Cryptanalysis and Improvements on Three-party-authenticated Key Agreement Protocols Based on Chaotic Maps

Chien-Ming Chen, Linlin Xu, King-Hang Wang, Shuai Liu, Tsu-Yang Wu

1 Harbin Institute of Technology Shenzhen Graduate School, China
2 Hong Kong University of Science and Technology, Hong Kong
3 College of Computer Science and Engineering, Shandong University of Science and Technology, China
4 Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, China
5 National Demonstration Center for Experimental Electronic Information and Electrical Technology Education, Fujian University of Technology, China

chienming@hit.edu.cn, 980742703@qq.com, kevinw@cse.ust.hk, 362575680@qq.com, wutsuyang@gmail.com

Abstract

Chaotic map has been receiving more and more attention in the cryptographic literature. In recent years, there are some scholars working on a particular type of authenticated key exchange protocol using chaotic map. Some of them identified a vulnerability of their precedences and presented their patching protocols. In this paper, we also identify a vulnerability in some of these authenticated key exchange protocols. We first redesign a protocol and optimize it in computation efficiency. Then we present the security analysis by a logic proof based on BAN logic.

Keywords: Chaotic map, Cryptography, Authentication key exchange protocol

1 Introduction

Authenticated Key Exchange (AKE) [1-6] is an important cryptographic tool to establish a confidential channel between two or more entities over a public network. Usually the entities share a common secret like a password or a secret key. During an AKE protocol, each participating entity would be required to response to a cryptographic challenge with his knowledge on the common secret in order to prove his identity. Meanwhile, as the protocol is executed over a public network, it is assumed there are adversaries could eavesdrop and inject messages in the channel in order to either impersonate one of the participating entities or to learn the session key established for the confidential channel.

A particular type of AKE is called three-party AKE (or 3AKE) [7-9]. In 3AKE, there are normally one server participant shares secrets with each of two client participants. The protocol again assumes the existence of adversaries to establish a confidential channel for the two clients to communicate. Despite the long history 3AKE in the literature, to our best knowledge, 3AKE on chaos can only be traced to the year 2012. Lai et al. [10] proposed a password-based 3AKE protocol using an enhanced Chebyshev chaotic map [11] in 2012. However, Zhao et al. [12] found that this protocol is vulnerable to a privileged insider attack and an off-line password guessing attack, and they proposed their improved protocol. In 2013, Lee et al. [13] and Xie et al. [14] proposed their chaos-based 3AKE protocols. Hu et al. [15] further showed that Lee et al.’s [13] protocol is vulnerable to a man-in-the-middle attack and a user anonymity attack. In 2015, Lee et al. [16] showed that Xie et al.’s protocol [14] also suffers from an on-line password guessing attack. They also described a new chaotic based 3AKE protocol. Besides, Li et al. [17] proposed another chaotic based 3AKE protocol to improve the existing protocols. Very recently, Chen et al. [18] showed that Li et al.’s protocol [13] is vulnerable to a user impersonation attack.

Unfortunately, in this paper we find that Lee et al.’s protocol [16], Li et al.’s protocol [17] and Xie et al.’s protocol [14] are insecure against a message replay attack. Under this attack, a malicious attacker can impersonate two honest client entities to establish a confidential session with the server without holding the clients’ secrets. We then propose a new 3AKE protocols based on chaotic maps. According the performance analysis, our protocol has better efficiency compared with previous protocols. We also present the security analysis by a logic proof based on BAN logic.

2 Chebyshev Chaotic Maps

Owing to its randomness feature, the family of...
chaotic maps receives more and more attention in the information security literatures [19-24] in the recent years. These high quality works demonstrate how to create secure cryptographic tools with using chaos as a primitive.

In this section, we introduce some basic concepts and preliminaries of the chaotic maps. We select the extended Chebyshev’s polynomial [11]

\[ T_n(x) = \begin{cases} 1, & \text{if } n = 0 \\ x, & \text{if } n = 1 \\ 2xT_{n-1}(x) - T_{n-2}(x), & \text{if } n \geq 2 \end{cases} \]

which developed from Chebyshev’s polynomial [25] as an instantiation of the chaotic map.

