 3PAKE schemes [19, 22, 23, 61]. For comparison of computational cost, we define some notations as follows.

$t_c$: time needed for Chebyshev polynomial operation
$t_h$: time needed for one-way hash function operation
$t_m$: time needed for a modular exponential operation
$t_q$: time needed for a modular squaring operation
$t_{sq}$: time needed for a square root modulo N operation
$t_s$: time needed for symmetric encryption/decryption operation

Table 4 shows the comparison of the computational cost of our proposed scheme and other 3PAKE schemes. As shown Table 4, the computational cost of our proposed scheme is lower than Jabbari et al.’s scheme, Li et al.’s scheme and Lu et al.’s scheme.

In order to presume the communication overhead of our proposed scheme, we consider the bit size of identity, random number $|N|$, timestamp, hash(SHA-1) output and Chebyshev chaotic maps as $|ID| = 160$, $|N| = 160$, $|Ts| = 32$, $|H| = 160$, $|CM| = 160$ bits, respectively. Furthermore, the bit sizes used for modular exponentiation and modular square operations are considered as $|ME| = 1024$ bits [25]. Table 5 shows the communication overhead of our proposed scheme according to above assumption.

Table 4. Comparison of the computational cost between the proposed scheme and other 3PAKE schemes.

|        | Irshad et al. [61] | Jabbari et al. [19] | Li et al. [22] | Lu et al. [23] | proposed |
|--------|-------------------|--------------------|---------------|---------------|----------|
| A      | $3t_c + 10t_h$    | $4t_c + 2t_h + 4t_h$ | $4t_c + 1t_h + 5t_h$ | $3t_c + 4t_h$ | $3t_c + 2t_h + 9t_h$ |
| B      | $3t_c + 1t_h + 10t_h$ | $3t_c + 2t_h + 4t_h$ | $4t_c + 2t_h + 5t_h$ | $2t_c + 3t_h + 5t_h$ | $3t_c + 2t_h + 7t_h$ |
| S      | $2t_c + 1t_h + 10t_h$ | $4t_c + 4t_h$ | $4t_c + 2t_h + 5t_h$ | $5t_c + 5t_h + 7t_h$ | $2t_c + 4t_h + 8t_h$ |
| Total  | $8t_c + 2t_h + 30t_h$ | $11t_c + 8t_h + 12t_h$ | $12t_c + 2t_m + 2t_q + 15t_h$ | $10t_c + 12t_h + 16t_h$ | $8t_c + 8t_h + 24t_h$ |
| Round  | 5                 | 5                  | 6             | 6             | 7        |

https://doi.org/10.1371/journal.pone.0273664.1004
Table 6 shows the comparison of the communication overhead of our proposed scheme and other 3PAKE schemes. As shown Table 6, our proposed scheme has many message rounds and its communication overhead is higher than other schemes.

Table 7 shows the comparative evaluation of the security function between the proposed scheme and other 3PAKE schemes.

As shown in Table 7, the proposed scheme outperforms the other schemes in terms of the security functions presented.

Irshad’s scheme has lower communication overhead and computational cost than the proposed scheme, but it does not provide untraceability and re-registration phase.

Jabbari’s scheme has higher communication overhead and more expensive computational costs than ours and it does not provide re-registration phase.

Li’s scheme has lower communication overhead than ours, but it has more expensive computational costs and his scheme attempted to provide user anonymity, but did not achieve
it. His scheme is also vulnerable to the verifier disclose attack, user impersonate attack and stolen verifier attack and it has faults in password change phase.

Lu’s scheme has lower communication overhead than ours, but it has more expensive computational costs and his scheme does not provide user anonymity, untraceability and re-registration phase.

7. Conclusion and future work

In this paper, we have analysed the Li et al.’s scheme and proved that his scheme has some faults, and proposed an enhanced three-party mutual authentication key exchange (3PMAKE) protocol based on chaotic maps using smart card to provide with user anonymity and untraceability in the environment for user-to-user communication. The proposed scheme provides strong mutual authentication between servers and users without using timestamp, can be re-registered to the system without changing the user’s identifier. The proposed scheme also provides anonymity and untraceability and is secure against several attacks such as user impersonate attacks, privileged insider attacks, stolen verifier attacks. In addition, we have formally analysed the security properties of proposed scheme and verified their validity based on BAN logic and AVISPA tool, and proved that the proposed scheme is secure against various attacks through informal security analysis.

