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

Qingping Wang, Ou Ruan(✉), and Zihao Wang

School of Computer Science, Hubei University of Technology, Wuhan, China
ruanou@163.com

Abstract. Three-party password-based authenticated key exchange (3PAKE) protocol allows two clients, each sharing a password with a trusted server, to establish a secret session key with the help of the server. It is a practical mechanism for establishing secure channels in the communication networks. Recently, Xu et al. proposed a 3PAKE protocol without the server's public key. They claimed that their protocol could withstand various attacks. In this paper, we show Xu et al.'s protocol is insecure against the stolen-verifier attack. Furthermore, we propose an improved 3PAKE protocol to overcome the weakness of Xu et al.'s protocol. Security and performance analysis shows that our protocol not only overcomes the security weakness, but also is more efficient. Therefore, our protocol is more suitable for the practical applications.

1 Introduction

Password-based authenticated key exchange (PAKE) protocols allow two or more specified parties to authenticate each other and establish a high-entropy secret session key by using only the weak, low-entropy and easily memorable passwords. This authenticated key exchange scheme is the most widely used in practice because no additional devices such as smart cards or hardware tokens is needed, but just a human-memorable password for authenticating the parties.

Bellovin and Merritt [1] first proposed a two-party PAKE protocol in 1992. The protocol allowed two parties to authenticate each other via a public, insecure network and establish a secure session key which is to be used for protecting their subsequent communication. Then, many efficient and practical PAKE protocols [2–6] have been proposed. The above two-party protocols were not scalable in a large-scale peer-to-peer system, since every pair of communication parties needs to share a password, so that each party in an n-party system has to maintain $n-1$ passwords [7]. To solve this problem, Three-party password-based authenticated key exchange (3PAKE) protocols were introduced [8–15]. However, these 3PAKE protocols still existed some security problems such as on-line undetectable password guessing attack [16] and off-line password guessing attack [10].

In order to increase the efficiency and preventing various attacks, in 2005 Lee et al. [17] proposed an efficient verifier-based key agreement protocol for three parties

L. Barolli et al. (eds.), *Advances in Internetworking, Data & Web Technologies*,
Lecture Notes on Data Engineering and Communications Technologies 6,
DOI 10.1007/978-3-319-59463-7_49

without server's public key. Lee et al. claimed the proposed protocol could resist various attacks and provide the perfect forward secrecy. Wang et al. [18] pointed out that it would be more dangerous when suffers from the impersonation attack in 2006. After the defects of Lee-3PAKE protocol are discovered, there are a lot of improved protocols, which are based on the three-party authenticated key exchange protocol. Kwon J O et al. [19] designed a secure three-party password authentication key agreement protocol, but the communication cost and computation cost of the protocol were larger than [17]. Li et al. [20] proposed an efficient three-party password-based authenticated key exchange protocol based on bilinear pairings. Recently, Xu et al. [21] proposed an efficient 3PAKE according to the Lee-3PAKA protocol, combined with symmetric encryption.

In this paper, we show that Xu et al.'s scheme is vulnerable to the stolen-verifier attack. In addition, we propose an improved scheme to solve this problem. The protocol also enjoys low computational complexity and is suitable for resource-constrained devices.

The rest of this paper is organized as follows. In Sect. 2, we review Xu et al.'s scheme. In Sect. 3, a stolen-verifier attack against their scheme is described in details. In Sect. 4, we propose an improved scheme and the security and performance analyses are discussed in Sect. 5. The paper is concluded in Sect. 6.

2 Review of Xu et al.'s Protocol

This section revisits the 3PAKE protocol proposed by Xu et al. [21].

2.1 Notations

The notations used throughout this paper are summarized in Table 1.

Table 1. Notations for the proposed protocols

Notation	Description
A	Alice's public identity
B	Bob's public identity
S	Authentication Server's public identity
M	The attacker
pw	A weak password
E	Symmetric encryption
D	Symmetric decryption
a, b, c, d	Session-independent random numbers
p	A large prime
g	A generator g in the cyclic group Z_p^*
$H(\cdot)$	A collision-resistant one-way hash function
$\oplus$	Bit-wise exclusive-OR (XOR) operation
K	A session key
V	Verifier computed from a password
c^{-1}	Inverse of c on Z_p^*

2.2 Protocol Description

For a detailed analysis, we review Xu et al.'s 3PAKE protocol [21]. The details of this protocol, shown in Fig. 1, are as follows:

