A Three-Party Password-based Authenticated Key Exchange Protocol for Wireless Communications

Yanrong Lu\textsuperscript{1,2}, Lixiang Li\textsuperscript{1,2}, Haipeng Peng\textsuperscript{1,2}, Yixian Yang\textsuperscript{1,2}

\textsuperscript{1} Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China
\textsuperscript{2} National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, China
e-mail: li_lixiang2006@163.com

Abstract.

A three-party password-based authenticated key exchange (3PAKE) protocol is an important cryptographic primitive which allows two entities to establish a session key with the help of a trusted server through an insecure channel. Recently, Farash and Attari (Information Technology and Control 43(2), 143-150, 2014) presented an improved 3PAKE protocol to erase the security flaws found in Tallapally’s 3PAKE protocol (Information Technology and Control 41(1), 15-22, 2012). They claimed that their improved protocol could withstand many security attacks. However, we identified that Farash and Attari’s protocol was still sensitive to the off-line password guessing attack which directly resulted in defencelessness to the impersonation attack. In order to cope with the loopholes of Farash and Attari’s protocol, we proposed a modified 3PAKE protocol without using smart cards for wireless communications. We demonstrate that the proposed protocol can mitigate all the problems of the protocol of Farash and Attari and possess more security properties. In addition, we make a comparison among the proposed protocol and the other related protocols regarding the performance and security properties.

Keywords: three-party; password-based; authenticated key exchange; wireless communications.

1. Introduction

With the rapid development of wireless communication, various portable devices (mobile phone, Laptop, USB thumb drives and PDAs.) have dramatically increased to provide a more convenient life to people. People can roam freely and use mobile services almost everywhere. However, open access to wireless services for wireless environment has raised a number of security concerns. Authenticated key exchange (AKE) protocols were proposed to authenticate the identities of entities involved in the open communication. Password-based authenticated key exchange (PAKE) protocol is a type of AKE protocols. It is where two or more parties, based only on their knowledge of a password, establish a session key by messages exchange, thus preventing an unauthorized entity from participating in the protocol. Hitherto, PAKE is widely applied because of its simplicity, convenience, adaptability, mobility, and less hardware requirement [1].

Bellovin and Merritt [2] proposed the first two-party authenticated key exchange (2PAKE) protocol which is employed to establish a session key between two communication parties. After that, numerous 2PAKE protocols were presented for different communication environments [3-12]. However, 2PAKE protocols cannot be applied in large-scale peer-to-peer architecture since each user must store a different password for each partner it communicates with which may strain the storage capacity of the users. Subsequently, researchers provided three-party authenticated key exchange (3PAKE) to conquer this problem effectively [13-19]. In 2009, Huang [20] proposed a 3PAKE protocol without using smart cards in the hope that the proposed scheme was secure against various attacks. Unfortunately, Tallapaly [21] pointed out that the protocol proposed by Huang was prone to suffer from undetectable online password guessing attack. In order to thwart security attack found in Huang, Tallapaly developed an enhanced 3PAKE protocol which requires only four message transmission rounds. Nevertheless, Farash and Attari [22] observed that Tallapally’s protocol was still vulnerable to undetectable online password guessing and off-line password guessing attacks. Farash and Attari then proposed their modified protocol based on Tallapally’s protocol and claimed that their protocol was immune to many kinds of security attacks.
In this paper, we focus on the protocol provided by Farash and Attari. We found that the protocol was still insecure against off-line password guessing attack which directly resulted in defenselessness to the impersonation attack. In order to cope with the loopholes of Farash and Attari’s protocol, we proposed a modified 3PAKE protocol without using smart cards for wireless communications. We demonstrate that the proposed protocol can mitigate all the problems of the scheme of Farash and Attari’s protocol and possess more security properties. In addition, we make a comparison between the proposed protocol and the other related protocols regarding the performance and security properties.

The remainder of this paper is organized as follows. Section 2 and Section 3 review and analyze Farash and Attari’s protocol. Section 4 presents our proposed protocol. Section 5 analyzes the proposed protocol. The performance and security properties comparison among the proposed protocol and other related schemes are shown in Section 6. Section 7 is a brief conclusion.

