On the Security of a Three-party Authenticated Key Agreement Protocol based on Chaotic Maps

Chien-Ming Chen
Harbin Institute of Technology, Shenzhen
HIT Campus Shenzhen University Town Xili, Shenzhen, China
chienming.taiwan@gmail.com

King-Hang Wang
Hong Kong University of Science and Technology, Hong Kong
kevinw@cse.ust.hk

Tsu-Yang Wu
Fujian Provincial Key Laboratory of Big Data Mining and Applications
National Demonstration Center for Experimental Electronic Information and Electronic Technology Education
Fujian University of Technology, Fujian, China
wutsutyang@gmail.com

Eric Ke Wang
Harbin Institute of Technology, Shenzhen
HIT Campus Shenzhen University Town Xili, Shenzhen, China
wk_hit@hitsz.edu.cn

ABSTRACT. Very recently, Lee et al. proposed a new three-party-authenticated key exchange protocol based on chaotic maps without using a password table. Their protocol enjoys many advantages over previous protocols, such as avoiding SQL injection or password guessing attacks. In this paper, however, we demonstrate that this protocol is insecure against a registration attack and a identity morphing attack. To the best of our understand, this is the first paper in the literature to demonstrate, with solid mathematical proofs, how registration attacks can be launched against chaotic based key exchange protocols. This idea may also be generalized to apply on other cryptographic protocols that user ID and credentials are highly tangled. Subsequently, we recommend some implementation level patches for the system.

Keywords: Cryptanalysis, Chaotic Map, Authenticated key agreement

1. Introduction. Secure communication over public network relies on secure encryption. An encryption is secure only if the encryption key can be securely setup. Turing award laureates Diffie and Hellman [1] established a fundamental model for key setup as known as the Diffie-Hellman Key Exchange and subsequently gave birth to an important stream of key exchange protocols, the Password-based Authenticated Key Exchange (PAKE for short), which utilizes passwords as a proof of identity. Given that a short password can be easily remembered and operated by human, PAKE received many attention in both the research community and the real world [2, 3, 4, 5].

In the recent years, some PAKE protocols built on chaotic maps [6, 7, 8, 9, 10] were proposed. Chaotic maps usually appear as iterating functions and exhibit chaotic behaviors
which appeal to fit in cryptographic use. Some chaotic based PAKE protocols for two parties [11, 12, 13, 14] first appeared in the literature. They are suitable for client-to-server environment but less suitable to client-to-client environment. Later on three-party chaotic based PAKE protocols were also proposed [15, 16, 17, 18, 19, 20]. These protocols are designed for client-to-client key exchange while the identities of clients are authenticated by a trusted server.

In 2015, Lee et al. [15] proposed a new three-party protocol based on chaotic maps. This work intended to improve the security over previous works. Instead of using a password table stored in the server, it generates users’ credential with their ID/username. Similar ideas can be traced back to Shamir’s ID based cryptography [21] and its realization by Boneh [22]. Since no password table is required, it reduces the server effort in maintaining the secrecy and consistency of a database table; instead the server only needs to securely store a server key. Potential threats like SQL injection [23] or offline cracking password shadow [24] are thus avoided. In this paper, unfortunately we found a loophole in Lee et al.’s work. Following this loophole, we devise two attacks to break the protocol. In particular, this protocol allows a malicious user to compute a legitimate user’s credential by carefully select a new username. We further propose several methods to patch the protocol and discuss their effectiveness.

1.1. Contribution. We first demonstrate how a registration attack can be launched against chaotic based protocols. The attacks illustrated in this paper can also easily generalized to apply on other ID-based cryptographic protocols which have ID and credential highly tangled, if similar loophole appears. The amendment provided in the late part of this paper constructively fixes Lee et al.’s protocol with minimal implementation efforts.


Chebyshev’s polynomial is one of the chaotic map that receives high attention in the literature. The extended Chebyshev’s polynomial [29] which developed from Chebyshev’s polynomial [30] is selected in Lee et al.’s work and is reviewed in this section. A Chebyshev’s polynomial \( T_n(x) \) is defined as follows:

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

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

\[
T_2(x) = 2xT_1(x) - T_0(x) = 2x^2 - 1, \\
T_3(x) = 2xT_2(x) - T_1(x) = 4x^3 - 3x, \\
T_4(x) = 2xT_3(x) - T_2(x) = 8x^4 - 8x^2 + 1.
\]

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 the Chebyshev polynomial \( T_n(x) : [-1, 1] \to [-1, 1] \) by \( T_n(x) = \cos(n \cdot \arccos(x)) \).