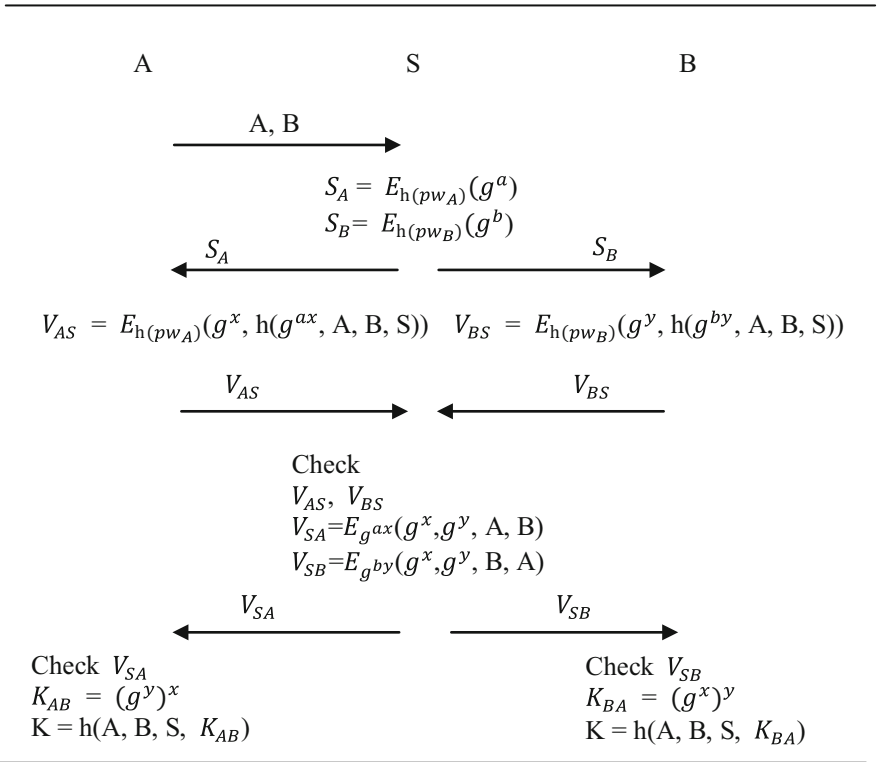

Fig. 1. Authentication and key exchange phase of Xu et al.'s protocol

Before the running of the protocol, Alice and Bob sends their verifiers $h(pw_A)$ and $h(pw_B)$ to S through a secure channel. S stores $h(pw_A)$ and $h(pw_B)$ in a password table.

Round 1: User A sends A and B to S.

$$A \rightarrow S : A, B.$$

Round 2: After receiving the messages sent by A, S randomly chooses a and b, computes $S_A = E_{h(pw_A)}(g^a)$, $S_B = E_{h(pw_B)}(g^b)$ and then sends S_A and S_B to A and B, respectively.

$$S : a, b \in_R Z_p^*$$
$$S : S_A = E_{h(pw_A)}(g^a).$$
$$S : S_B = E_{h(pw_B)}(g^b).$$
$$S \rightarrow A : S_A.$$
$$S \rightarrow B : S_B.$$

Round 3: After receiving the message sent by S, A computes $g^a = D_{\mathrm{h}(pw_A)}(S_A)$ and $V_{AS} = E_{\mathrm{h}(pw_A)}(g^x, \mathrm{h}(g^{ax}, \mathrm{A}, \mathrm{B}, \mathrm{S}))$ by choosing x $\in_R Z_p^*$, and sends V_{AS} to S. Similarly, after receiving the message from S, B computes $g^b = D_{\mathrm{h}(pw_B)}(S_B)$ and $V_{BS} = E_{\mathrm{h}(pw_B)}(g^y, \mathrm{h}(g^{by}, \mathrm{A}, \mathrm{B}, \mathrm{S}))$ by choosing y $\in_R Z_p^*$, and sends V_{BS} to S.

$$
\begin{aligned}
&\mathrm{A} : g^a = D_{\mathrm{h}(pw_A)}(S_A).\\
&\mathrm{A} : \ \mathrm{x} \in_R Z_p^*.\\
&\mathrm{A} : V_{AS} = E_{\mathrm{h}(pw_A)}\,(g^x,\ \mathrm{h}(g^{ax}, \mathrm{A}, \mathrm{B}, \mathrm{S})).\\
&\mathrm{A} \rightarrow \mathrm{S} : V_{AS}\\
&\mathrm{B} : g^b = D_{\mathrm{h}(pw_B)}(S_B).\\
&\mathrm{B} : \mathrm{y} \in_R Z_p^*.\\
&\mathrm{B} : V_{BS} = E_{\mathrm{h}(pw_B)}\,(g^y,\ \mathrm{h}(g^{by}, \mathrm{A}, \mathrm{B}, \mathrm{S})).\\
&\mathrm{B} \rightarrow \mathrm{S} : V_{BS}
\end{aligned}
$$