2. Review of Farash and Attari’s protocol

In this section, we briefly review Farash and Attari’s 3PAKE protocol. First, in Table 1, we introduce some notations used in this paper.

Table 1. Notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, B</td>
<td>Users</td>
</tr>
<tr>
<td>S</td>
<td>Server</td>
</tr>
<tr>
<td>pw_r</td>
<td>The password of r</td>
</tr>
<tr>
<td>p, q</td>
<td>Large prime numbers, where q = 2p - 1</td>
</tr>
<tr>
<td>g</td>
<td>A generator of G</td>
</tr>
<tr>
<td>Z_q</td>
<td>The ring of integers modulo q</td>
</tr>
<tr>
<td>Z'_q</td>
<td>The multiplicative group of Z_q</td>
</tr>
<tr>
<td>F_r()</td>
<td>A trapdoor function</td>
</tr>
<tr>
<td>h()</td>
<td>A one-way hash function</td>
</tr>
<tr>
<td>⊕</td>
<td>Exclusive-or operation</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1) A randomly selects x , r_a ∈ Z'_p , and computes
\[ R_a = g^x + h(pw_r, A, B) \mod q . \]
Then, he sends the message \{A, B, R_a\} to S. Similarly, B also generates two random numbers y , r_b ∈ Z'_p , and computes
\[ R_b = g^y + h(pw_b, A, B) \mod q . \]
B sends \{A, B, R_b\} to S.

2) Upon receiving the messages, S first computes
\[ K_a = h(R_a || A || pw_r || B) \mod q , \]
and \[ K_b = h(R_b || A || pw_b || B) \mod q . \]
After that, S selects a random number z ∈ Z_p and computes,
\[ K_{sa} = (R_a)^z \mod q , \]
\[ K_{sb} = (R_b)^z \mod q , \]
\[ Z_s = h(pw_r || A || K_{sa} || B || 0 || T || S) , \]
\[ Z_a = h(pw_r || 0 || K_{sa} || B || A || S) . \]
Finally, S sends back the message \{Z_s, T_s\} and \{Z_a, T_a\} to A and B, respectively.

3) After receiving the message, A computes,
\[ K_{sa} = (T_s)^y \mod q \]
and checks whether \( Z_s = Z_a \). If holds, A computes \( V_a = h(A || B || K_{sa} || pw_a || 0 || T || S) \) and sends \( V_a \) to S. Simultaneously, once receiving the message, B also computes \( K_{sb} = (T_b)^y \mod q \) and verifies whether
\[ Z_b = h(pw_b || A || K_{sb} || B || 0) . \]
If it is correct, S is authenticated.

Then, B computes \( V_b = h(A || B || K_{sb} || pw_b || 0 || T || S) \) and sends \( V_b \) to S.

4) When receiving the messages, S checks whether \( V_a = h(A || B || K_{sa} || pw_a || 0 || T || S) \) and \( V_b = h(A || B || K_{sb} || pw_b || 0 || T || S) \).
If hold, S computes \( X_a = K_{sa} + h(1 || A || B || S || K_{sa} || pw_a) \)
and \( X_b = K_{sb} + h(1 || A || B || S || K_{sa} || pw_b) \). Then, S sends \( X_a \) and \( X_b \) to A and B, respectively.

5) When receiving the message, A computes \( K_{sa} = X_a - h(A || B || K_{sa} || pw_{sa} || S || 1) \), \( K_{sb} = (K_{sb})^y \mod q \), \( S_a = h(K_{sa} || A) \) and sends \( S_a \) to B. Simultaneously, B computes \( K_{sa} = X_b - h(A || B || K_{sa} || pw_{sa} || S || 1) \), \( K_{sb} = (K_{sb})^y \mod q \), \( S_b = h(K_{sa} || B) \) and sends \( S_b \) to A.

6) Finally, A and B check the correctness of \( S_a \) and \( S_b \) and respectively compute the session key
\[ sk = h(A || B || S || K_{sa} || K_{sb} || K_{sa} \). \]

3. Security analysis of Farash and Attari’s protocol

In this section, we analyze the protocol proposed by Farash and Attari. The following attacks are based on the assumption that a malicious adversary has totally supervised the communication channel and has the capacity to intercept, insert, delete, refresh or update any information delivered in the public channel [23].