For \( n \geq 2 \) the Chebyshev’s polynomial \( T_n(x) \) satisfies the following two properties:
(1) The semi group property.
\[
T_a(T_b(x)) = \cos(a \arccos(\cos(b \arccos(x))))
\]
\[
= \cos(ab \arccos(x))
\]
\[
= T_{ab}(x)
\]
\[
= T_b(T_a(x))
\]
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 \((-\infty, \infty)\), \(T_n(x) \equiv 2xT_{n-1}(x) - T_{n-2}(x)\) mod \(p\) for \(n \geq 2\) and \(p\) is a large prime while the semi group property, \(T_a(T_b(x)) \equiv T_{ab}(x) \equiv T_b(T_a(x))\) mod \(p\) for any \(a, b \geq 2\) still holds. Kocarev [29] provided the period of enhanced Chebyshev’s polynomial as the lemma below.

**Lemma 2.1.** Let \(N\) be an odd prime and let \(x \in Z\) such that \(0 \leq x < N\). Then the period of the sequence \(T_n(x)\) mod \(N\), for \(n = 0, 1, 2, \ldots\), is a divisor of \(N^2 - 1\).

**Lemma 2.2.** Let \(p\) be a prime, \(\forall a, b \in Z\) and \(T_n(x)\) mod \(p\) is a Chebyshev’s polynomial of degree \(n\). Given the value \(a\) and \(T_{ab}(x)\), \(T_b(x)\) can be computed in polynomial time.

**Proof of Lemma 2.2.** From Lemma 2.1 the period of the polynomial is divisible by \(p^2 - 1\). \(a^{-1}\), the inverse of \(a\) over \(p^2 - 1\), can be computed using Chinese Reminder Theorem [31]. By the semi-group property, \(T_{a^{-1}}(T_{ab}(x)) = T_b(x)\). \(\square\)

3. **Review of Lee et al.’s Protocol.** In this section, we briefly review Lee et al.’s protocol (illustrated in Fig. 1). The detailed of their protocol can refer to their original paper [15]. We adopt their symbols of notation and summarized them in the following table.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A, B)</td>
<td>Two legitimate users</td>
</tr>
<tr>
<td>(S)</td>
<td>A server for establishing the protocol</td>
</tr>
<tr>
<td>(ID_A, ID_B)</td>
<td>The ID names of users (A) and (B)</td>
</tr>
<tr>
<td>(k)</td>
<td>The server’s secret key</td>
</tr>
<tr>
<td>(h())</td>
<td>A secure one-way hash function</td>
</tr>
<tr>
<td>(p)</td>
<td>A sufficient large prime number</td>
</tr>
<tr>
<td>(T_x())</td>
<td>Extended Chebyshev chaotic function over prime (p)</td>
</tr>
<tr>
<td>(E_K(), D_K())</td>
<td>Secure symmetric encryption and decryption function with the key (K).</td>
</tr>
<tr>
<td>(T_b(ID_X))</td>
<td>A credential for a user (X) that is issued by the server during registration.</td>
</tr>
</tbody>
</table>

3.1. **Lee et al.’s Protocol.** It is assumed there exist a server \(S\) where various users registered with it and desire to establish secure communication. \(S\) owns a secret key \(k\) and securely stored. \(S\) will also publish a set of system parameters: \(p\) as a large prime number; \(T()\) a Chebyshev chaotic function over \(p\); \(E()\) and \(D()\) a pair of encryption and decryption function; \(h()\) a secure one-way hash function.
Each user has registered a user ID and received a credential. Assuming a user A should have registered ID$_A$ with the server and received a credential $T_k(ID_A)$. Note this credential is computed using server’s secret key $k$ using the chaotic function with ID$_A$ as the initial seed.

Assume that two participants A and B desire to establish a common session key with S.

Step 1. A selects a random number $a \in [0, p - 1]$ and computes (1),(2),(3).