Round 4: After receiving the messages sent by A and B, S checks whether $V_{AS} = E_{\mathrm{h}(pw_A)}(g^x, \mathrm{h}(g^{ax}, \mathrm{A}, \mathrm{B}, \mathrm{S}))$ and $V_{BS} = E_{\mathrm{h}(pw_B)}(g^y, \mathrm{h}(g^{by}, \mathrm{A}, \mathrm{B}, \mathrm{S}))$ hold or not. If it holds, S computes $V_{SA} = E_{g^{ax}}(g^x, g^y, \mathrm{A}, \mathrm{B})$ and $V_{SB} = E_{g^{by}}(g^x, g^y, \mathrm{B}, \mathrm{A})$ and sends V_{SA} and V_{SB} to A and B, respectively. Otherwise S aborts the protocol.

$$
\begin{aligned}
&\mathrm{S} : \mathrm{Checks}\\
&V_{AS} = E_{\mathrm{h}(pw_A)}(g^x, \mathrm{h}(g^{ax}, \mathrm{A}, \mathrm{B}, \mathrm{S})).\\
&V_{BS} = E_{\mathrm{h}(pw_B)}(g^y, \mathrm{h}(g^{ay}, \mathrm{A}, \mathrm{B}, \mathrm{S})).\\
&\mathrm{S} : V_{SA} = E_{g^{ax}}(g^x, g^y, \mathrm{A}, \mathrm{B}).\\
&\mathrm{S} : V_{SB} = E_{g^{by}}(g^x, g^y, \mathrm{B}, \mathrm{A}).\\
&\mathrm{S} \rightarrow \mathrm{A} : V_{SA}.\\
&\mathrm{S} \rightarrow \mathrm{B} : V_{SB}.
\end{aligned}
$$

Finally: After receiving the message sent by S, A checks whether $g^x \in (g^x, g^y, \mathrm{A}, \mathrm{B})$ hold or not, If it holds, A computes $K_{AB} = (g^y)^x$. Otherwise A aborts the protocol. Similarly, after receiving the message sent by S, B checks whether $g^y \in (g^x, g^y, \mathrm{B}, \mathrm{A})$ hold or not, If it holds, B computes $K_{BA} = (g^x)^y$. Otherwise B aborts the protocol. Finally, A and B compute a common session key K = h(A, B, S, K_{AB}) = h(A, B, S, K_{BA}) = h(A, B, S, g^{xy}), respectively.

$$
\begin{aligned}
&\mathrm{A}: \text{Checks} \\
&\qquad g^x \in (g^x, g^y, \mathrm{A}, \mathrm{B}). \\
&\mathrm{A}: K_{AB} = (g^y)^x. \\
&\mathrm{A}: \mathrm{K} = \mathrm{h}(\mathrm{A}, \mathrm{B}, \mathrm{S}, K_{AB}). \\
&\mathrm{B}: \text{Checks} \\
&\qquad g^y \in (g^x, g^y, \mathrm{B}, \mathrm{A}). \\
&\mathrm{B}: K_{BA} = (g^x)^y \\
&\mathrm{B}: \mathrm{K} = \mathrm{h}(\mathrm{A}, \mathrm{B}, \mathrm{S}, K_{BA})
\end{aligned}
$$

3 Attacks on Xu et al.'s 3PAKE Protocol

In this section, we show that Xu et al.'s 3PAKE is vulnerable to stolen-verifier attack.

Through the security analysis of the Xu-3PAKE protocol, the author points out that the protocol provides forward security and resist man-in-the-middle attack, Denning-Sacco attack, password guessing attack, stolen-verifier attack and replay attack. Among them, the author claims that the protocol cannot be directly impersonate the user when the adversary obtains the authentication value of a user's password on the server, but in fact it still cannot resist the attack of the stolen-verifier.

According to the security model proposed by Dolev and Yao [23], an active attacker can control the communication channels through intercepting the communication and inserting data into the channels. Below are the details of our attacks.

The attack of the stolen-verifier:

We assume that M is an attacker who has got A's verifier V_A. M can impersonate A to communicate with B by performing the following steps.

Round 1: Like the normal interaction, M sends S the message(A, B).

$$\mathrm{M} \rightarrow \mathrm{S}: \mathrm{A}, \mathrm{B}.$$

Round 2: After receiving the messages sent by M, S randomly chooses a and b, computes $S_A = E_{\mathrm{h}(pw_A)}(g^a)$, $S_B = E_{\mathrm{h}(pw_B)}(g^b)$ and then sends S_A and S_B to A and B, respectively. But S_A is intercepted by M.

$$
\begin{aligned}
&\mathrm{S}: \mathrm{a}, \mathrm{b} \in_R Z_p^*. \\
&\mathrm{S}: S_A = E_{\mathrm{h}(pw_A)}(g^a). \\
&\mathrm{S}: S_B = E_{\mathrm{h}(pw_B)}(g^b). \\
&\mathrm{S} \rightarrow \mathrm{M}: S_A. \\
&\mathrm{S} \rightarrow \mathrm{B}: S_B.
\end{aligned}
$$