The proposed scheme is designed to provide strong mutual authentication between communication participants without a timestamp, so the number of message exchanges and communication overhead are relatively high. In addition, since key exchange is performed based on chaotic-maps, the security performance of the proposed scheme is enhanced, but has the limitation of increasing computational cost compared to lightweight schemes that do not use public key encryption. The proposed method is suitable for systems that have to provide stronger security properties in environments where timestamp is not available and there is no restriction on communication overhead.

In the future, we will investigate more improved authentication key exchanges in IoT or WSN environments that requires lightweight scheme in terms of communication overhead or computational cost. That is, instead of public key encryption, we only use hash functions to reduce the computational cost of key exchange and reduce the communication overhead.

Supporting information

S1 Protocol.

(DOCX)

Author Contributions

Conceptualization: Kyong-Sok Pak.

Data curation: Song-Ho Pak.

Formal analysis: Song-Ho Pak.

Investigation: Chol-Man Ho.

Methodology: Kyong-Sok Pak.

Resources: Mi-Hyang Kim.

Software: Mi-Hyang Kim.

Validation: Chol-Man Ho.
References

1. 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

2. 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):4166–4181. https://doi.org/10.1002/sec.1596

3. 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

4. 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

5. Odelu V, Saha S, Prasath R, Sadineni L, Conti M, Jo M. Efficient privacy preserving device authentication in WBANs for industrial e-health applications. COMPUTERS & SECURITY. 2019; 83:300–312. https://doi.org/10.1016/j.cose.2019.03.002

6. Tallapally S. Security enhancement on simple three party PAKE protocol. Inform Technol Control. 2012; 41(1):15–22.

7. Farash MS, Attari MA. An enhanced and secure three-party password-based authenticated key exchange protocol without using server’s public-keys and symmetric cryptosystems. Inform Technol Control. 2014; 43(2):143–150.

8. Lu Y, Peng H, Yang Y et al. A three-party password-based authenticated key exchange protocol for wireless communications. Inform Technol Control. 2015; 44(4):404–409.

9. Chen CM, Wang KH, Yeh KH, Xiang B, Wu TY. Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications. Journal of Ambient Intelligence and Humanized Computing. 2018; 10(8):3133–3142. https://doi.org/10.1007/s12652-018-1029-3

10. Youn TY, Kang ES, Lee C. Efficient three-party key exchange protocols with round efficiency. Telecommunication Systems. 2013; 52(2):1367–1376. https://doi.org/10.1007/s11235-011-9649-3

11. 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):337–356. https://doi.org/10.1007/s11277-015-3123-6

12. Lin TH, Lee TF. Secure verifier-based three-party authentication schemes without server public keys for data exchange in telecare medicine information systems. Journal of Medical Systems. 2014; 38(5):30–38.

13. Chiou SY, Lin CH. An Efficient Three-Party Authentication Scheme for Data Exchange in Medical Environ. Secur Commun Netw. 2016; 2018:1–15. https://doi.org/10.1155/2016/9146297

14. Wu F, Xu L, Kumari S, Li X, Khan MK, Das AK. An enhanced mutual authentication and key agreement scheme for mobile user roaming service in global mobility networks. Ann. Telecommun. 2017; 72 (3):131–144.

15. Gupta M, Chaudhri NS. Anonymous two factor authentication protocol for roaming service in global mobility network with security beyond traditional limit. Ad Hoc Networks. 2019; 84:56–67. https://doi.org/10.1016/jadhoc.2018.07.030

16. Xie Q, Zhao J, Yu X. Chaotic maps-based three-party password-authenticated key agreement scheme. Nonlinear Dyn. 2013; 74(4):1021–1027. https://doi.org/10.1007/s11071-013-1020-7

17. 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. 2014; 79(4):2485–2495. https://doi.org/10.1007/s11071-014-1827-x

18. Xie Q, Hu B, Chen KF, Liu W H, Tan X. Chaotic maps and biometrics-based anonymous three-party authenticated key exchange protocol without using passwords. Chin. Phys. B. 2015; 24(11):1–8. Article No. 110505.

