ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2014, T. 43, Nr. 1

Security Analysis and Improvements of a Three-Party Password-Based Key Exchange Protocol

Hang Tu[1], Han Shen[2], Debiao He[2], Jianhua Chen[3]

[1]*School of Computer, Wuhan University, Wuhan, China*

[2,3]*School of Mathematics and Statistics, Wuhan University, Wuhan, China*
e-mail: [3]hedebiao@163.com

crossref http://dx.doi.org/10.5755/j01.itc.43.1.5322

Abstract. Recently Xie et al. [Q. Xie, N. Dong, X. Tan. D. Wong, G. Wang. Improvement of a three-party password-based key exchange protocol with formal verification. *Information Technology and Control*, 2013, Vol. 42, No. 3, 231-237] proposed an efficient three-party password-based key exchange protocol and used a formal verification tool to verify its security. In this paper, we demonstrate that their protocol is vulnerable to the off-line password guessing attack and the key compromise impersonation attack. The analysis shows that their protocol is not secure for practical applications. To overcome weaknesses in Xie et al.'s protocol, we also propose an improved 3PAKE protocol. Analysis shows that our protocol not only overcomes those weaknesses, but also has better performance. Therefore, our protocol is more suitable for practical applications.

Keywords: Key exchange protocol; Three-party; Password guessing attack; Key compromise impersonation attack; Denning-Sacco attack.

1. Introduction

The concept of the two-party password-based authenticated key exchange (2PAKE) protocol was first proposed by Bellovin and Merritt [1]. In such protocols, two parties could authenticate each other and generate a session key for future communications through a password shared between them. To ensure secure communication in the peer-to-peer system, a different password should be shared in each pair of communication parties. Then every party of the system has to maintain $k-1$ passwords if there are k parties in the system. Therefore, 2PAKE is not suitable for the large-scale peer-to-peer system. To solve the problems, many three-party password-based authenticated key exchange (3PAKE) protocols [2-12] were proposed during the last few years.

Lu and Cao [2] proposed an efficient 3PAKE protocol to improve performance in previous protocols. However, many researchers [3-8] demonstrated that Lu and Cao's protocol is vulnerable to the off-line password guessing attack and the man-in-the-middle attack. To overcome those weaknesses, Huang [9] proposed a new 3PAKE protocol. However, Yoon and Yoo [10] pointed out that Huang's protocol is vulnerable to the undetectable on-line password guessing attack and the off-line password guessing attack. In 2011, Lou and Huang [11] used elliptic curve cryptography to construct a new 3PAKE protocol for resource-constrained devices. Although their protocol has better performance, Xie et al. [12] found that their protocol is vulnerable to the off-line password guessing attack and the partition attack. Xie et al. also proposed an improved 3PAKE protocol to overcome weaknesses in Lou and Huang's protocol. Unfortunately, in this paper, we will demonstrate that Xie et al.'s 3PAKE protocol is vulnerable to the off-line password guessing attack and the key compromise impersonation attack. To overcome weaknesses in Xie et al.'s protocol, we also propose an improved 3PAKE protocol.

The organization of the paper is described as follows. Section 2 gives a brief review of Xie et al.'s 3PAKE protocol. Security analysis of their protocol is proposed in Section 3. Section 4 proposes our improved 3PAKE protocol. Security analysis and performance analysis of our protocol are proposed in Section 5 and Section 6 separately. At last, some conclusions are proposed in Section 7.

2. Review of Xie et al.'s 3PAKE protocol

In this section, we will give a brief review of Xie et al.'s 3PAKE protocol. For convenience, some notations are defined as follows.

- q, n: two large prime number;
- F_q: a finite field;
- $E(F_q)$: an elliptic curve over F_q;
- G: a cyclic additive group over $E(F_q)$ with order n;
- P: a generator of G;
- TS: the trusted server;
- A, B: two users;
- pw_A: the password shared between A and TS;
- pw_B: the password shared between B and TS;
- d: the secret key of TS;
- F: the public key of TS, where $F = dP$;
- $H(\cdot)$: a secure hash function, where $H(\cdot): \{0,1\}^* \rightarrow G$;
- $h(\cdot)$: a secure hash function, where $h(\cdot): \{0,1\}^* \rightarrow Z_n^*$;
- $\|$: the string concatenation operation;
- $\oplus$: the exclusive OR operation;

The trusted server (TS) chooses a large prime number q, an elliptic curve $E(F_q)$ defined over a finite field F_q, a cyclic group of points G over $E(F_q)$, a generator P of G and a secure hash function $H(\cdot)$, where $H(\cdot): \{0,1\}^* \rightarrow G$. TS also generates a random number d as his secret key and computes his public key $F = dP$. Let pw_A/pw_B be the password shared between the user A/B and TS. As shown in Fig. 1, the detail of Xie et al.'s 3PAKE protocol is described as follows.