Round 3: After intercepting the message in Round 2, M computes $g^a = D_{\mathrm{h}(pw_A)}(S_A)$ and $V_{AS} = E_{\mathrm{h}(pw_A)}(g^x, \mathrm{h}(g^{ax}, \mathrm{A}, \mathrm{B}, \mathrm{S}))$ by choosing x $\in_R Z_p^*$, and sends V_{AS} to S. After receiving the message from S, B computes $g^b = D_{\mathrm{h}(pw_B)}(S_B)$ and $V_{BS} = E_{\mathrm{h}(pw_B)}(g^y, \mathrm{h}(g^{by}, \mathrm{A}, \mathrm{B}, \mathrm{S}))$ by choosing y $\in_R Z_p^*$, and sends V_{BS} to S.

$$\begin{aligned}
&\mathrm{M}: g^a = D_{\mathrm{h}(pw_A)}(S_A).\\
&\mathrm{M}: \mathrm{x} \in_R Z_p^*.\\
&\mathrm{M}: V_{AS} = E_{\mathrm{h}(pw_A)}(g^x, \mathrm{h}(g^{ax}, \mathrm{A}, \mathrm{B}, \mathrm{S})).\\
&\mathrm{M} \rightarrow \mathrm{S}: V_{AS}.\\
&\mathrm{B}: g^b = D_{\mathrm{h}(pw_B)}(S_B).\\
&\mathrm{B}: \mathrm{y} \in_R Z_p^*.\\
&\mathrm{B}: V_{BS} = E_{\mathrm{h}(pw_B)}(g^y, \mathrm{h}(g^{by}, \mathrm{A}, \mathrm{B}, \mathrm{S})).\\
&\mathrm{B} \rightarrow \mathrm{S}: V_{BS}.
\end{aligned}$$

Round 4: After receiving the messages sent by M and B, S checks whether $V_{AS} = E_{\mathrm{h}(pw_A)}(g^x, \mathrm{h}(g^{ax}, \mathrm{A}, \mathrm{B}, \mathrm{S}))$ and $V_{BS} = E_{\mathrm{h}(pw_B)}(g^y, \mathrm{h}(g^{by}, \mathrm{A}, \mathrm{B}, \mathrm{S}))$ hold or not. If it holds, S computes $V_{SA} = E_{g^{ax}}(g^x, g^y, \mathrm{A}, \mathrm{B})$ and $V_{SB} = E_{g^{by}}(g^x, g^y, \mathrm{B}, \mathrm{A})$ and sends V_{SA} and V_{SB} to A and B, respectively, But V_{SA} is intercepted by M. Otherwise S aborts the protocol.

$$\begin{aligned}
&\mathrm{S}: \text{Checks}\\
&\qquad V_{AS} = E_{\mathrm{h}(pw_A)}(g^x, \mathrm{h}(g^{ax}, \mathrm{A}, \mathrm{B}, \mathrm{S})).\\
&\qquad V_{BS} = E_{\mathrm{h}(pw_B)}(g^y, \mathrm{h}(g^{ay}, \mathrm{A}, \mathrm{B}, \mathrm{S})).\\
&\mathrm{S}: V_{SA} = E_{g^{ax}}(g^x, g^y, \mathrm{A}, \mathrm{B}).\\
&\mathrm{S}: V_{SB} = E_{g^{by}}(g^x, g^y, \mathrm{B}, \mathrm{A}).\\
&\mathrm{S} \rightarrow \mathrm{M}: V_{SA}.\\
&\mathrm{S} \rightarrow \mathrm{B}: V_{SB}.
\end{aligned}$$

Round 5: After intercepting the message in Round 4, M checks whether $g^x \in (g^x, g^y, \mathrm{A}, \mathrm{B})$ hold or not, If it holds, M computes $K_{AB} = (g^y)^x$. Otherwise M aborts the protocol. After receiving the message sent by S, B checks whether $g^y \in (g^x, g^y, \mathrm{B}, \mathrm{A})$ hold or not, If it holds, B computes $K_{BA} = (g^x)^y$. Otherwise B aborts the protocol. Then, M and B compute a common session key K = h(A, B, S, K_{AB}) = h(A, B, S, K_{BA}) = h(A, B, S, g^{xy}), respectively. Finally, B believes the common session key K = h(A, B, S, K_{AB}) is true. B also believes that he communicate with A. In fact, M gets the session key K = h(A, B, S, K_{AB}) and impersonates A to communicate with B.