By the above recursive approach, we can obtain some examples of the Chebyshev’s polynomial:

\[ T_0(x) = 1, \quad T_1(x) = x, \quad T_2(x) = 2x^2 - 1. \]
\[ T_3(x) = 2xT_2(x) - T_1(x) = 4x^3 - 3x. \]
\[ T_4(x) = 2xT_3(x) - T_2(x) = 8x^4 - 8x^2 + 1. \]

It is easy to see that \( T_n(x) \) is a polynomial of degree \( n \). If the variable \( x \in [-1, 1] \), then we have \( T_n(x) \in [-1, 1] \). Hence, we can define a special case of Chebyshev polynomial \( T_n(x) : [-1, 1] \to [-1, 1] \) by \( T_n(x) = \cos(n \arccos(x)) \).

For \( n \geq 2 \), the Chebyshev’s polynomial \( T_n(x) \) satisfies the following two properties:

1. The semi group property:
\[
T_n(T_a(x)) = \cos(a \arccos(\cos(b \arccos(c(x)))) = \cos(ab \arccos(c(x))) = T_{ab}(x) = T_{ab}(T_a(x))
\]

for any positive integers \( a, b \) and \( x \in [-1, 1] \).

2. The chaotic property. \( T_n(x) \) is a prototype of a chaotic map. It has a unique absolutely continuous invariant measure \( \mu(x) = \frac{1}{\pi \sqrt{1-x^2}} \) with positive Lyapunov exponent \( \lambda = \ln n \).

An enhanced Chebyshev’s polynomial is defined on \( (-\infty, \infty) \), \( T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \mod p \) for \( n \geq 2 \) and \( p \) is a large prime while the semi group property, \( T_n(T_a(x)) = T_{ab}(x) = T_{ab}(T_a(x)) \mod p \) for any \( a, b \geq 2 \) still holds.

3 Cryptanalysis of Previous Protocol

In this section we would like to identify a vulnerability that appears in some of these protocols. The Lee et al.’s protocol [16] is selected for the purpose of illustration while similar vulnerability appears in other protocols as well [14, 17].

3.1 Review Lee et al.’s Protocol

Here we briefly review Lee et al.’s protocol [16]. The steps of it are illustrated in Figure 1. The setting of the protocol assumes there is a single honest server \( S \) and a set of clients \( \{U_1, U_2, \ldots\} \). Each client may desire to establish a confidential channel with another client through the server over a public network. The client will initiate the protocol while all messages sent in the protocol can be eavesdropped and changed by a network attacker. The goal of the attacker is to either convince a client or a server to authenticate a wrong client (impersonation) or to learn the session key of the confidential channel. We assume the attacker may exploit a small set of clients’ secrets or passwords.

Here we describe the protocol. Assume there are two client participants \( U_A \) and \( U_B \) desire to establish a session through a trusted server \( S \). Each client has a certificate issued by \( S \) prior to the protocol. \( T_A(ID_A) \) and \( T_B(ID_B) \) are the certificate of \( U_A \) and \( U_B \) respectively.

**Step 1.** \( U_A \) selects a random number \( a \in [1, p-1] \) and computes \( K_{AS} = T_AT_A(ID_A) \), \( H_a = h(T_A(ID_A)||ID_B)||ID_B) \) and \( C_A = E_{K_{AS}}(ID_A || ID_B || H_a || T_A(ID_A)) \). Note that \( h() \) denotes a one-way hash function based on chaotic maps, \( E_{K_A}(\cdot) \) means symmetric encryption function with key \( K \), and \( ID_A, ID_B \) means \( U_A \)'s and \( U_B \)'s identity respectively. Then, \( U_A \) sends \( m_1 = \{T_A(ID_A), C_A\} \) to \( U_B \).

**Step 2.** While receiving \( \{m_1\} \), \( U_B \) selects a random number \( b \in [1, p-1] \) and calculates \( K_{BS} = T_BT_B(ID_B) \), \( H_b = h(T_B(ID_B)||ID_B) \) and \( C_B = E_{K_{BS}}(ID_B || H_B || T_B(ID_B))) \). Then, \( U_B \) sends \( \{m_1, m_2\} \) to \( S \) where \( m_2 = \{T_B(ID_B), C_B\} \).