$K_{AS} = T_a(T_k(ID_A))$  \hspace{1cm} (1)

$H_A = h(T_a(ID_A)||ID_A||ID_B)$  \hspace{1cm} (2)

$C_A = E_{K_{AS}}(ID_A||ID_B||H_A||T_a(ID_B))$  \hspace{1cm} (3)

Then, A sends $m_1$ to B where $m_1 = \{T_a(ID_A), C_A\}$. Step 2. Upon receiving $m_1$, B selects a random number $b \in [0, p - 1]$ and calculates (4),(5),(6).

$K_{BS} = T_b(T_k(ID_B))$  \hspace{1cm} (4)

$H_B = h(T_b(ID_B)||ID_B)$  \hspace{1cm} (5)

$C_B = E_{K_{BS}}(ID_B||H_B||T_b(ID_B))$  \hspace{1cm} (6)

\[1\] The credential was referred as certificated in their original paper. However it should be aware this credential should not be known by any attack as certificate/public key usually does. Otherwise anyone could impersonate the user as shown in the later of the paper.
Then, $B$ sends $\{m_1, m_2\}$ to $S$ where $m_2 = \{T_b(x), ID_B, C_B\}$.

Step 3. While receiving $\{m_1, m_2\}$, $S$ calculates $(7),(8),(9),(10)$.

\[
\begin{align*}
K_{SA} &= T_k(T_a(ID_A)) \\
K_{SB} &= T_k(T_b(ID_B)) \\
D_A &= D_{K_{SA}}(C_A) \\
D_B &= D_{K_{SB}}(C_B)
\end{align*}
\]

$S$ checks if $H_A$ is equal to $h(T_a(ID_A)||ID_B)$ and $H_B$ is equal to $h(T_b(ID_B)||ID_B)$ respectively. If both holds, $S$ calculates $(11)(12)(13)(14)$ and sends $(13)(14)$ to $B$.

\[
\begin{align*}
H_{SA} &= h(T_k(ID_A)||T_a(ID_A)) \\
H_{SB} &= h(T_k(ID_B)||T_b(ID_B)) \\
C'_A &= E_{K_{SA}}(ID_A||ID_B||T_b(ID_B)||H_{SA}) \\
C'_B &= E_{K_{SB}}(ID_A||ID_B||T_a(ID_B)||H_{SB})
\end{align*}
\]

Step 4. $B$ obtains and checks $ID_A$ and $H_{SB}$ by decrypting $C'_B$. After that, $B$ computes $SK = T_b(T_a(ID_B))$ and $H_{BA} = h(SK||C'_A')$. $B$ then sends $C'_A$ and $H_{BA}$ to $A$.

Step 5. $A$ checks $H_{SA}$ by decrypting $C'_A$ and then computes $SK = T_a(T_b(ID_B))$. $A$ also checks if $H_{BA}$ is equal to $h(SK||C'_A)$. If it holds, $A$ computes $H_{AB} = h(SK||ID_A||T_a(ID_B))$ and sends $H_{AB}$ to $B$.

Step 6. $B$ calculates $h(SK)$ after verifying $H_{AB}$. Note that $h(SK)$ is the session key between $A$ and $B$.

3.2. Security Claimed by the Authors. Lee et al. claimed their protocol can securely authenticate the protocol participants. This means that no any adversary may impersonate others to complete the protocol with another honest client. The protocol also preserves user’s anonymity that no one except the participants and the server will know the users identities. Besides, it does not require the use of password table so that it would immune from password guessing attacks or insider leaking passwords.

4. Cryptanalysis on Lee et al.’s Protocol. In this section, we would like to demonstrate two vulnerabilities of the protocol. The first one we call it a ”Registration Attack”. The second vulnerability is an variant of the first one and we call it a ”Identity Morphing”. We first demonstrate the registration attack by the following two Lemmas. Then we illustrate the second attack with subtly alternation on the registration attack.

4.1. Registration Attack.

**Lemma 4.1.** There exist an attacker who can impersonate a legitimate user $A$ if the attacker process $A$’s credential $T_k(ID_A)$.

**Lemma 4.2.** There exist an attacker who has non-negligible probability to obtain a selected victim $A$’s credential after several registration interaction with the server.

It would be obvious the protocol is not sufficient to provide enough security given the correctness of the two lemma.