$$\begin{aligned}
&\mathrm{M}: \text{Checks}\\
&\qquad g^x \in (g^x, g^y, \mathrm{A}, \mathrm{B}).\\
&\mathrm{M}: K_{AB} = (g^y)^x.\\
&\mathrm{M}: \mathrm{K} = \mathrm{h}(\mathrm{A}, \mathrm{B}, \mathrm{S}, K_{AB})\\
&\mathrm{B}: \text{Checks}\\
&\qquad g^y \in (g^x, g^y, \mathrm{B}, \mathrm{A}).\\
&\mathrm{B}: K_{BA} = (g^x)^y\\
&\mathrm{B}: \mathrm{K} = \mathrm{h}(\mathrm{A}, \mathrm{B}, \mathrm{S}, K_{BA})
\end{aligned}$$

4 Improved Scheme

In this section, we present an enhanced protocol to remedy the security loopholes existing in Xu et al.'s protocol. The protocol depicted in Fig. 2 works as follows:

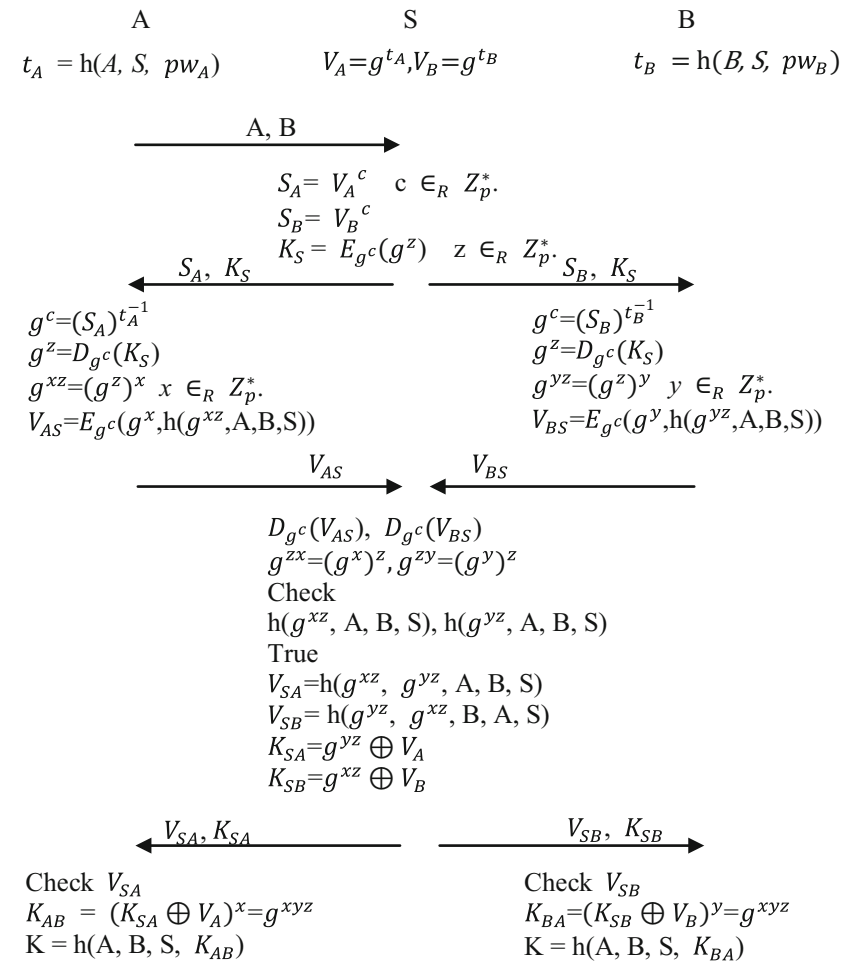

Fig. 2. The proposed protocol

Before the running of the protocol, Alice and Bob sends their verifiers V_A and V_B to S through a secure channel. S stores V_A and V_B in a password table.

Step 1: User A sends A and B to S.

$$A \rightarrow S : A, B.$$

Step 2: After receiving the messages sent by A, S randomly chooses z and c, computes $S_A = V_A^c$, $S_B = V_B^c$, $K_S = E_{g^c}(g^z)$ and then sends (S_A, K_S) and (S_B, K_S) to A and B, respectively.

$$\begin{aligned}
&\mathrm{S}: \mathrm{z},\ \mathrm{c} \in_R Z_p^*.\\
&\mathrm{S}: S_A = V_A^c.\\
&\mathrm{S}: S_B = V_B^c.\\
&\mathrm{S}: K_S = E_{g^c}(g^z).\\
&\mathrm{S} \rightarrow \mathrm{A}: S_A, K_S.\\
&\mathrm{S} \rightarrow \mathrm{B}: S_B, K_S.
\end{aligned}$$

