Cryptanalysis of Guo et al.’s three-party password-based authenticated key exchange (G-3PAKE) protocol

Sung-Bae Choi\textsuperscript{a}, Eun-Jun Yoon\textsuperscript{b,∗}

\textsuperscript{a}Korea Institute of Science and Technology Information, Republic of Korea
\textsuperscript{b}School of Computer Engineering, Kyungil University, Republic of Korea

Abstract

In 2008, Guo et al. have shown that Lu and Cao’s simple three-party protocol for password-authenticated key exchanges (S-3PAKE) is indeed completely insecure against a kind of man-in-the-middle attack and the undetectable on-line password guessing attack. In addition, they have provided an improved protocol (G-3PAKE) that addresses the identified security problems. However, this paper demonstrates G-3PAKE protocol still falls to undetectable on-line password guessing attack by any other client.

©2011 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of ICAE2011.

Keywords: Password-authenticated key exchange, Cryptanalysis, Three-party, Security

1. Introduction

To apply two-party password-based authenticated key exchange (2PAKE) protocols to a large scale peer-to-peer system, 2PAKE protocols have a common problem, that is, each pair of communication parties in a group needs to pre-share a secret. It means that each user has to keep a large number of secrets for communicating with a group of users. To solve this problem, various three-party password-based authenticated key exchange (3PAKE) protocols were proposed ([1, 2, 3, 4, 5]). In the 3PAKE protocols, a trusted server assists each pair of users to authenticate each other and share a session key. In addition, the user does not need to keep a large number of secrets for a group of users. With the server’s help, each user only shares one secret with the server in 3PAKE protocols.

Because users usually choose easy-to-remember passwords, PAKE protocols can be vulnerable to password guessing attacks. Unlike typical private keys, the password has limited entropy,

\textsuperscript{∗}Corresponding author. Tel.: +82-53-850-7291; Fax: +82-53-850-7609.

Email address: ejyoon@kiu.ac.kr (Eun-Jun Yoon)
and is constrained by the memory of the user. For example, one alphanumerical character has 6 bits of entropy. Therefore, the goal of the attacker, which is to obtain a legitimate communication party’s password, can be achieved within a reasonable time. Thus, the password guessing attacks on PAKE protocols should be considered realistic. In general, the password guessing attacks can be divided into three classes (Kim et al. [5], Ding et al. [6]):

- Detectable on-line password guessing attacks: an attacker attempts to use a guessed password in an on-line transaction. He/she verifies the correctness of his/her guess using the response from server. A failed guess can be detected and logged by the server.
- Undetectable on-line password guessing attacks: similar to above, an attacker tries to verify a password guess in an online transaction. However, a failed guess cannot be detected and logged by server, as the server is not able to distinguish an honest request.
- Off-line password guessing attacks: an attacker guesses a password and verifies his/her guess off-line. No participation of server is required, so the server does not notice the attack from a malicious one.

In 2007, Lu and Cao ([1]) proposed a simple three-party password-based authenticated key exchange (S-3PAKE) protocol, where two clients, each shares a human-memorable password with a trusted server, can construct a secure session key. They argued that their S-3PAKE protocol can resist against various known attacks. In 2008, Guo et al. ([4]), however, have shown that Lu and Cao’s S-3PAKE protocol is indeed completely insecure against a kind of man-in-the-middle attack and the undetectable on-line password guessing attack. In addition, they have provided an improved protocol (G-3PAKE) that addresses the identified security problems. Nevertheless, this paper demonstrates G-3PAKE protocol still falls to undetectable on-line password guessing attack ([3, 4, 7]) by any other client.

2. A review of G-3PAKE protocol

This section briefly reviews G-3PAKE protocol ([4]).

2.1. Notations

- \( S, A, B \): a trusted server and two clients, respectively.
- \( pw_A, pw_B \): the password shared between \( A \) and \( S \) and between \( B \) and \( S \), respectively.
- \( k_{AS}, k_{BS} \): the MAC key shared between \( A \) and \( S \) and between \( B \) and \( S \), respectively.
- \( G, g, p \): a finite cyclic group \( G \) generated by an element \( g \) of prime order \( p \).
- \( M, N \): two elements in \( G \).
- \( x \in Z_p^* \): randomly choosing an element \( x \) of \( Z_p^* \).
- \( MAC(\cdot) \): a message authentication code.
- \( H_1(\cdot), H_2(\cdot) \): two secure one-way hash functions.
- \( || \): a bitwise concatenation.