**Step 3.** Upon receiving \( \{m_1, m_2\} \), \( S \) first calculates \( K_{ST} = T_ST_A(ID_A), K_{SB} = T_ST_B(ID_B), D_A = D_{K_{ST}}(C_A) \), and \( D_B = D_{K_{SB}}(C_B) \). Note that \( D_{K}(\cdot) \) means a symmetric decryption function with key \( key \) and \( k \) means \( S \)'s secret key respectively. Then, \( S \) first checks \( ID_A \) and \( ID_B \).\( S \) also checks if \( H_a \) is equal to \( h(T_A(ID_A)||ID_B)||ID_B) \) and \( H_b \) is equal to \( h(T_B(ID_B)||ID_B) \) respectively. If both hold, \( S \) computes \( H_{SA} = h(T_B(ID_B)||T_A(ID_A)), H_{SB} = h(T_A(ID_A)||T_B(ID_B)), C'_A = E_{K_{SA}}(ID_B || ID_B || T_B(ID_B)) \), and \( C'_B = E_{K_{SB}}(ID_A || ID_A || T_A(ID_A)) \) and sends \( C'_A \) and \( C'_B \) to \( U_A \).

**Step 4.** \( U_B \) obtains and checks \( ID_A \) and \( H_{SB} \) by decrypting \( C'_B \). After that, \( U_B \) computes \( K = T_B(ID_B) \) and \( H_{SA} = h(K||C'_A) \). \( U_B \) then sends \( C'_A \) and \( H_{SA} \) to \( U_A \).

**Step 5.** \( U_A \) first checks \( H_{SA} \) by decrypting \( C'_A \) and then computes \( K = T_ST_A(ID_A) \). After that, \( U_A \) checks if \( H_{SA} \) is equal to \( h(K||C'_A) \). If it holds, \( U_A \) computes \( H_{AB} = h(K||ID_B)||T_B(ID_B)) \) and sends \( H_{AB} \) to \( U_B \).

Now, \( U_A, U_B \) and \( S \) can authenticate each other and establish the confidential channel using the session key \( SK \).
Cryptanalysis and Improvements on Three-party-authenticated Key Agreement Protocols Based on Chaotic Maps 681

3.2 Cryptanalysis of Lee et al.’s Protocol

After the protocol, $U_A$ and $U_B$ shall be authenticated by the server $S$ as they have demonstrated the knowledge of the certificates. However, we argue that it is not true. In fact, under a replay attack described below, $S$ still cannot confirm whether $U_A$ and $U_B$ are really initiating a 3AKE protocol or not. Sequently, the impact of this vulnerability will lead to an incorrect login status to the server and releasing unauthorized access of content, for example.

The replay attack is described as follows. Assume that there is a passive adversary $E$ eavesdrops $m_1 = \{T_a(ID_A), C_A\}$ from $U_A$ and $m_2 = \{T_b(ID_B), C_B\}$ from $U_B$. Later $E$ sends $m_1$ and $m_2$ to $S$. Since these two messages $m_1$ and $m_2$ are generated by honest clients, the identities of $U_A$ and $U_B$ are authenticated by $S$ although $U_A$ and $U_B$ are not intended to initiate a protocol. It shows that $E$ can successfully convince $S$ to authenticate two phantom clients in initiating the protocol. Consequently this will lead to an incorrect login record at the server and may release unauthorized access of content. Similar attack also applies to other protocols [14, 17].

4 The Proposed Protocol

Here we propose an efficient chaos-based 3AKE protocol. The notations used in the section are listed in Table 1. The proposed protocol has three phases, the initialization phase, the registration phase, and the authentication and key exchange phase.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_i$</td>
<td>a legitimate user $i$</td>
</tr>
<tr>
<td>$S$</td>
<td>server</td>
</tr>
<tr>
<td>$ID_i$</td>
<td>user $i$’s identity</td>
</tr>
<tr>
<td>$PW_i$</td>
<td>user $i$’s password</td>
</tr>
<tr>
<td>$h(\cdot), h_2(\cdot), h_3(\cdot)$</td>
<td>three secure one-way hash functions</td>
</tr>
<tr>
<td>$SK$</td>
<td>session key</td>
</tr>
</tbody>
</table>

4.1 The Initialization Phase

In this phase, the server $S$ initializes and selects the following system parameters.
- A large prime number $p$
- $a \in Z_p$, such that the minimal period of Chebyshev polynomial sequence $(T_a(a) \mod p)_{a=0}^{p}$ is $p+1$
- Three hash functions $h_1(\cdot), h_2(\cdot)$ and $h_3(\cdot)$
Finally, $S$ publishes the above parameters \( \{p, \alpha, h_1(\cdot), h_2(\cdot), \text{id}_S \} \).