1) A chooses a random number t_A, computes $Q_A = t_A P$, $F_A = t_A F$ and $Z_A = Q_A \oplus H(pw_A, A, B)$. Then A sends the message $\{A, Z_A, F_A\}$ to B.

2) Upon receiving $\{A, Z_A, F_A\}$, B chooses a random number t_B, computes $Q_B = t_B P$, $F_B = t_B F$ and $Z_B = Q_B \oplus H(pw_B, A, B)$. Then B sends the message $\{A, Z_A, F_A, B, Z_B, F_B\}$ to TS.

3) Upon receiving $\{A, Z_A, F_A, B, Z_B, F_B\}$, TS computes $Q_A = Z_A \oplus H(pw_A, A, B)$, $F_A' = dQ_A$, $Q_B = Z_B \oplus H(pw_B, A, B)$ and $F_B' = dQ_B$. TS checks whether both of the equations $F_A' = F_A$ and $F_B' = F_B$ hold. If either of them does not hold, TS stops the session; otherwise, TS chooses a random number t_{TS}, computes $R_A = t_{TS} Q_A \oplus H(pw_A, B, A)$,

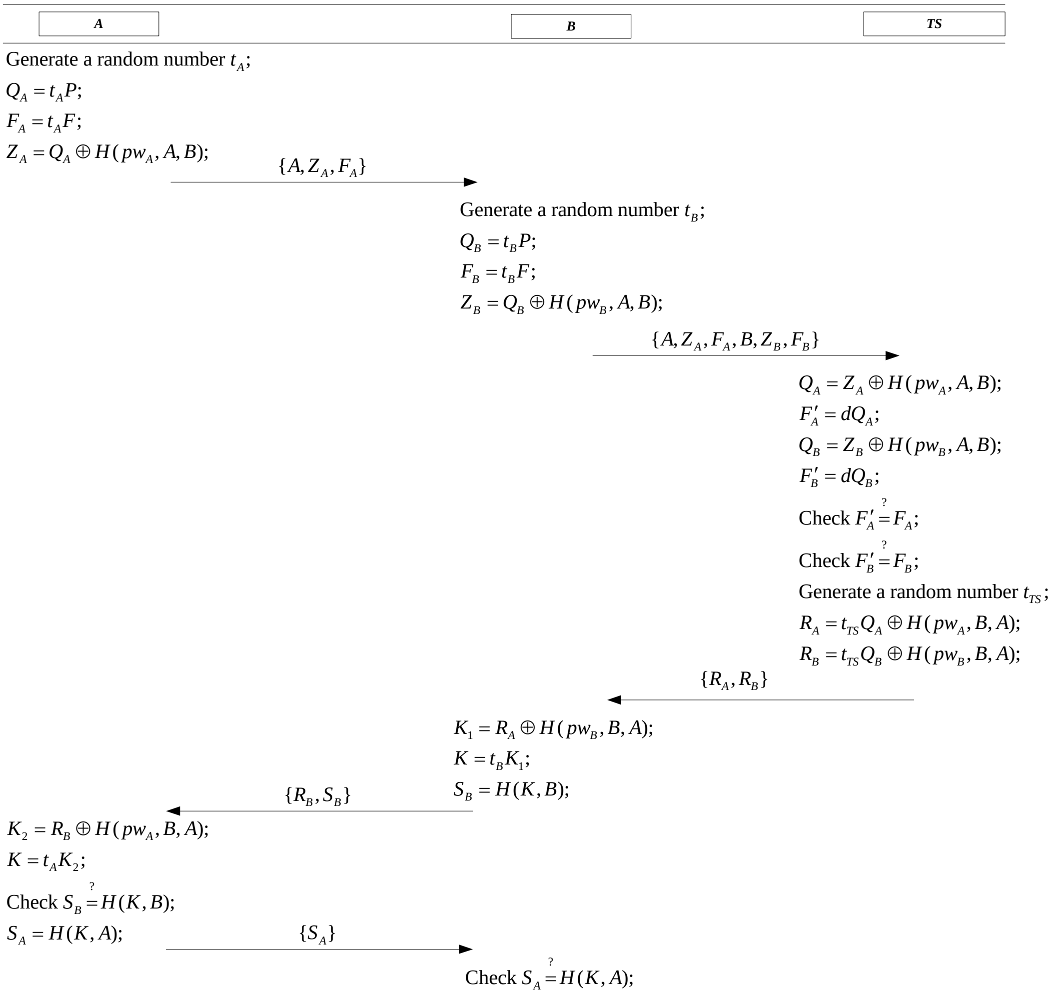

Figure 1. Xie et al.'s 3PAKE protocol

$R_B = t_{TS}Q_B \oplus H(pw_B, B, A)$ and sends the message $\{R_A, R_B\}$ to B. $\{R_A, R_B\}$

4) Upon receiving $\{R_A, R_B\}$, B computes $K_1 = R_A \oplus H(pw_B, B, A)$, $K = t_B K_1$ and $S_B = H(K, B)$. Then B sends the message $\{R_B, S_B\}$ to A.

5) Upon receiving $\{R_B, S_B\}$, A computes $K_2 = R_B \oplus H(pw_A, B, A)$, $K = t_A K_2$ and checks whether the equation $S_B = H(K, B)$ holds. If it does not hold, A stops the session; otherwise, A computes $S_A = H(K, A)$ and sends the message $\{S_A\}$ to B.

6) Upon receiving $\{S_A\}$, B checks whether the equation $S_A = H(K, A)$ holds. If it does not hold, B stops the session; otherwise, A and B generates the session key $K = t_A t_B t_{TS} P$.

3. Security analysis of Xie et al.'s 3PAKE protocol

Xie et al. claimed that their 3PAKE protocol could withstand various attacks. In this section, we will show their protocol is vulnerable to two kinds of attack in different subsections.

3.1. Off-line password guessing attack