2.2. Protocol description

Assume that two clients \( A \) and \( B \) wish to agree on a common session key. As they do not hold any shared information in advance, they cannot directly authenticate each other and have to resort to the trusted server \( S \). The detailed steps of the G-3PAKE protocol, as shown in Fig. 1, are described as follows:
Step 1. To establish a MAC key $k_{AS}$, a secure 2PAKE protocol is executed between $A$ and $S$.
Step 2. To establish a MAC key $k_{BS}$, a secure 2PAKE protocol is executed between $B$ and $S$.
Step 3. $A$ chooses a random number $x \in Z_p^*$, computes $X = g^x \cdot M_{pwA}$ and $\delta_A = MAC_{k_{AS}}(X)$, and then sends $A||X||\delta_A$ to $B$.
Step 4. $B$ also chooses a random number $y \in Z_p^*$, computes $Y = g^y \cdot N_{pwB}$ and $\delta_B = MAC_{k_{BS}}(Y)$, and then sends $A||X||\delta_A||B||Y||\delta_B$ to $S$.
Step 5. Upon receiving $A||X||\delta_A||B||Y||\delta_B$, the server $S$ first uses the shared MAC keys $k_{AS}$ and $k_{BS}$ to verify the MAC $\delta_A$ of $X$ and the MAC $\delta_B$ of $Y$, respectively. If they do not hold, $S$ terminates the protocol. Otherwise, $S$ uses the passwords $pwA$ and $pwB$ to compute $g^x = X/M_{pwA}$ and $g^y = Y/N_{pwB}$, respectively. Then, $S$ chooses another random number $z \in Z_p^*$ and computes $g^{\gamma_z} = (g^x)^z$ and $g^{\delta_z} = (g^y)^z$. Finally, $S$ sends $X'||Y'$ to $B$, where $X' = g^{\gamma_z} \cdot H_1(A, B, S, g^{\gamma_z})$ and $Y' = g^{\delta_z} \cdot H_1(B, A, S, g^{\delta_z})$.
Step 6. Upon receiving $X'||Y'$, $B$ computes $g^{\gamma_z} = Y'/H_1(B, A, S, g^{\gamma_z})$ with the password $pw_B$ and computes $g^{\delta_z} = (g^{\gamma_z})^y$ with the random number $y$. $B$ then forwards $X'||\alpha$ to $A$, where $\alpha = H_1(A, B, g^{\delta_z})$.
Step 7. Upon receiving $X'||\alpha$, $A$ first computes $g^{\gamma_z} = X'/H_1(A, B, S, g^{\gamma_z})$ and $g^{\delta_z} = (g^{\gamma_z})^y$. Then, $A$ checks whether $\alpha = H_1(A, B, g^{\delta_z})$ holds or not. If it does not hold, $A$ terminates the protocol. Otherwise, $A$ is convinced that $g^{\gamma_z}$ is valid. In this case, $A$ can compute the session key $SK_A = H_2(A, B, g^{\gamma_z})$. Finally, $A$ sends $\beta$ to $B$ for validation, where $\beta = H_1(B, A, g^{\delta_z})$.
Step 8. Upon receiving $\beta$, $B$ checks whether $\beta = H_1(B, A, g^{\delta_z})$ holds or not. If it does hold, $B$
can compute the session key $SK_B = H_2(A, B, g^{xZ})$. Otherwise, $B$ terminates the protocol.

Finally, both $A$ and $B$ share a common session key $SK_A = SK_B = H_2(A, B, g^{xZ})$.

3. Cryptanalysis of G-3PAKE protocol

This section shows that G-3PAKE protocol is insecure to an undetectable on-line password guessing attack ([3, 4]), where a malicious client $B$ of G-3PAKE is able to legally gain information about the password by repeatedly and indiscernibly asking queries to the authentication server. The attack scenario is outlined in Fig. 2. A more detailed description of the attack is as follows:

---

**Figure 2: Undetectable on-line password guessing attack on on G-3PAKE protocol**

Step 1. $A$ operates as specified in the G-3PAKE protocol in the first step.