### 4.2 The Registration Phase

This phase is invoked if a user $U_i$ desires to register himself to a server $S$. The detailed steps are described as follows.

Step 1: $U_i$ selects his $\text{id}_i$ and password $\text{pw}_i$ and then sends $\text{id}_i$ and $\text{pw}_i$ to $S$ through a secure channel.

Step 2: $S$ randomly selects a number $r_i \in \mathbb{R}[1, p+1]$ and computes $\text{id}_i = (r_i(\alpha) + \text{pw}_i) \mod p$. Then $S$ stores $\{\text{id}_i, r_i \}$ in its database and sends $\text{id}_i$ and $r_i$ to $U_i$ through a secure channel.

Step 3: $U_i$ keeps $r_i$ as a secret and does not reveal it to any other users.

### 4.3 The Authentication and Key Exchange Phase

Figure 2 illustrates the authentication and key exchange phase. Assume two users $U_A$ and $U_B$ attempt to authenticate each other and establish a common session key, they perform the following steps.

**Figure 2. The authentication and key exchange phase**

Step 1: $U_A$ randomly chooses $r_A \in \mathbb{R}[1, p+1]$ and computes $\text{id}_A = T_{r_A}(\alpha) \mod p$ and $r_{s_A} = r_A - \text{pw}_A$. $U_A$ also computes $H_{\text{AS}} = h_1(\text{id}_A \| \text{id}_B \| \text{id}_S \| R_A \| r_{s_A})$. Then $U_A$ sends $\{\text{id}_A, R_A, H_{\text{AS}}\}$ to $U_B$.

Step 2: $U_B$ chooses a random number $r_B \in \mathbb{R}[1, p+1]$ and then computes $r_B = T_{r_B}(\alpha) \mod p$. Then $U_B$ also computes $H_{\text{BS}} = h_1(\text{id}_B \| \text{id}_D \| \text{id}_S \| R_B \| r_{s_B})$. Then $U_B$ sends $\{\text{id}_B, R_B, H_{\text{BS}}\}$ to $S$.

Step 3: Upon receiving the messages sent from $U_B$, $S$ first verifies $H_{\text{AS}}$ and $H_{\text{BS}}$. If both hold, $S$ chooses a random number $R_S$ and computes $H_{\text{SA}} = h_1(\text{id}_A \| \text{id}_B \| \text{id}_S \| R_A \| R_B \| r_{s_A} \| r_{s_B})$ and $H_{\text{SB}} = h_1(\text{id}_B \| \text{id}_S \| H_{\text{BS}} \| R_B \| R_S \| R_{s_B})$ then sends $\{H_{\text{SA}}, H_{\text{SB}}, R_S\}$ to $U_B$.

Step 4: $U_B$ verifies if $H_{\text{SB}}$ is equal to $h_1(\text{id}_B \| \text{id}_S \| H_{\text{BS}} \| R_B \| R_{s_B} \| R_{s_S})$. If it holds, $U_B$ computes $K = T_{r_B}(R_S) = T_{r_B}(\alpha) \mod p$ and $H_{\text{BA}} = h_1(\text{id}_S \| H_{\text{BS}} \| K)$ then sends $\{R_B, H_{\text{DS}}, H_{\text{BS}}, H_{\text{AS}}\}$ to $U_A$.