For password-based protocols, the password guessing attack is very dangerous, since many users would like to choose simple and easy-to-remember password for their convenience. According to Ding and Horster's work, there are three kinds of password guessing attacks [13], i.e. the detectable on-line password guessing attack, the undetectable password guessing attack and off-line password guessing attack. The off-line password guessing attack is more dangerous than the other two attacks since there is no participation of the user or the server. Xie et al. claimed that their protocol could withstand the off-line password guessing attack. However, in this subsection, we will show that an adversary $\mathscr{A}$ could get the user's password through the off-line password guessing attack. Let the equation of the elliptic curve $E(F_q)$ be $y^2 = x^3 + ax + b$, where $a, b \in F_q$ and $4a^3 + 27b^2 \neq 0 \bmod q$. The detail of the attack is described as follows.

1) $\mathscr{A}$ intercepts the mesage$\{A, Z_A, F_A\}$ sent by A, where $Q_A = t_A P$, $F_A = t_A F$ and $Z_A = Q_A \oplus H(pw_A, A, B)$.

2) $\mathscr{A}$ chooses a possible password pw'_A from a dictionary D and computes $Z_A = Q_A \oplus H(pw'_A, A, B)$.

3) $\mathscr{A}$ checks whether the point Q'_A is a point on $E(F_q)$ by checking if the equation $y^2_{Q'_A} = x^3_{Q'_A} a x_{Q'_A} + b \bmod q$ holds, where $x_{Q'_A}$ and $y_{Q'_A}$ are the x-coordinate and the y-coordinate of Q'_A respectively. If Q'_A is a point on $E(F_q)$, $\mathscr{A}$ finds the correct password; otherwise, $\mathscr{A}$ repeats 2) and 3) until the correct password is found.

Since $h(\cdot)$ is a secure hash function, we could get that the computational result $Q'_A = Z_A \oplus H(pw'_A, A, B)$ is a random number pair $\left(x_{Q'_A}, y_{Q'_A}\right)$ if pw'_A is not the correct password. Let n be the order of the group G. Then the probability that the point $\left(x_{Q'_A}, y_{Q'_A}\right)$ lies on $E(F_q)$ is no larger than $\frac{2}{n}$ [14]. Therefore, the adversary could find the correct password pw'_A using the above-described attack with a probability of $\left(1 - \frac{1}{n}\right)^{|D|-1} \approx 1 - \frac{2(|D|-1)}{n} \approx 1$ since the size of the dictionary D could be ignored compared with the order of G. Therefore, we could conclude that Xie et al.'s 3PAKE protocol is vulnerable to the off-line password guessing attack.