Step 3: After receiving the message sent by S, A computes $g^c = (S_A)^{t_A^{-1}}$, $g^z = D_{g^c}(K_S)$, $g^{xz} = (g^z)^x$, and $V_{AS} = E_{g^c}(g^x, \mathrm{h}(g^{xz}, \mathrm{A}, \mathrm{B}, \mathrm{S}))$ by choosing $\mathrm{x} \in_R Z_p^*$, and sends V_{AS} to S. Similarly, after receiving the message from S, B computes $g^c = (S_B)^{t_B^{-1}}$, $g^z = D_{g^c}(K_S)$, $g^{yz} = (g^z)^y$ and $V_{BS} = E_{g^c}(g^y, \mathrm{h}(g^{yz}, \mathrm{A}, \mathrm{B}, \mathrm{S}))$ by choosing $\mathrm{y} \in_R Z_p^*$, and sends V_{BS} to S. Note that $t_A = \mathrm{h}(A, S, pw_A)$ and $t_B = \mathrm{h}(B, S, pw_B)$.

$$\begin{aligned}
&\mathrm{A}: g^c = (S_A)^{t_A^{-1}}.\\
&\mathrm{A}: g^z = D_{g^c}(K_S).\\
&\mathrm{A}: g^{xz} = (g^z)^x\ x \in_R Z_p^*.\\
&\mathrm{A}: V_{AS} = E_{g^c}(g^x, \mathrm{h}(g^{xz}, \mathrm{A}, \mathrm{B}, \mathrm{S}))\\
&\mathrm{A} \rightarrow \mathrm{S}: V_{AS}.\\
&\mathrm{B}: g^c = (S_B)^{t_B^{-1}}.\\
&\mathrm{B}: g^z = D_{g^c}(K_S).\\
&\mathrm{B}: g^{yz} = (g^z)^y\ y \in_R Z_p^*.\\
&\mathrm{B}: V_{BS} = E_{g^c}(g^y, \mathrm{h}(g^{yz}, \mathrm{A}, \mathrm{B}, \mathrm{S}))\\
&\mathrm{B} \rightarrow \mathrm{S}: V_{BS}.
\end{aligned}$$

Step 4: After receiving the messages sent by A and B, S computes $g^{xz} = (g^x)^z$, $g^{yz} = (g^y)^z$ by $D_{g^c}(V_{AS})$ and $D_{g^c}(V_{BS})$ and verifies whether $\mathrm{h}(g^{zx}, \mathrm{A}, \mathrm{B}, \mathrm{S}) = \mathrm{h}(g^{xz}, \mathrm{A}, \mathrm{B}, \mathrm{S})$, $\mathrm{h}(g^{zy}, \mathrm{A}, \mathrm{B}, \mathrm{S}) = \mathrm{h}(g^{yz}, \mathrm{A}, \mathrm{B}, \mathrm{S})$ or not. If they hold, S computes and sends $V_{SA} = \mathrm{h}(g^{xz}, g^{yz}, \mathrm{A}, \mathrm{B}, \mathrm{S})$, $K_{SA} = g^{yz} \oplus V_A$. and $V_{SB} = \mathrm{h}(g^{yz}, g^{xz}, \mathrm{B}, \mathrm{A}, \mathrm{S})$, $K_{SB} = g^{xz} \oplus V_B$. to A and B, respectively. Otherwise, S terminates the protocol.

$$\mathrm{S}: D_{g^c}(V_{AS}),\ D_{g^c}(V_{BS}).$$
$$\mathrm{S}: g^{xz} = (g^x)^z\ g^{yz} = (g^y)^z$$
$$\mathrm{S}: \text{Check}$$
$$\mathrm{h}(g^{zx}, \mathrm{A}, \mathrm{B}, \mathrm{S}) = \mathrm{h}(g^{xz}, \mathrm{A}, \mathrm{B}, \mathrm{S})$$
$$\mathrm{h}(g^{zy}, \mathrm{A}, \mathrm{B}, \mathrm{S}) = \mathrm{h}(g^{yz}, \mathrm{A}, \mathrm{B}, \mathrm{S})$$
$$\text{True.}$$
$$\mathrm{S}: V_{SA} = \mathrm{h}(g^{xz}, g^{yz}, \mathrm{A}, \mathrm{B}, \mathrm{S}),\ K_{SA} = g^{yz} \oplus V_A.$$
$$\mathrm{S}: V_{SB} = \mathrm{h}(g^{yz}, g^{xz}, \mathrm{B}, \mathrm{A}, \mathrm{S}),\ K_{SB} = g^{xz} \oplus V_B.$$
$$\mathrm{S} \rightarrow \mathrm{A}: V_{SA}, K_{SA}.$$
$$\mathrm{S} \rightarrow \mathrm{B}: V_{SB}, K_{SB}.$$