Step 5: $U_A$ verifies if $H_{\text{SA}}$ is equal to $h_1(\text{id}_A \| \text{id}_B \| \text{id}_S \| R_A \| R_{s_A} \| R_{s_B})$. If it holds, $U_A$ computes $K = T_{r_A}(R_B)$. If both hold, $U_A$ and $U_B$ both compute $K$ and verify $H_{\text{BA}}$ and $H_{\text{SA}}$.
\[
T_{\alpha}(R_\alpha) = T_{\alpha'}(\alpha) \mod p. \quad U_A \text{ further checks if } H_{BS} \text{ is equal to } h_1(ID_S || H_{SA} || K). \text{ If it holds, } U_A \text{ sets } SK = h_1(K) \text{ as a session key. After that, } U_A \text{ computes } H'_{BS} = h_1(ID_A || ID_B || ID_S || R_A || R_B || R_S || r_{s_1}) \text{ and } M = h_2(K) \text{ and then sends } \{H'_{AS}, M\} \text{ to } U_B.
\]

Step 6: \( U_B \) verifies if \( M \) is equal to \( h_2(K) \). If it holds, \( U_B \) sets \( SK = h_3(K) \) as a session key. \( U_B \) computes \( H'_{BS} = h_1(ID_A || ID_B || ID_S || R_A || R_B || R_S || r_{s_1}) \) and then sends \( \{H'_{AS}, H'_{BS}\} \) to \( S \).

Step 7: \( S \) verifies the integrity of received \( \{H'_{AS}, H'_{BS}\} \). If both hold, \( S \) can assure that \( U_A \) and \( U_B \) have successfully established a common session key.

## 5 Security Analysis

In this section, we demonstrate the security of our protocol. We first analyze our protocol using the BAN logic [26]. Then we show that our protocol is secure against several kinds of attacks.

### 5.1 BAN Logic

The BAN logic is widely used to analyze the security of authenticated key agreement protocols. The detailed steps are listed in the following subsections.

**Notations of BAN logic.** Here we define some notations used in this analysis.

1. \( P \models X \): \( P \) believes \( X \) or called \( P \) would be entitled to believe \( X \). In particular, \( P \) may act as though \( X \) is true.

2. \( P \models X \): \( P \) sees \( X \). Someone has sent a message containing \( X \) to \( P \) and \( P \) can read and repeat \( X \).

3. \( P \models X \): \( P \) once said \( X \). \( P \) sent a message including \( X \) at some time. Note that it does not know whether the message was sent long ago or during the current run of the protocol, but it knows that \( P \models X \) when the message was sent.

4. \( P \models X \): \( P \) has jurisdiction over \( X \). \( P \) controls \( X \) which is subject to jurisdiction of \( P \) and \( P \) is trusted for \( X \).

5. \( \#(X) \): \( X \) is fresh. \( X \) has not been sent in a message at any time before the execution of current round of the protocol.

6. \( P \models X \): \( P \) and \( Q \) may use the shared key \( K \) to communicate securely. We say that \( K \) is good, if \( K \) will never be discovered by any principal except \( P \) or \( Q \), or a principal trusted by either \( P \) or \( Q \).

7. \( P \models X \): The formula \( X \) is a secret known only to \( P \) and \( Q \), and possibly to principals trusted by \( P \) and \( Q \).

8. \( \{X\}_K \): The formula \( X \) is encrypted under a key \( K \).

9. \( \langle X \rangle \): The formula \( X \) is combined with a secret \( Y \).

**BAN logic rules.**

1. **Message meaning rule for shared keys:**

   \[
P \models X \quad \leadsto \quad P \models \{X\}_K
   \]

   \( K \) is a shared key with \( Q \) and \( P \) sees \( X \) encrypted under \( K \), then \( P \) believes that \( Q \) once said \( X \).

2. **Message meaning rule for shared secrets:**

   \[
P \models \langle X \rangle \quad \leadsto \quad P \models \{X\}_K
   \]

   \( Y \) is a secret known only to \( P \) and \( Q \) and \( P \) sees \( Y \) under \( X \), then \( P \) believes that \( Q \) once said \( Y \).

3. **Nonce verification rule:**

   \[
P \models \#(X) \quad \leadsto \quad P \models \{X\}_K
   \]

   It means that if \( P \) believes that \( X \) is fresh and \( Q \) once said \( X \), then \( P \) believes \( Q \) believes \( X \).