3.2. Key compromise impersonation attack

As a key exchange protocol, the 3PAKE protocol should provide the known-key security, the perfect forward secrecy, the key compromise impersonation resilience, the unknown key share resilience and the no key control. However, we find that Xie et al.'s 3PAKE protocol cannot provide the key compromise impersonation resilience, i.e. it is vulnerable to the key compromise impersonation attack. In the 3PAKE protocol, the key compromise impersonation resilience means that any adversary $\mathscr{A}$ cannot impersonate another user B or the trusted server TS to the user A when he gets A's password. Suppose that $\mathscr{A}$ gets A's password pw_A, then he could impersonate B and TS to A through the following steps.

1) A chooses a random number t_A, computes $Q_A = t_A P$, $F_A = t_A F$ and $Z_A = Q_A \oplus H(pw_A, A, B)$. Then A sends the message $\{A, Z_A, F_A\}$ to B.

2) $\mathscr{A}$ intercepts the message $\{A, Z_A, F_A\}$ and computes $Q_A = Z_A \oplus H(pw_A, A, B)$.

3) $\mathscr{A}$ chooses two random numbers t_B, t_{TS} and computes $Q_B = t_B P$, $R_A = t_{TS} Q_A$, $R_B = t_{TS} Q_B \oplus H(pw_A, B, A)$, $K_1 = R_A$, $K = t_B K_1$ and $S_B = H(K, B)$. Then $\mathscr{A}$ sends the message $\{R_B, S_B\}$ to A.

4) Upon receiving $\{R_B, S_B\}$, A computes $K_2 = R_B \oplus H(pw_A, B, A)$, $K = t_A K_2$ and checks whether the equation $S_B = H(K, B)$ holds. It is easy to see that the equation holds. Then A computes $S_A = H(K, A)$ and sends the message $\{S_A\}$ to B.

From the above description, we know that A confirms the message $\{R_B, S_B\}$ is sent by B. Then $\mathscr{A}$ impersonates B and TS to A successfully. Therefore, Xie et al.'s 3PAKE protocol is vulnerable to the key compromise impersonation attack.

4. Our improved 3PAKE protocol

To overcome weaknesses in Xie et al.'s 3PAKE protocol, we proposed an improved 3PAKE protocol in this section.

The trusted server (TS) chooses a large prime number q, an elliptic curve $E(F_q)$ defined over a finite field F_q, a cyclic group of points G over $E(F_q)$, a generator P of G and a secure hash functions $h(\cdot)$, where $h(\cdot): \{0,1\}^* \rightarrow Z_n^*$. TS also generates a random number d as his secret key and computes his public key $F = dP$. Let pw_A/pw_B be the password shared between the user A/B and TS. As shown in Fig. 2, the detail of our improved 3PAKE protocol is described as follows.

1) A chooses a random number t_A, computes $Q_A = t_A P$, $F_A = t_A F$ and $Z_A = h(pw_A, A, B, Q_A, F_A)$. Then A sends the message $\{A, Q_A, Z_A\}$ to B.

2) Upon receiving $\{A, Z_A, F_A\}$, B chooses a random number t_B, computes $Q_B = t_B P$, $F_B = t_B F$ and $Z_B = h(pw_B, A, B, Q_B, F_B)$. Then B sends the message $\{A, Q_A, Z_A, B, Q_B, Z_B\}$ to TS.

3) Upon receiving $\{A, Q_A, F_A, B, Z_B, F_B\}$, TS computes $F'_A = dQ_A$, and $F'_B = dQ_B$. TS checks whether both of the equations $Z_A = h(pw_A, A, B, Q_A, F'_A)$ and $Z_B = h(pw_B, A, B, Q_B, F'_B)$ hold. If either of them does not hold, TS stops the session; otherwise, TS computes $R_A = h(pw_A, A, B, Q_A, F'_A, Q_B)$, $R_B = h(pw_B, A, B, Q_B, F'_B, Q_A)$ and sends the message $\{R_A, R_B\}$ to B.

4) Upon receiving $\{R_A, R_B\}$, B checks whether the equation $R_B = h(pw_B, A, B, Q_B, F'_B, Q_A)$ holds. If it does not hold, B stops the session; otherwise, B computes $K = t_B Q_A = t_A t_B P$ and $S_B = h(K, B)$. Then B sends the message $\{R_A, Q_B, S_B\}$ to A.