19. Jabbari A, Mohasefi J B. Improvement in new three-party-authenticated key agreement scheme based on chaotic maps without password table. Nonlinear Dyn. 2019; 95:3177–3191. https://doi.org/10.1007/s11071-018-0474-y

20. 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(7):399–411. https://doi.org/10.1007/s11071-014-1304-6
21. 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. 2014; 79(4):2345–2358. https://doi.org/10.1007/s11071-014-1816-0

22. 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. 2018; 22:2495–2506. https://doi.org/10.1007/s00500-017-2504-z

23. 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

24. Pak KS, Pak SH, Ho CM, Pak MS, Hwang CJ. Anonymity preserving and round effective three-party authentication key exchange protocol based on chaotic maps. Plos One. 2019; 14(3):e0213976. https://doi.org/10.1371/journal.pone.0213976 PMID: 30893354

25. Jangirala S, Das K, Wazid M, Kumar N. Anonymous Lightweight Chaotic Map-Based Authenticated Key Agreement Protocol for Industrial Internet of Things. IEEE Transactions on Dependable and Secure Computing. 2020; 17(6):1133–1146. https://doi.org/10.1109/TDSC.2018.2857811

26. Xue K, Hong P, Ma C. A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. J. Comput. System Sci. 2014; 80(1):195–206.

27. Li X, Xiong Y, Ma J, Wang W. An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. Journal of Network and Computer Applications. 2012; 35(2):763–769.

28. Gupta PC, Dhar J. Hash based multi-server key exchange protocol using smart card. Wireless Pers Commun. 2016; 87(1):225–244.

29. Amin R, Kumar N, Biswas GP, Iqbal R, Chang V. A light weight authentication protocol for IoT-enabled devices in distributed Cloud computing environment. Future Generation Computer Systems. 2018; 78:1005–1019. https://doi.org/10.1016/j.future.2016.12.028

30. Tomar A, Dhar J. An ECC Based Secure Authentication and Key Exchange Scheme in Multi-server Environment. Wireless Pers Commun. 2019; 107(1):351–372. https://doi.org/10.1007/s11277-019-06280-7

31. Zhou J, Ma M, Sun S. A Hybrid Authentication Protocol for LTE/LTE-A Network. IEEE Access. 2019; 7:28319–28333.

32. Challa S, Wazid M, Das AK, Kumar N, Reddy AG, Yoon EJ, et al. Secure signature-based authenticated key establishment scheme for future iot applications. IEEE Access. 2017; 5:3028–3043.

33. Jia X, He D, Li L, Choo KKR. Signature-based three-factor authenticated key exchange for internet of things applications. Multimed Tools Appl. 2018; 2017(2):1–28. https://doi.org/10.1007/s11042-017-5560-6

34. Jia X, He D, Kumar N, Choo KKR. Authenticated key agreement scheme for fog-driven IoT healthcare system. Wireless Networks. 2018; 25(8):4737–4750. https://doi.org/10.1007/s11276-018-1759-3

35. Ma M, He D, Wang H, Kumar N, Choo KKR. An Efficient and Provably-Secure Authenticated Key Agreement Protocol for Fog-Based Vehicular Ad-Hoc Networks. IEEE Internet of Things Journal. 2019; 6(5):8056–8075. https://doi.org/10.1109/JIOT.2019.2902840

36. Reddicherla VR, Rawat U, Garg K. Securing NEMO Using a Bilinear Pairing-Based 3-Party Key Exchange (3PKE-NEMO) in Heterogeneous Networks. Foundations of Science. 2019; 25(4):1125–1146. https://doi.org/10.1007/s10699-019-09591-x

37. Chuang MC, Chen MC. An anonymous multi-server authenticated key agreement scheme based on trust computing using smartcards and biometrics. Expert Syst. Appl. 2014; 41(4):1411–1418. https://doi.org/10.1016/j.eswa.2013.08.040

38. Wei F, Zhang R. A Provably Secure Anonymous Two-Factor Authenticated Key Exchange Protocol for Cloud Computing. Fundamenta Informaticae. 2018; 157:201–220. https://doi.org/10.3233/FI-2018-1626

39. Yang Z, Lai J, Sun Y, Zhou J. A NovelAuthenticated Key Agreement Protocol With Dynamic Credential for WSNs. ACM Transactions on Sensor Networks. 2019; 15(2):1–27. Article No. 22. https://doi.org/10.1145/3303704

40. Marko K, Islam SH, and Marko H. A Robust and Efficient Mutual Authentication and Key Agreement Scheme with Untraceability for WBANs. Comput. Networks. 2019; 148:196–213. https://doi.org/10.1016/j.comnet.2018.11.016