Step 2. Let $B$ be a malicious client mediating between $S$ and $A$. Upon receiving $A||X||\delta_A$ from the client $A$ in Step 3 of the G-3PAKE protocol in Fig. 1, $B$ first randomly generates $X'$ and $\hat{\alpha}$, and then sends $X'||\hat{\alpha}$ to $A$ as a message of Step 7 of the G-3PAKE protocol in Fig. 1. $A$ would verify $\hat{\alpha}$. Even $A$ may detect only once that $\hat{\alpha}$ is invalid, it really does not mean that $S$ detects a failure of $B$’s malicious trial.
Step 3. On the other hand, $B$ guesses a password $\text{pw}_A^*$ and establishes an authenticated and private channel with $S$. $B$ firstly computes $g^{x_A'} = X/M^{\text{pw}_A}$ for an unknown element $x_A' \in Z_p^*$. Then, $B$ selects a random element $y \in Z_p^*$, and computes $Y' = (g^{x_A'})^y \cdot N^{\text{pw}_B}$ and $\delta_y = MAC_{k_B}(Y')$. Finally, $B$ sends $A||X||\delta_y||B||Y'||\delta_y^S$ to $S$ in Step 4 of the G-3PAKE protocol in Fig. 1.

Step 4. Upon receiving $A||X||\delta_y||B||Y'||\delta_y^S$, $S$ firstly uses the shared MAC keys $k_{AS}$ and $k_{BS}$ to verify the MAC $\delta_y$ of $X$ and the MAC $\delta_y^S$ of $Y'$, respectively. Because they always hold, $S$ decrypts ciphertexts $X$ and $Y'$ using $M^{\text{pw}_A}$ and $N^{\text{pw}_B}$. Then, $S$ selects a random value $z \in Z_p^*$ and computes

$$g^x = (g^z)^y \text{ and } g^{x\cdot z} = (g^{x\cdot y})^z$$

$$X' = g^{x\cdot z}\cdot H_1(A, B, S, g^z)^{\text{pw}_A}, \text{ and } Y' = g^{x\cdot z}\cdot H_1(B, A, S, g^{x\cdot y})^{\text{pw}_B}$$

Finally, $S$ sends $X'||Y'$ to $B$ in Step 5.

Step 5. Upon receiving $X'||Y'$, $B$ computes

$$g^{x\cdot z} = Y'/H_1(B, A, S, g^{x\cdot y})^{\text{pw}_B} \text{ and } g^{x\cdot z} = X'/H_1(A, B, S, g^{x\cdot y})^{\text{pw}_A}$$

$B$ checks if $(g^{x\cdot z})_B = g^{x\cdot z}$ in Step 4. If the check passes, then $B$ confirms that the guessed password $\text{pw}_A^*$ is the correct one.

Step 6. Otherwise, $B$ repeatedly performs the above Steps 3-5 without being noticed by $S$.

**Experiment to verify the proposed attack:** The typical Pentium computer can search $\approx 17 \times 10^9$ password in an hour and the supercomputer ($\approx 70 \times 10^{12}$ per second) can search $\approx 252 \times 10^{15}$ password in an hour ([7]). In the case of the undetectable on-line password guessing attacks, additionally time is needed for the round trip delay time between the malicious user $B$ and the server $S$ as shown in Fig. 2, beside the guessing attack costs. Based on the proposed experiment (for the detail, please refer to our previous experiment ([7])), we can see that the proposed password guessing attacks are feasible.

4. Conclusions

Three-party authenticated key exchange technology has been widely deployed in various kinds of applications. In 2008, Guo et al. proposed an improved three-party password-based authenticated key exchange (G-3PAKE) protocol. However, we have demonstrated that G-3PAKE protocol still falls to undetectable on-line password guessing attack by any other client. For this reason, G-3PAKE protocol is insecure for practical application.

References

[1] Lu R X, Cao Z F, Simple three-party key exchange protocol, Computers and Security, Vol. 26, pp. 94-97, 2007.
[2] Yoon E J, Yoo K Y, Improving the novel three-party encrypted key exchange protocol, Computer Standards & Interfaces, Vol. 30, No. 5, pp. 309-314, 2008.
[3] Phan R C W, Yau W C, Goi B M, Cryptanalysis of simple three-party key exchange protocol (S-3PAKE), Information Sciences, Vol. 178, pp. 2849-2856, 2008.
[4] Guo H, Li Z, Mu Y, Zhang X, Cryptanalysis of simple three-party key exchange protocol. Computers & Security, Vol. 27, pp. 16-21, 2008.
[5] Kim H S, Choi J Y, Enhanced password-based simple three-party key exchange protocol, Computers & Electrical Engineering, Vol. 35, pp. 107-114, 2009.
[6] Ding Y, Horster P, Undetectable on-line password guessing attacks, ACM Operating Systems Review, Vol. 29, No. 4, pp. 77-86, 1995.
[7] Yoon E J, Yoo K Y, Cryptanalysis of a simple three-party password-based key exchange protocol, International Journal of Communication Systems, Vol. 24, No. 4, pp. 532-542, 2011.