5) Upon receiving $\{R_A, Q_B, S_B\}$, A checks whether the equation $R_A = h(pw_A, A, B, Q_A, F_A, Q_B)$ holds. If it does not hold, A stops the session; otherwise, A computes $K = t_A Q_B = t_A t_B P$ and checks whether the equation $S_B = h(K, B)$ holds. If it does not hold, A stops the session; otherwise, A computes $S_A = h(K, A)$ and sends the message $\{S_A\}$ to B.

6) Upon receiving $\{S_A\}$, B checks whether the equation $S_A = h(K, A)$ holds. If it does not hold, B stops the session; otherwise, A and B generate the session key $K = t_A t_B P$.

5. Security analysis

In this section, we will analyze the security of our 3PAKE protocol. We will show that our protocol could provide perfect forward secrecy and mutual

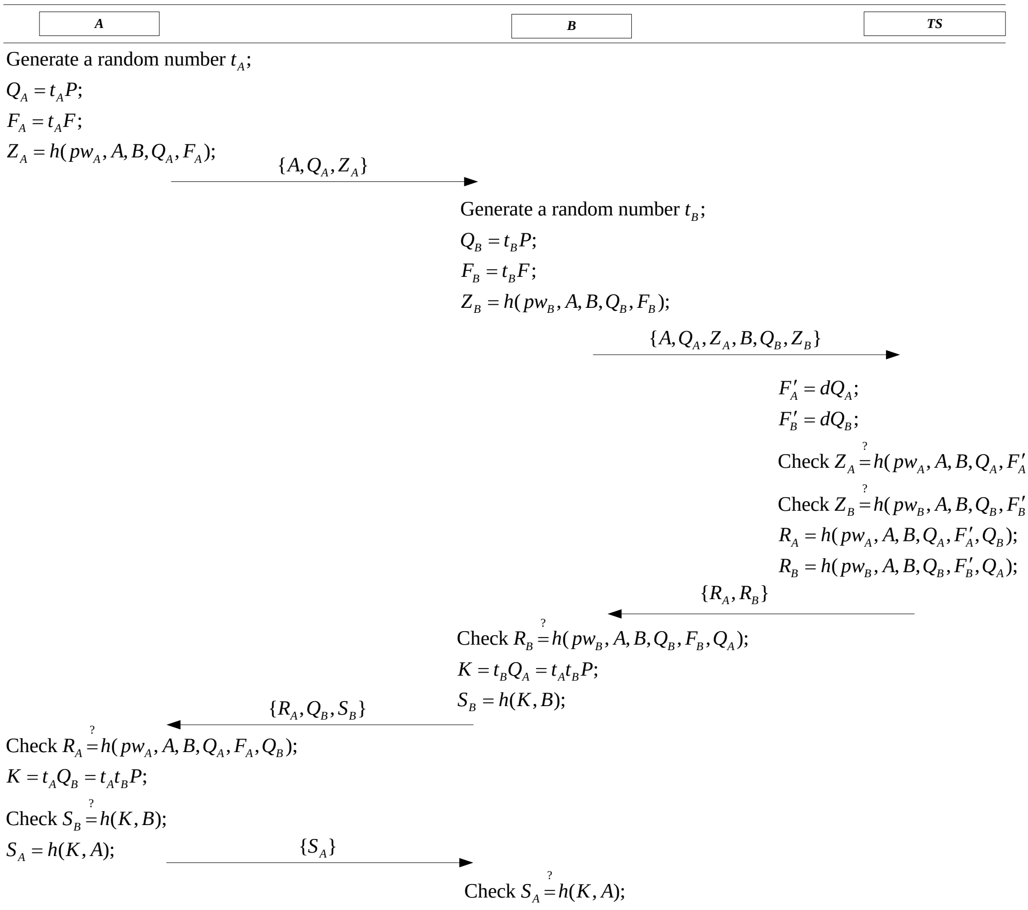

Figure 2. Our improved 3PAKE protocol

authentication. We will also show that our protocol could withstand the password guessing, the key compromise impersonation attack, the man-in-the-middle attack, the replay attack, the Denning-Sacco attack, the impersonation attack and the server spoofing attack.