4. **Jurisdiction rule:**

   \[
P \models X \quad \leadsto \quad P \models \{X\}_K
   \]

   It means that if \( P \) believes that \( Q \) has jurisdiction over \( X \) and believes \( Q \) believes \( X \), then \( P \) believes \( X \).

5. **Belief rule I:**

   \[
P \models \#(X) \quad \leadsto \quad P \models \{X\}_K
   \]

   It means that if \( P \) believes \( X \) and \( P \) believes \( Y \), then \( P \) believes \( (X, Y) \).

6. **Belief rule II:**

   \[
P \models \#(X, Y) \quad \leadsto \quad P \models \{X\}_K
   \]

   It means that if \( P \) believes \( Q \) believes \( X \) then \( P \) believes \( Q \) believes \( X \).

**Goals.** We want to show that our proposed 3AKE protocol should achieve the following goals:

\[
G_1: A \models \{A \leftrightarrow B\}.
\]

\[
G_2: B \models \{A \leftrightarrow B\}.
\]

\[
G_3: S \models \{A \leftrightarrow B\}.
\]

\[
G_4: A \models B \models \{A \leftrightarrow B\}.
\]

\[
G_5: B \models A \models \{A \leftrightarrow B\}.
\]

\[
G_6: S \models A \models \{A \leftrightarrow B\}.
\]

\[
G_7: S \models B \models \{A \leftrightarrow B\}.
\]

**Idealize the communication messages.** We idealize the communication messages of proposed protocol listed as below:

\[
M_1: A \rightarrow B: \{ID_A, R_{a_1}\}
\]

\[
M_2: A \rightarrow S: \{ID_A, R_{a_1}, H_{AS}\}
\]

\[
M_3: B \rightarrow S: \{ID_B, R_{b_1}, H_{BS}\}
\]

\[
M_4: S \rightarrow A: \{H_{SA}, R_{s_1}\}
\]

\[
M_5: S \rightarrow B: \{H_{SB}, R_{s_2}\}
\]

\[
M_6: B \rightarrow A: \{R_{b_1}, H_{BS}\}
\]

\[
M_7: A \rightarrow B: \{M\}
\]

\[
M_8: A \rightarrow S: \{H'_{AS}\}
\]

\[
M_9: B \rightarrow S: \{H'_{BS}\}
\]

**Initial state assumptions.** We define some initial state assumptions of the proposed protocol as follows:

\[
A_1: \; A \models \#(r_{a_1})
\]
can obtain

According to

can obtain

to

Since

meaning rule for shared secrets. Since

the initial state assumptions.

logic, we prove the proposed three parties key agreement protocol is said to provide perfect session key attack.

In an authenticated key agreement protocol, a known session key attack means that an adversary E still cannot compute the further session keys even if E obtains the session key SK. In the proposed protocol, the session key SK = T_{r_{12}}(α) only relies on the random values r_1 and r_2. Since the two values of one session are independent on once of other sessions. Hence, even if an adversary E obtains a session key SK = T_{r_{12}}(α), it cannot compute the further session key SK' = T_{r_{12}}(α) without knowing r_1' and r_2'. In other words, the proposed protocol is secure against a known session key attack.

Providing perfect forward secrecy. An authenticated key agreement protocol is said to provide perfect forward secrecy if an adversary E having both two user’s password and secret information R_{sk}, still unable to get previously generated session keys. To compute a session key in our protocol users choose their new random r_1 unique for each session, so fressness of session key is guaranteed. The adversary
having password and secret information \( R_e \) still need to know the session specific \( r_i \). Therefore the proposed protocol provides perfect forward secrecy.

**Impersonation attack.** If an adversary \( E \) wants to impersonate the user \( A \), he must obtain the key point \( r_{s_A} = R_{s_A} - PW_A \) for \( A \). However, it is impossible because \( A \)'s password \( PW_A \) and secret information \( A_S \) are kept secret. Hence, if \( E \) eavesdrops the message \( \{ID_A, R_A, H_{AS}\} \) it still cannot impersonate \( A \). By the similar reason, \( E \) cannot impersonate the user \( B \). Thus, the proposed protocol is secure against impersonation attacks.