3.1. Off-line password guessing attack

Assume A is a malicious user. He can obtain the password of the real user A by performing the following process.

1) A guesses a password \( pw_r \) and computes \( R_a = g^x + h(pw_r, A, B) \), where \( x \) is a random
number of $A$. Finally, $A$ submits $\{A, B, R_3\}$ and $\{B, A, R_3\}$ to $S$.

2) When receiving the messages, $S$ computes $R_a = R_a - h(p_{wa}, A, B)$, $R_b = R_b - h(p_{wb}, A, B)$. After that, $S$ generates a random number $z \in Z_p^*$ and computes the values $T_s = g^z$, $K_\alpha = (R_a^z)$, $K_{ab} = (R_b^z)$, $Z_a = h(0, A, B, S, p_{wa}, K_{ab})$ and $Z_b = h(0, A, B, S, p_{wb}, K_{ab})$. Then, $S$ sends back $\{T_s, Z_a\}$ and $\{T_s, Z_b\}$ to $A$ and $B$, respectively.

3) When $A$ receives the message, he computes $K_{aa} = (T_s)^{x}$ at first. $A$ then checks whether $h(0, A, B, S, p_{wa}, K_{ab}) = Z_a$. If the equation is true, $A$ has guessed the correct password; Otherwise, he continues to guess a candidate password and repeat 1)-3) until he succeeds.

3.2. Impersonation attack

As described in the previous subsection, when an adversary got the password of one of the entities, he could easily impersonate as the legal user to cheat the server. The details will follow.

1) Assume $A$’s password is leaked. The adversary $A$ generates a random number $x$, computes $R_a = g^x + h(p_{wa}, A, B)$, and sends $\{A, B, R_3\}$ and $\{B, A, R_3\}$ to $S$.

2) After receiving the message, $S$ performs the original protocol in their scheme without detecting $A$ who is actually a disposer. Finally, $S$ respectively delivers $\{T_s, Z_a\}$ and $\{T_s, Z_b\}$ to $A$ and $B$.

3) When receiving the message, the adversary first checks the correctness of $Z_a$. If it holds, he computes $V_a = h(A, B, S, p_{wa}, K_{ab}, T_s)$, and sends $V_a$ and $V_b$ to $S$.

4) After receiving the message, $S$ surely verifies the correctness of $V_a$ since all the values indeed come from $A$. Then, $S$ computes $X_a = K_{ab} + h(1, A, B, K_{aa}, S, p_{wa})$, $X_b = K_{ab} + h(1, A, B, K_{ab}, S, p_{wa})$ and sends $X_a$ and $X_b$ to $A$ and $B$, respectively.

5) When receiving the message, $A$ computes $K_{ab} = h(1, A, B, S, p_{wa}, K_{ab})$, $S_a = h(K_{ab}, A)$, $K_a = (R_a^z)^{g^{x_1}}$, $S_b = h(K_{ab}, B)$ and the session key $sk = h(A, B, S, K_{aa}, K_{ab}, K_{ab})$. That is, $A$ successfully has been authenticated by the server.

4. The proposed protocol

In this section, we introduce our enhanced protocol which inherits the merits and remedies the weaknesses of Farash and Attari’s protocol. The proposed protocol is shown in Fig. 1.

1) $A$ chooses two random numbers $x, r_a \in Z_p^*$ and computes $H_a = h(p_{wa}) \oplus r_a$, $V_a = h(p_{wa} \parallel r_a \parallel A)$, and $r_3 = x \oplus V_a \mod q$. $A$ sends $\{H_a, V_a, R_3, A\}$ to $S$.

Similarly, $B$ also generates two random numbers $y, r_b \in Z_p^*$ and calculates the values $H_b = h(p_{wb}) \oplus r_b$, $V_b = h(p_{wb} \parallel r_b \parallel B)$, and $r_3 = y \oplus V_b \mod q$. $B$ sends $\{H_b, V_b, R_3, B\}$ to $S$.