5.1. Perfect forward secrecy

Suppose an adversary $\mathscr{A}$ could get A's password pw_A, B's password pw_B and TS's secret key d. We also assume that $\mathscr{A}$ could intercept the message $\{A, Q_A, Z_A\}$, $\{A, Q_A, Z_A, B, Q_B, Z_B\}$, $\{R_A, R_B\}$, $\{R_A, Q_B, S_B\}$ and $\{S_A\}$ transmitted among A, B and TS, where $Q_A = t_A P$, $F_A = t_A F$, $Z_A = h(pw_A, A, B, Q_A, F_A)$, $Q_B = t_B P$, $F_B = t_B F$, $Z_B = h\,(pw_B, A, B, Q_B, F_B)$, $R_A = h(pw_A, A, B, Q_A, F_A, Q_B)$, $R_B = h(pw_B, A, B, Q_B, F_B, Q_A)$, $K = t_A t_B P$, $S_A = h(K, A)$ and $S_B = h(K, B)$. $\mathscr{A}$ could compute F_A and F_B from Q_A and Q_B. However, he cannot compute $K = t_A t_B P$ from Q_A and Q_B since he will face the computational Diffie-Hellman problem. Therefore, our 3PAKE protocol could provide perfect forward secrecy.

5.2. Mutual authentication

Without A/B's password pw_A/pw_B, any adversary cannot generate a legal $Z_A = h(pw_A, A, B, Q_A, F_A)/Z_B = h\,(pw_B, A, B, Q_B, F_B)$. Then TS could authenticate A/B by checking the correctness of Z_A/Z_B. Without A/B's password pw_A/pw_B and TS's secret key d, any adversary cannot generate a legal $R_A = h(pw_A, A, B, Q_A, F_A, Q_B)/R_B = h(pw_B, A, B, Q_B, F_B, Q_A)$. Then A/B could authenticate TS by checking the correctness of R_A/R_B. Besides, A and B could authenticate each other by checking correctness of S_B and S_A separately. Therefore, our 3PAKE protocol could provide mutual authentication among A, B and TS.

5.3. Password guessing attack

It is easy to withstand the detectable on-line password guessing attack and the undetectable password guessing attack by limiting the login time in some period. Then we just need to show that our protocol could withstand the off-line password guessing attack. The information of A's password is included in Z_A and R_A, where $Z_A = h(pw_A, A, B, Q_A, F_A)$ and $R_A = h(pw_A, A, B, Q_A, F'_A, Q_B)$. $\mathscr{A}$ could guess a password pw'_A from a dictionary. However, he cannot verify its correctness since he cannot compute F_A without TS's secret key. Then $\mathscr{A}$ cannot get A's password through the off-line password guessing attack. Through a similar method, we could show that $\mathscr{A}$ cannot get B's password through the off-line password guessing attack. Therefore, our protocol could withstand the password guessing attack.

5.4. Key compromise impersonation attack

Suppose an adversary $\mathscr{A}$ could get A's password pw_A and intercept the message $\{A, Q_A, Z_A\}$ sent by A, where $Q_A = t_A P$, $F_A = t_A F$ and $Z_A = h(pw_A, A, B, Q_A, F_A)$. To impersonate B and TS to A, $\mathscr{A}$ has to generate a legal message $\{R_A, Q_B, S_B\}$, where $R_A = h(pw_A, A, B, Q_A, F'_A, Q_B)$, $Q_B = t_B P$, $K = t_B Q_A = t_A t_B P$ and $S_B = h(K, B)$. However, $\mathscr{A}$ cannot compute correct R_A since he cannot compute F_A without TS's secret key. Therefore, $\mathscr{A}$ cannot impersonate B and TS to A and our protocol could withstand the key compromise impersonation attack.

5.5. Man-in-the-middle attack

From the above description, we know that our 3PAKE protocol could provide mutual authentication among A, B and TS. Therefore, our 3PAKE protocol could withstand the main-in-the-middle attack.

5.6. Replay attack

Suppose that an adversary could intercept the message $\{A, Q_A, Z_A\}$ and replay it to B, where $Q_A = t_A P$, $F_A = t_A F$ and $Z_A = h(pw_A, A, B, Q_A, F_A)$. However, he cannot generate a legal message $\{S_A\}$ since he does not know t_A, where $S_A = h(K, A)$ and $K = t_A Q_B$. Then B could disclose the attack by checking the correctness of S_A. Through a similar method, we could show that A and TS also could detect the replay attack. Therefore, our 3PAKE protocol could withstand the replay attack.