Finally: After receiving the message sent by S, A computes $g^{yz} = K_{SA} \oplus V_A$, than verifies whether h(g^{xz}, g^{yz}, A, B, S) = V_{SA} or not. If it holds A computes $K_{AB} = (K_{SA} \oplus V_A) = g^{xyz}$ and K = h(A, B, S, K_{AB}). Otherwise, A terminates the protocol. Similarly, After receiving the message sent by S, B computes $g^{xz} = K_{SB} \oplus V_B$, than verifies whether h(g^{yz}, g^{xz}, B, A, S) = V_{SB} or not. If it holds B computes $K_{BA} = (K_{SB} \oplus V_B)^y = g^{xyz}$ and K = h(A, B, S, K_{BA}). Otherwise, B terminates the protocol. Finally, Alice and Bob negotiate a common session key K = h(A, B, S, K_{AB}) = h(A, B, S, K_{BA}).

5 Security Analysis and Performance Comparison

5.1 Security Analysis

In this section, we prove the security of 3PAKE using those definitions in [22].

Theorem 1. The proposed protocol provides the property of the perfect forward secrecy.

Proof. Perfect forward secrecy is provided in the situation that even though a password is compromised M cannot derive previous session keys. To analyze this, suppose that M knows the password pw Then M tries to find previous session keys from the information collected by passive attack in past communication sessions, i.e., $K_S = E_{g^c}(g^z)$, g^x, g^y, g^{yz}, g^{xz}. However, she cannot do these using them without solving DLP and DHP. Therefore, the proposed protocol provides the property of perfect forward secrecy.

Theorem 2. The proposed protocol is secure against the Denning-Sacco attack.

Proof. To be secure against the Denning-Sacco attack, the protocol should be designed such that even though a session key is compromised, M cannot compute the password and confirm the correctness of the guessed password. To analyze this, suppose that M knows a session key K = h(A, B, S, K_{AB}). Then M tries to compute the password or confirm the correctness of the guessed password from it and the information collected by passive attack in past communication sessions, i.e., g^x, g^y, g^{yz}, g^{xz}, h(A, B, S, K_{AB}).

However, M cannot do these using them without solving DLP and DHP. Therefore, PAKE is secure against the Denning Sacco attack.

Theorem 3. The proposed protocol is secure against stolen-verifier attack.

Proof. The protocol being secure against stolen-verifier attack means an attacker not being able to pose as a client after compromising the server. In the proposed protocol, if M gains password file, M may know two client's verifiers $V_A = g^{h(A,S,pw_A)}$ and $V_B = g^{h(A,S,pw_B)}$. However, M cannot pose as the clients because of not knowing $t_A = h(A, S, pw_A)$ and $t_B = h(A, S, pw_B)$ used in step 3. Therefore, the proposed protocol is secure against server compromise.

Theorem 4. The proposed protocol is secure against man-in-the-middle attack.

Proof. We analyze if a malicious insider M can succeed in launching man-in-the-middle attack. Suppose that M tries to masquerade A or B. However, S can detect this attack when verifying $V_{AS} = E_{g^c}(g^x, h(g^{xz}, A, B, S))$ and $V_{BS} = E_{g^c}(g^y, h(g^{yz}, A, B, S))$. M cannot compute the valid g^{xz} or g^{yz} due to not knowing their correct passwords. Therefore, the improved scheme can resist man-in the-middle attack.

5.2 Efficiency Analysis

Performance of key agreement protocols can be approximated in terms of communication and computation loads. We compare our improved 3PAKE with the protocol of Xu et al. Table 2 shows the comparison regarding with several efficiency factors such as the number of rounds, random numbers, exponentiations, symmetric encryption/decryption, hash functions.

Table 2. The performance comparison

	Xu et al.			Our scheme		
	A	B	S	A	B	S
Random number	1	1	2	1	1	1
Exponentiation	3	3	4	4	4	6
Sym. enc./dec.	3	3	6	2	2	3
Hash function	2	2	2	3	3	4
Round	4			4		

As shown in Table 2, for user A and B, our scheme has one more exponentiation operation and one more hash operation than Xu et al.'s scheme, but our scheme has one less symmetric encryption/decryption computations than Xu et al.'s scheme. For server S our scheme has two more exponentiation operations and two more hash operation than Xu et al.'s scheme, but our scheme has three less symmetric encryption/decryption computations than Xu et al.'s scheme. Usually the cost of symmetric encryption/decryption is much larger than the cost of exponentiation operation (160bit)

and hash operation. Thus, our protocol has better performance than Xu et al.'s protocol. Moreover, Xu et al.'s protocol is vulnerable to the stolen-verifier attack and our protocol could overcome such weakness. Therefore, our protocol is more suitable for the practical applications.

6 Conclusion

In this paper, we show that Xu et al.'s 3PAKE protocol is vulnerable to the stolen-verifier attack and propose a new 3PAKE protocol to solve this problem. Security and performance analysis show our protocol overcome the weakness in Xu et al.'s protocol and has better performance. One of our future works is to extend our new scheme to multi-server architecture for the distributed systems.