41. Zhou L, Li X, Yeh KH, Su C, Chiu W. Lightweight IoT-based authentication scheme in cloud computing circumstance. Future Generation Computer Systems. 2019; 91:244–251. https://doi.org/10.1016/j.future.2018.08.038
42. Mehra PS, Doja MN, Alam B. Codeword Authenticated Key Exchange (CAKE) light weight secure rout-
ing protocol for WSN. Int J Commun Syst. 2019; 32:1–27. Article No. e3879. https://doi.org/10.1002/
dac.3879

43. Li X, Ibrahim MH, Kumar S, Sangaih AK, Gupta V, Choo KKR. Mutual authentication and key
agreement scheme for wearable sensors in wireless body area networks. Comput. Networks. 2017;
129:429–443. https://doi.org/10.1016/j.comnet.2017.03.013

44. Khan MK, Alghathar K. Cryptanalysis and Security Improvements of ‘Two-Factor User Authentica-
tion in Wireless Sensor Networks’. Sensors. 2010; 10:2450–2459. https://doi.org/10.3390/s100302450
PMID: 22294935

45. Diffie W, Hellman M. New directions in cryptography. IEEE Trans. Inf. Theory. 1976; 22(6):644–654.

46. Koblitz N. Elliptic curve cryptosystems. Math Comput. 1987; 48(177):203–209.

47. Mason JC, Handscomb DC. C книжные полиномы. London: Chapman & Hall/CRC Press; 2003.

48. Zhang L. Cryptanalysis of the public key encryption based on multiple chaotic systems. Chaos Solit-
ons Fract. 2012; 37(3):669–674. https://doi.org/10.1016/j.chaos.2006.09.047

49. Burrows M, Abadi M, Needham R. A logic of authentication. Acm Sigops Operating Systems Review.
1990; 8(1):18–36.

50. AVISPA: Automated Validation of Internet Security Protocols and Applications. http://www.avispa-
project.org/(accessed on January 2019).

51. Jin ATB, Ling DNC, Goh A. Biohashing: two factor authentication featuring fingerprint data and toke-
nised random number. Pattern Recogn. 2004; 37(11):2245–2255.

52. Lumini A, Nanni L. An improved BioHashing for human authentication. Pattern Recogn. 2007; 40
(3):1057–1065.

53. Dodis Y, Kanukurthi B, Katz J, Reyzin L, Smith A. Robust Fuzzy Extractors and Authenticated Key
Agreement From Close Secrets. IEEE Trans. Inf. Theory. 2008; 58(9):6207–6222. https://doi.org/10.
1109/TIT.2012.2200290

54. Dolev D, Yao A, On the security of public key protocols, IEEE Trans. Inform. Theory. 1983; 29 (2)
(1983): 198–208.

55. Veyrat-Charvillon, N.; Standaert, F.X. Generic side-channel distinguishers: Improvements and limita-
tions. In Proceedings of the Annual Cryptology Conference, Santa Barbara, CA, USA, 14–18 August
2011; Springer: Berlin/Heidelberg, Germany, 2011; 354–372.

56. Bellare M, Pointcheval D, Rogaway P. Authenticated key exchange secure against dictionary attacks.
Tecnologia Electronica E Informatica 1. 2000; 807:139–155.

57. Kocher P, Jaffe J, Jun B, Rohatgi P. Introduction to differential power analysis. Journal of Cryptographic
Engineering. 2011; 1(1): 5–27. https://doi.org/10.1007/s13389-011-0006-y

58. Messerges TS, Dabbish EA, Sloan RH. Examining smart-card security under the threat of power analy-
sis attacks. IEEE Trans Comput. 2002; 51(5):541–552. https://doi.org/10.1109/TC.2002.1004593

59. Horng SJ, Tzeng SF, Pan Y, Fan P, Wang X, Li T, et al. b-SPEC S+: Batch Verification for Secure Pseu-
donymous Authentication in VANET. IEEE Transactions on Information Forensics and Security. 2013;
8(11):1860–1875. https://doi.org/10.1109/TIFS.2013.2277471

60. Tzeng SF, Horng SJ, Li T, Wang X, Huang PH, Khan MK. Enhancing Security and Privacy for Identity-
based Batch Verification Scheme in VANET. IEEE Transactions on Vehicular Technology. 2017; 66
(4):3235–3248. https://doi.org/10.1109/TVT.2015.2406877

61. Irfahd A, Sher M, Chaudhry S, Xie Q, Kumari S, Wu F. An improved and secure chaotic map based
authenticated key agreement in multi-server architecture. Multimed. Tools Appl. 2017; 77(01): 1–38.