2) When receiving the messages, $S$ first derives the two random numbers $r_a$ and $r_b$ using the known passwords $p_{wa}$ and $p_{wb}$.

Then, $S$ verifies whether $h(A \parallel p_{wa} \parallel r_a)$ and $h(B \parallel p_{wb} \parallel r_b)$ are equal to the received $V_a$ and $V_b$, respectively. If hold, $S$ computes $R_a = R_a \oplus h(A \parallel p_{wa} \parallel r_a)$, and $R_b = R_b \oplus h(A \parallel p_{wa} \parallel r_b)$. After that, $S$ selects a random number $z \in Z_p^*$ and computes $N_a = g^z \mod q$, $K_{aa} = (R_a^z)^{g^{y^r}}$, $K_{ab} = (R_b^z)^{g^{x^r}}$, $T_s = h(p_{wa} \parallel r_a \parallel K_{ab} \parallel A)$, $T_b = h(p_{wb} \parallel r_b \parallel K_{ab} \parallel B)$. Finally, $S$ sends back the messages $\{T_s, B, N_a\}$ and $\{T_s, A, N_a\}$ to $A$ and $B$, respectively.

3) Upon receiving the message, $A$ computes $K_{aa} = (N_a)^{x_1} \mod q$ and verifies whether $T_s = h(p_{wa} \parallel r_a \parallel K_{aa} \parallel A)$. If the equation is true, $A$ computes $W_a = h(A \parallel B \parallel K_{aa} \parallel p_{wa} \parallel r_a)$ and sends $\{A, W_a\}$ to $S$.

Simultaneously, when receiving the message, $B$ also computes $K_{ab} = (N_a)^{y_1} \mod q$ and checks whether $T_s = h(p_{wb} \parallel r_b \parallel K_{ab} \parallel B)$. If the equation is true, $B$ computes $W_b = h(A \parallel B \parallel K_{ab} \parallel p_{wb} \parallel r_b)$ and sends $\{B, W_b\}$ to $S$.

4) Once receiving the messages, $S$ checks whether $W_a = h(A \parallel B \parallel K_{aa} \parallel p_{wa} \parallel r_a)$ and $W_b = h(A \parallel B \parallel K_{ab} \parallel p_{wb} \parallel r_b)$. If hold, $S$ computes $U_a = K_{aa} \oplus h(A \parallel B \parallel K_{aa} \parallel p_{wa} \parallel r_a)$. Finally, $S$ sends $U_a$ and respectively $U_b$ to $A$ and $B$.

5) After receiving the message, $A$ computes $K_{ab} = U_a \oplus h(A \parallel B \parallel K_{ab} \parallel p_{wa} \parallel r_a)$, $sk = (K_{ab})^{y_1} \mod q$, and $A_y = h(sk \parallel A \parallel B)$.
Then, A sends $A_u$ to B. Upon receiving the message, B retrieves $K_u$ by computing $U_0 \oplus h(A \parallel B \parallel K_{ab} \parallel pw_s \parallel r_z)$ and then computes $sk = (K_u)^y, A_b = h(sk \parallel A \parallel B)$.

Finally, B sends $A_u$ to A.

6) After receiving the message, A and B verify the correctness of $A_b$ and respectively $A_u$. If they respectively hold, A and B successfully agree on the session key $sk = (g)^y$ with the help of the server.

5. Security analysis

This section will present a cryptanalysis of the proposed protocol. The following attacks are based on the assumption that a malicious adversary has totally supervised the communication channel and has the capacity to intercept, insert, delete, refresh or update any information delivered in the public channel [23].

5.1. Off-line password guessing attack

Without loss of generality, we assume the adversary has intercepted the message \{H_s, V_s, R_s, A_T, B, N_z, W_s, U_s\} transmitted from A to S. However, the adversary cannot guess correctly of the password even if he knows the transmitted message. In the proposed protocol, A’s password $pw_s$ is mingled with the random number $r_z$, which is needed if the adversary intends to verify a guessed password. This random number cannot be obtained without knowledge of the password. Therefore, the proposed protocol is secure against the off-line password guessing attack.
5.2. On-line password guessing attack

Without loss of generality, an adversary may eavesdrop \( \{H_u, V_u, R_u, A, T_u, B, N, W, U_u\} \) and plan to pretend to be a legal user \( A \). But the adversary cannot send a new valid message \( \{H_u, V_u, R_u, A, T_u, B, N, W, U_u\} \) to the trusted server unless he has guessed the correct password. And all the values \( \{H_u, V_u, R_u, A, T_u, B, N, W, U_u\} \) are hidden from the one-way hash functions, the adversary cannot get them because of the security of hash function. Therefore, our protocol can resist the un-detectable online password guessing attack.