**Resistance of password guessing attack.** As mentioned above, several 3AKE protocols suffered from password guessing attacks. In our protocol, the value \( R_e \) referred to password is kept secret. In other aspect, \( H_{AS} = h_1(ID_A \parallel ID_B \parallel ID_S \parallel R_A \parallel R_{s_A}) \), where \( r_{s_A} = R_{s_A} - PW_A \) is stored in database. Hence, it is not easy to guess the \( A \)'s password \( PW_A \). In other words, our protocol is secure against password guessing attacks.

### 6 Performance Evaluation

In this section, we evaluate the performance of our proposed protocol and then compare it with other well-known chaos-based 3PAKE protocols \([6, 8-12, 26]\) in Table 2. \( C \), \( H \) and \( S \) refer to a Chebyshev polynomial computation operation, a hash function operation and a symmetric encryption/decryption operation, respectively. Here we utilize the experimental results from Xue and Hong in 2011 [28]. According to their results, time consumptions for Chebyshev polynomial computation, symmetric encryption, and one-way hash function are 32.2ms, 0.45ms and 0.2ms, respectively under the environment of 3.2 GHz CPU and 3.0G RAM.

| Table 2. Performance comparisons |
|-------------------------------|------------------|-------------------|-------------------|------------------|
| User A                        | User B           | Server            | Total cost        | Time cost (ms)   |
| Lai et al. [6]                | 3C+5H            | 3C+5H             | 2C+6H+2S         | 8C+16H+2S        | 261.7            |
| Zhao et al. [8]               | 3C+6H+1S         | 3C+6H+1S         | 2C+8H+2S         | 8C+20H+4S        | 263.2            |
| Lee et al. [9]                | 3C+4H            | 3C+5H             | 2C+7H            | 8C+16H           | 260.8            |
| Farash et al. [23]            | 3C+4H            | 3C+5H             | 2C+7H            | 8C+16H           | 260.8            |
| Hu et al. [11]                | 3C+5H            | 3C+6H             | 2C+7H            | 8C+18H           | 261.2            |
| Xie et al. [10]               | 3C+5H+2S         | 3C+5H+2S         | 2C+4H+4S         | 8C+14H+8S        | 264              |
| Lee et al. [12]               | 4C+4H+2S         | 3C+4H+2S         | 4C+4H+4S         | 11C+12H+8S       | 360.2            |
| Ours                          | 2C+6H            | 2C+6H             | 2C+6H            | 6C+18H           | 196.8            |

As shown in Table 2, our protocol is the most efficient one since it uses less Chebyshev polynomial computation.

### 7 Conclusion

In this paper we identified a replay attack on a few existing 3AKE protocols. We then also proposed a new protocol. According to the security and performance analysis, our new protocol is secure against various kinds of attacks and has better efficiency compared with previous works.

### Acknowledgement

The work was supported in part by Shenzhen Technical Project under Grant number JCYJ2017030 7151750788, in part by Shenzhen Technical Project under Grant number KQJSCX20170327161755, and in part by the Natural Science Foundation of Fujian Province under Grant number 2018J01636.

### References


**Biographies**

**Chien-Ming Chen** received his Ph.D. from the National Tsing Hua University, Taiwan. He is currently an associate professor in Harbin Institute of Technology Shenzhen Graduate School, China. His current research interests include network security, mobile internet, wireless sensor network and cryptography.

**Linlin Xu** is currently pursuing the M.S. degree in Harbin Institute of Technology Shenzhen Graduate School, China. His current research interests include security protocol, mobile security, and cryptography.
**King-Hang Wang** received his Ph.D. from the National Tsing Hua University, Taiwan. He worked in the Hong Kong Institute of Technology in 2010 as a lecturer. He joined the Hong Kong University of Science and Technology in 2015 for teaching. His research focus is cryptography, mobile security, and steganography.

**Shuai Liu** is currently pursuing the M.S. degree in Harbin Institute of Technology Shenzhen Graduate School, China. His current research interests include security protocol, mobile security, and cryptography.

**Tsu-Yang Wu** received the Ph.D. degree in Department of Mathematics, National Changhua University of Education, Taiwan, in 2010. He is currently an associate professor in College of Computer Science and Engineering, Shandong University of Science and Technology, China. His research interests include applied cryptography, pairing-based cryptography and information security.