5.7. Denning-Sacco attack

Suppose that an adversary $\mathscr{A}$ could get the session key $K = t_A t_B P$ and intercepts the message $\{A, Q_A, Z_A\}$, $\{A, Q_A, Z_A, B, Q_B, Z_B\}$, $\{R_A, R_B\}$, $\{R_A, Q_B, S_B\}$ and $\{S_A\}$ transmitted among A, B and TS, where $Q_A = t_A P$, $F_A = t_A F$, $Z_A = h(pw_A, A, B, Q_A, F_A)$, $Q_B = t_B P$, $F_B = t_B F$, $S_A = h(K, A)$, $S_B = h(K, B)$, $Z_B = h\,(pw_B, A, B, Q_B, F_B)$. $R_A = h(pw_A, A, B, Q_A, F_A, Q_B)$, $R_B = h(pw_B, A, B, Q_B, F_B, Q_A)$. However, he still cannot get F_A and F_B since he does not possess TS's secret key d. Then he cannot get A/B's password pw_A/pw_B. Therefore, our 3PAKE protocol could withstand the Denning-Sacco attack.

5.8. Impersonation attack

To impersonate A to B and TS, the adversary $\mathscr{A}$ has to generate a legal message $\{A, Q_A, Z_A\}$, where $Z_A = h(pw_A, A, B, Q_A, F_A)$, $Q_A = t_A P$, and $F_A = t_A F$. $\mathscr{A}$ could generate a random number t_A and compute $Q_A = t_A P$, $F_A = t_A F$. However, he cannot compute $Z_A = h(pw_A, A, B, Q_A, F_A)$ since he does not have A's password pw_A. Arguing analogously, we could show that $\mathscr{A}$ cannot impersonate B to A and TS. Therefore, our 3PAKE protocol could withstand the impersonation attack.

5.9. Server spoofing attack

To impersonate TS to A, the adversary $\mathscr{A}$ has to generate a legal message $R_A = h(pw_A, A, B, Q_A, F_A, Q_B)$ when he receives the message $\{A, Q_A, Z_A\}$, where $Z_A = h(pw_A, A, B, Q_A, F_A)$, $Q_A = t_A P$, and $F_A = t_A F$. However, he cannot compute F_A from Q_A since he does not TS's secret key d. Then $\mathscr{A}$ cannot generate R_A and impersonate TS to A. Using a similar method, we can show that $\mathscr{A}$ cannot impersonate TS to B. Therefore, our 3PAKE protocol could withstand the server spoofing attack.

6. Performance analysis

In this section, we will analyze the computational cost and communicational cost of our 3PAKE protocol. We also compare the performance of our protocol with Lou and Huang's 3PAKE protocol [11] and Xie et al.'s 3PAKE protocol [12]. For convenience, some notations are defined as follows.

- T_{SM}: the running time of a scalar multiplication operation;
- T_{MH}: the running time of a map-to-point hash function operation;
- T_H: the time of executing a general hash function operation;

It is well known that the running time of a scalar multiplication operation is more time-consuming than other operations. Many implementations of those operations have been reported. In Scott et al.'s [15], a supersingular curve or non-supersingular curve $E(F_q)$ over a finite field F_q is chosen, where the length of big number q and the order of $E(F_q)$ is 512bits and 160 bits, respectively. They evaluate the running time using a Pentium IV processor with 512MB RAMS. Besides, the machine under Windows XP offers a maximum clock speed of 3 GHz. The implement results are listed in *Table 1* [15].

Table 1. Running time of different operations

T_{SM}	T_{MH}	T_H
$1.17ms$	$\approx 1.00ms$	$\approx 0.01ms$

In Table 2, we list comparisons among, our 3PAKE protocol, Lou and Huang's 3PAKE protocol [11] and Xie et al.'s 3PAKE protocol [12] in terms of computational cost, where the execution times are measured using Table 1. Our 3PAKE protocol has better performance than Lou and Huang's 3PAKE protocol at the trusted server side. Lou and Huang's 3PAKE protocol has better performance at the user side. Lou and Huang's 3PAKE protocol is vulnerable to the off-line password guessing attack and the partition attack. Xie et al.'s 3PAKE protocol cannot withstand the off-line password guessing attack and the key compromise impersonation attack. Analysis shows that our 3PAKE protocol could overcome weaknesses and has better performance than Xie et al.'s 3PAKE protocol. Therefore, we can conclude that our protocol is more suitable for practical applications.