Acknowledgments. The work was supported by the Educational Commission of Hubei Province of China (No. D20151401) and the Green Industry Technology Leading Project of Hubei University of Technology (No. ZZTS2017006).

References

1. Bellovin, S.M., Merritt, M.: Encrypted key exchange: password based protocols secure against dictionary attacks. In: Proceedings of IEEE Symposium on Research in Security and Privacy, pp. 72–84 (1992)
2. Ruan, O., Kumar, N., He, D.B., Lee, J.H.: Efficient provably secure password-based explicit authenticated key agreement. Pervasive Mob. Comput. **24**(12), 50–60 (2015)
3. Yi, X., Rao, F.Y., Tari, Z., Hao, F.: ID2S password-authenticated key exchange protocols. IEEE Trans. Comput. **65**, 1–14 (2016)
4. Lu, Y., Zhang, Q., Li, J., Shen, J.: Comment on a certificateless one-pass and two-party authenticated key agreement protocol. Inf. Sci. **369**, 184–187 (2016)
5. Zhang, L.: Certificateless one-pass and two-party authenticated key agreement protocol and its extensions. Inf. Sci. **293**(1), 182–195 (2015)
6. Farash, M.S., Islam, S.H., Obaidat, M.S.: A provably secure and efficient two-party password-based explicit authenticated key exchange protocol resistance to password guessing attacks. Concurrency Comput. Prac. Experience **27**(17), 4897–4913 (2015)
7. Xie, Q., Dong, N., Tan, X., et al.: Improvement of a three-party password-based key exchange protocol with formal verification. Inf. Technol. Control **42**(3), 231–237 (2013)
8. Chang, C.-C., Cheng, Y.-F.: A novel three-party encrypted key exchange protocol. Comput. Stan. Interfaces **26**(5), 471–476 (2004)
9. Lee, T.-F., Hwang, T., Lin, C.-L.: Enhanced three-party encrypted key exchange without server public keys. Comput. Secur. **23**, 571–577 (2004)
10. Lin, C.-L., Sun, H.-M., Hwang, T.: Three-party encrypted key exchange: attacks and a solution. ACM Operating Syst. Rev. **34**(4), 12–20 (2000)
11. Sun, H.-M., Chen, B.-C., Hwang, T.: Secure key agreement protocols for three-party against guessing attacks. J. Syst. Softw. **75**(1–2), 63–68 (2005)
12. Islam, S.H.: Design and analysis of a three party password-based authenticated key exchange protocol using extended chaotic maps. Inf. Sci. **312**(C), 104–130 (2015)

13. Amin, R., Biswas, G.P.: Cryptanalysis and design of a three-party authenticated key exchange protocol using smart card. Arab. J. Forence Eng. **40**(11), 1–15 (2015)
14. Lu, C.F.: Multi-party password-authenticated key exchange scheme with privacy preservation for mobile environment. Ksii Trans. Internet Inf. Syst. **9**(12), 5135–5149 (2015)
15. Nam, J., Paik, J., Kim, J., Lee, Y., Won, D.: Server-aided password-authenticated key exchange: from 3-party to group. In: International Conference on Human Interface & The Management of Information, vol. 6771, pp. 339–348 (2011)
16. Ding, Y., Horster, P.: Undetectable on-line password guessing attack. ACM SIGOPS Operating Syst. Rev. **29**(4), 77–86 (1995)
17. Lee, S.W., Kim, H.S., Yoo, K.Y.: Efficient verifier-based key agreement protocol for three parties without server's public key. Appl. Math. Comput. **167**(2), 996–1003 (2005)
18. Wang, R.C., Mo, K.R.: Security enhancement on efficient verifier-based key agreement protocol for three parties without server's public key. Int. Math. Forum **1**(17–20), 965–972 (2006)
19. Kwon, J.O., Jeong, I.R., Sakurai, K., et al.: Efficient verifier-based password-authenticated key exchange in the three-party setting. Comput. Stand. Interfaces **29**(5), 513–520 (2007)
20. Li, W., Wen, Q., Zhang, H.: Verifier-based password-authenticated key exchange protocol for three-party. J. Commun. **29**(10), 149–152 (2008)
21. Xu, et al.: Efficient three-party password-based authenticated key exchange protocol. J. Univ. Electron. Sci. Technol. China **41**(4), 596–598 (2012)
22. Lee, S.W., Kim, W.H., Kim, H.S., et al.: Efficient password-based authenticated key agreement protocol. Lecture Notes in Computer Science, pp. 617–626 (2004)
23. Dolev, D., Yao, A.C.: On the security of public key protocols. IEEE Trans. Inf. Theory **29**, 198–208 (1983)