5.3. Mutual authentication

In the proposed protocol, \( S \) authenticates \( A \) by computing \( V_u \) and \( W_u \). Furthermore, \( S \) authenticates \( B \) by computing \( V_u \) and \( W_u \). \( S \) is verified through the correctness of \( T_u \) and \( T_b \). Therefore, mutual authentication is achieved in the proposed protocol.

5.4. Impersonation attack

Assume that the adversary wants to impersonate \( A \) to cheat \( S \) via intercepting the messages transmitted in the public channel. However, he cannot succeed until he can get the password of \( A \). As described in the previous subsection, the adversary is not possible to launch an off-line password guessing attack. That is, the proposed protocol is immune to the impersonation attack.

5.5. The session key perfect forward secrecy

Suppose \( A \) has compromised all the passwords of the communication entities. In order to get the session key \( s = (g)^{x+y} \), he needs to know the random numbers \( \{x, y, z\} \). Nevertheless, he will face the DLP if he tries to get \( \{x, y, z\} \) from \( \{K_u, K_v, N\} \). Therefore, the proposed protocol can ensure the perfect forward secrecy.

5.6. Replay attack

In our protocol, we use numerous random numbers for each session to resist replay attack. If the adversary intends to resend the old messages, the corresponding receiver will immediately detect the attack from the sender. Therefore, it seems to be impossible to perform the replay attack to our protocol.

6. Performance and security properties comparison

In this section, we compare the performance and security properties of our protocol with other related protocols [20-22] in Fig. 2 and Table 2.

In Fig. 2, our proposed protocol has a similar efficiency with Farash and Attari’s protocol even if the costs of our protocol are slightly higher than Tallapally and Huang’s protocol. However, Tallapally and Huang’s protocol is vulnerable to the on-line password guessing and off-line password guessing attacks and Huang’s protocol cannot provide mutual authentication. As shown in Table 2, none of the other protocols can resist the off-line password guessing attack. Therefore, our protocol is efficient and secure compared with other related protocols.

![Performance comparison](image)

No.of T: the number of trapdoor functions;
No.of H: the number of hashing operations;
No.of E: the number of exponentiation operations.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
<td>No</td>
</tr>
<tr>
<td>N2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>N3</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>N4</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>N5</td>
<td>Yes</td>
<td>No</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>N6</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

N1: Provide mutual authentication;
N2: Provide the session key perfect forward secrecy;
N3: Withstanding off-line password guessing attack;
N4: Withstanding on-line password guessing attack;
N5: Withstanding impersonation attack;
N6: Withstanding replay attack.

7. Conclusion

This paper mainly discussed the protocol proposed by Farash and Attari. We found that their protocol was insecure against the off-line password guessing attack, thus suffering from an impersonation attack. In order to eliminate the weaknesses of Farash and Attari’s protocol, we presented an enhanced three-party password-based authenticated key exchange protocol for wireless communications. We demonstrated that
the proposed protocol could withstand various kinds of attacks including attacks found in Farash and Attari’s protocol. In addition, we compared the proposed protocol with other related protocols regarding the performance and security features. The results showed that the proposed protocol was more secure than Farash and Attari’s protocol without increasing the computation cost. All in all, the proposed protocol is suitable for the wireless environments.

Acknowledgments

The authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper. This paper is supported by the National Natural Science Foundation of China (Grant Nos. 61472045, 61573067), the BUPT Excellent Ph.D. Students Foundation (Grant No. CX2015310), the Asia Foresight Program under NSFC Grant (Grant No. 61411146001) and the Beijing Natural Science Foundation (Grant No. 4142016).

References


Received January 2015.