Table 2. Comparison of computational costs

	A	B	TS
Lou and Huang's 3PAKE protocol [11]	$3T_{SM} + 3T_H$ $\approx 3.54ms$	$3T_{SM} + 3T_H$ $\approx 3.54ms$	$4T_{SM} + 3T_H$ $\approx 4.71ms$
Xie et al.'s 3PAKE protocol [12]	$3T_{SM} + 4T_{MH}$ $\approx 7.51ms$	$3T_{SM} + 4T_{MH}$ $\approx 7.51ms$	$4T_{SM} + 4T_{MH}$ $\approx 8.68ms$
Our 3PAKE protocol	$3T_{SM} + 4T_H$ $\approx 3.55ms$	$3T_{SM} + 4T_H$ $\approx 3.55ms$	$2T_{SM} + 4T_H$ $\approx 2.38ms$

7. Conclusion

In this paper, we give some analysis about the security of the Xie et al.'s 3PAKE protocol. We point out that their protocol is vulnerable to dangerous attacks. To overcome those weaknesses, we also propose an improved 3PAKE protocol. Analysis shows that our improved protocol not only overcomes those weaknesses, but also has better performance. Therefore, our protocol is more suitable for practical applications.

References

[1] **S. M. Bellovin, M. Merritt**. Encrypted key exchange: password based protocols secure against dictionary attacks. In: *Proceedings of IEEE Symposium on Research in Security and Privacy*, 1992, pp. 72–84.

[2] **R. X. Lu, Z. F. Cao**. Simple three-party key exchange protocol. *Computers and Security*, 2007, Vol. 26, 94–97.

[3] **H. Guo, Z. J. Li, Y. Mu, X. Y. Zhang.** Cryptanalysis of simple three-party key exchange protocol. *Computers and Security*, 2008, Vol. 27, 16-21.

[4] **Y. F. Chang**. A practical three-party key exchange protocol with round efficiency. *International Journal of Innovative Computing, Information and Control*, 2008, Vol. 4, 953-960.

[5] **H. R. Chung, W. C. Ku**. Three weaknesses in a simple three-party key exchange protocol. *Information Sciences*, 2008, Vol. 178, 220-229.

[6] **R. C. W. Phan, W. C. Yau, B. M. Goi**. Cryptanalysis of simple three-party key exchange protocol (S-3PAKE). *Information Sciences*, 2008, Vol. 178, 2849-2856.

[7] **J. Y. Nam, J. Y. Paik, H. K. Kang, U. M. Kim, D. H. Won**. An off-line dictionary attack on a

simple three-party key exchange protocol. *IEEE Communication Letters*, 2009, Vol. 13, 205-207.

[8] **H. S. Kim, J. Y. Choi**. Enhanced password-based simple three-party key exchange protocol. *Computers and Electrical Engineering*, 2009, Vol. 35, 107-114.

[9] **H. F. Huang**. A simple three-party password-based key exchange protocol. *International Journal of Communication Systems*, 2009, Vol. 22, 857-862.

[10] **E. J. Yoon, K. Y. Yoo**. Cryptanalysis of a simple three-party password-based key exchange protocol. *International Journal of Communication Systems*, 2011, Vol. 24, 532-542.

[11] **D. C. Lou, H. F. Huang**. Efficient three-party password-based key exchange scheme. *International Journal of Communication Systems*, 2011, Vol. 24, 504-512.

[12] Q. Xie, N. Dong, X. Tan. D. S. Wong, G. Wang. Improvement of a three-party password-based key exchange protocol with formal verification. *Information Technology and Control*, 2013, Vol. 42, No. 3, 231-237.

[13] Y. Ding, P. Horster. Undetectable on-line password guessing attacks. *ACM Operating Systems Review*, 1995, Vol. 29, No. 4, 77-86.

[14] **D. He, S. Wu**. Security flaws in a smart card based authentication scheme for multi-server environment. *Wireless Personal Communications*, 2013, Vol. 70, No. 1, 323-329.

[15] M. Scott, N. Costigan, W. Abdulwahab. Implementing cryptographic pairings on smartcards. In: *Cryptographic Hardware and Embedded Systems – CHES 2006*, LNCS, 2006, Vol. 4249, pp. 134–147.

Received September 2013.