**Proof of Lemma 4.1.** Assuming there is an attacker $E$ obtaining user $A$’s credential $T_k(ID_A)$. As the protocol initializer $E$ can execute the protocol by computing equations $(1)(2)(3)$ to invoke a session with $B$ (or any other legitimate users). At Step 5, $E$ has the ability to compute $SK$ as he knew the value of $a$. Then he can also compute $H_{AB}$ based on the knowledge of $SK$. All values $E$ computes would be legitimate and be verified by $S$ or $B$. Thus $E$ can impersonate $A$ to establish connection with $B$ and $S$. 
Similarly, if $E$ has obtained user $B$’s credential $T_k(ID_B)$, he would be able to impersonate $B$ in the protocol as a responder. On receiving the message at Step 2, $E$ computes the equations (4), (5), (6) with his knowledge on $T_k(ID_B)$. At Step 4, $E$ follows the protocol to compute $SK$ with his knowledge of $K_{SB}$ and $b$. Again, all values computed can be verified by the protocol initializer and the server thus $E$ can impersonate $B$ to establish connection with others.

Although the authors name the credential as certificate or other explicit statement, we have no doubt the author had recognized this lemma.

**Proof of Lemma 4.2.** It is not explicit in the original paper of how a user may register an account from the server. We assume it works like a normal account registration process that we usually found on web in our daily life. That is, a user may select an ID during registration and that ID is required to fulfill certain requirement like a limitation of length, or not containing special characters, etc. We assume there is an attacker $E$ who wishes to obtain the credential of user $A$ with the user ID $IDA$. $E$ conducts Algorithm 1 to obtain $ID'$ and a counter value $\lambda$. We argue later that there is non-negligible probably the following algorithm will return without give up. $E$ then registers the $ID'$ to the server. The server will compute the credential $T_k(ID')$ and return to $E$. On receiving the $T_k(ID')$, $E$ computes $\alpha = T_{\lambda-1}(T_k(ID'))$. Since $ID' = T_\lambda ID_A$, according to Lemma 2.2, $\alpha = T_{\lambda-1}(T_k(T_\lambda(ID_A))) = T_k(ID_A)$.

**Algorithm 1** Algorithm for the registration attack

```plaintext
function GEN_ID(string $IDA$)
    $\lambda \leftarrow 2$
    $ID' \leftarrow T_\lambda(ID_A)$
    while $ID'$ is not a valid ID do
        $ID' \leftarrow T_\lambda(ID_A)$
        $\lambda \leftarrow$ random
        if timeout then give up and exit
        end if
    end while
    return $ID'$, $\lambda$
end function
```

The probability for the algorithm returns without give up depends on the timeout period, the density of valid ID, the size of the extended Chebyshev’s polynomial ($p$). Assume the size of $p$ is 1024-bits, an attacker has two weeks time to run the program on 10 time more computation power, and password allows all alphabet and numeric set for no more than 20 characters. As the encoding method from ID to number is unspecified, we assume using a very simple method, taking the alphabet and numeric set into consideration ($26 + 26 + 10 = 62$ combinations). If the targeted number $IDA$ has a length shorter than 9 characters ($< 62^9$), $T_2(ID_A) = 2(ID_A)^2 - 1$ has a length less than 19 characters ($< 62^{19}$) already.

This is an illustration of how attack works against an unspecified but reasonable implementation. In case the situation is less favorable to the attacker, say there is another string encoding algorithm, a more restrictive ID checking rules, etc. There are plenty of rooms in revising the algorithm to work out a solution with non-negligible probability.

4.2. **Identity Morphing Attack.** This attack allows an attacker $E$ to forge different user identities after registered with the server for once. Assuming that $E$ owns a credential $T_k(ID_E)$ after a registration with server. Unlike the previous attack, it does not require
a special choice on the ID. During the protocol execution, \( E \) runs Algorithm 1 to obtain another \( ID' \) and \( \lambda \). Then he computes a credential \( \beta = T_\lambda(T_k(ID'_E)) = T_k(T_\lambda(ID'_E)) = T_k(ID') \). Then he imitates the protocol using \( <ID', \beta> \).

This attack is based on the assumption that the server does not securely maintain a user ID list or the server does not explicitly check the validity of the ID during the protocol execution. In that paper, neither of these two protection mechanisms is explicit.

5. **Quick Patching the System.** There are several ways to fix the protocol by revising the protocol design. However, this will cause the protocol losing some nice property like keeping no password table and this will not be discussed in this paper. Instead we try to explore several ways that can be patch the system through implementation or policy level’s amendment.

1. **Check user ID during protocol execution.** As pointed by the second attack, the need of checking user’s identity against a secure maintained user list is required. The list shall be securely store so that it is unrevealed and unmodified by other parties. The protocol shall also ends with the same error code in the case of wrong user identity or wrong credential so that attackers would not be able to learn the set of registered IDs.

2. **Disable the choose of ID.**
   
   The attack mentioned in this paper requires the choose of ID from the attacker; therefore, the most trivial way to stop the attack is to disallow user to choose their ID. The registration process simply assign an ID to a user. However, since the ID is likely to be very short to be human processable, it is still possible a coincident would occur, i.e., \( T_2(ID_A) = ID_B \) and the protocol is still insecure.

3. **Tiding Users’ Real Identity with their ID.** Another way to prevent the problem is to link the user’s real identity with their ID. On registration a user will need to present their real identity. This practice has been implemented in certain country already [32]. This will significantly reduce the attempt for user to launch this attack. However, this approach does not really address the problem from the root as there is no evident of any person registering an account has an intention to launch this attack. Secondly, once an account being impersonated, there is no trace of which attacker account is in fact launching the attack.

4. **Padding on ID.**
   
   This approach is more on the implementation details. Instead of creating the credential with \( T_k(ID_A) \), the system instead apply certain padding protocol to the ID to create the credential. For instance, one may append \( k \) zeros after the ID, like \( T_k(ID_A || 0^k) \). Or a more sophisticated approach that is used in digital signature like OAEP [33] and its variant [34]. The result brought from digital signature can be applied here - it is merely impossible for attacker to compute the credential for another chosen ID using this attack.

6. **Conclusion.** In this paper, we have identified a security loophole in Lee et al.’s protocol and devised two distinct attacks to break their protocol through solid mathematically proofs. We also recommend several system-wise patches for systems which are using this protocol. Our work may also be generalized to attack other ID-based cryptographic protocol if the credentials are deducible by carefully selecting ID. This would demonstrate to the cryptographic community to carefully design their protocol.
Acknowledgement. The work of Chien-Ming Chen was supported in part by the Shenzhen Peacock Project under Contract KQJSCX20170327161755 and in part by Shenzhen Technical Project under Grant number JCYJ20170307151750788. The work of Eric Ke Wang was supported in part by National Natural Science Foundation of China (No.61572157), grant No.2016A030313660 from Guangdong Province Natural Science Foundation, JCYJ20160608161351559 from Shenzhen Municipal Science and Technology Innovation Project.

REFERENCES

On the Security of a Three-party Authenticated Key Agreement Protocol


Chien-Ming Chen received his PHD from the National Tsing Hua University, Taiwan. He is currently an associate professor Harbin Institute of Technology Shenzhen Graduate School, China. He has published over 40 journal papers and 50 conference papers. His current research interests include network security, mobile internet, wireless sensor network and cryptography.
King-Hang Wang

King-Hang Wang received his PhD from the National Tsing Hua University and BEng from the Chinese University of Hong Kong. He worked in the Hong Kong Institute of Technology in 2010 as a lecturer. He joined the Hong Kong University of Science and Technology since 2015. His research focus is cryptography, mobile security, and provable authentication.

Tsu-Yang Wu

Tsu-Yang Wu received the PhD degree in Department of Mathematics, National Changhua University of Education, Taiwan in 2010. Currently, he is an associate professor in College of Information Science and Engineering at Fujian University of Technology, China. In the past, he is an assistant professor in Innovative Information Industry Research Center at Shenzhen Graduate School, Harbin Institute of Technology, He is a member of Chinese Cryptology and Information Security Association (CCISA) and China Computer Federation (CCF). He serves as associate editor of two international journals: Data Science and Pattern Recognition, Journal of Network Intelligence. His research interests include cryptography and Network security.

Eric Ke Wang

Eric Ke Wang is an associate professor of Harbin Institute of Technology (HIT), Shenzhen, China. Currently, he work as a senior researcher at Key Laboratory of Shenzhen Internet Information Collaborative Technology and Application of HIT. He received a Phd from department of computer science, the University of Hong Kong in 2009. His main research interests include network security, deep learning. He has obtained two granted projects from National Science Funding (NSFC) of China. Besides, he has developed two software platforms for opportunistic social networks and obtained two authorized related patents.