Dictionary Attacks against Password-Based Authenticated Three-Party Key Exchange Protocols

Junghyun Nam[1], Kim-Kwang Raymond Choo[2], Moonseong Kim[3], Juryon Paik[4] and Dongho Won[4]
[1] Department of Computer Engineering, Konkuk University, Korea
[e-mail: jhnam@kku.ac.kr]
[2] Information Assurance Research Group, Advanced Computing Research Centre, University of South Australia, Australia
[e-mail: raymond.choo@unisa.edu.au]
[3] Information and Communications Examination Bureau, Korean Intellectual Property Office, Korea
[e-mail: moonseong@kipo.go.kr]
[4] Department of Computer Engineering, Sungkyunkwan University, Korea
[e-mail: wise96@ece.skku.ac.kr, dhwon@security.re.kr]
*Corresponding author: Dongho Won

Received October 7, 2013; accepted November 29, 2013; published December 27, 2013

Abstract

A three-party password-based authenticated key exchange (PAKE) protocol allows two clients registered with a trusted server to generate a common cryptographic key from their individual passwords shared only with the server. A key requirement for three-party PAKE protocols is to prevent an adversary from mounting a dictionary attack. This requirement must be met even when the adversary is a malicious (registered) client who can set up normal protocol sessions with other clients. This work revisits three existing three-party PAKE protocols, namely, Guo et al.'s (2008) protocol, Huang's (2009) protocol, and Lee and Hwang's (2010) protocol, and demonstrates that these protocols are not secure against offline and/or (undetectable) online dictionary attacks in the presence of a malicious client. The offline dictionary attack we present against Guo et al.'s protocol also applies to other similar protocols including Lee and Hwang's protocol. We conclude with some suggestions on how to design a three-party PAKE protocol that is resistant against dictionary attacks

Keywords: Password-based authenticated key exchange (PAKE), three-party key exchange, password security, offline dictionary attack, undetectable online dictionary attack

This work is supported by Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2010-0020210).

http://dx.doi.org/10.3837/tiis.2013.12.016

1. Introduction

Key exchange (also known as key establishment) is defined to be any process whereby a shared high-entropy key (also known as a session key) becomes available to two or more parties for subsequent cryptographic use. Password-based authenticated key exchange (PAKE) protocols are a class of key exchange protocols, and enable two or more parties communicating over a public network to generate a session key from their low-entropy passwords which are easy for humans to remember. It is generally regarded that the design of secure key exchange protocols (including PAKE protocols) is notoriously hard [1][2][3][4], and conducting security analysis for such protocols is time-consuming and error-prone. One of the key challenges in designing a PAKE protocol, for example, is to prevent dictionary attacks, in which an attacker exhaustively enumerates all possible passwords to discover the correct password. Dictionary attacks have been used by both criminals as well as law enforcement officers and digital forensics practitioners to gain access to password-protected data (e.g. on smartphones and portable devices based on RIM BlackBerry and Apple iOS platforms - see Elcomsoft Phone Password Breaker *http://www.elcomsoft.com/eppb.html*). The difficulty of designing PAKE protocols secure against dictionary attacks is increased in the three-party setting. Unlike the two-party setting where each pair of parties is assumed to hold a shared password, the three-party setting assumes that each party (commonly known as a client) shares no password with other clients but holds their individual password shared only with a trusted server. Therefore in three-party PAKE protocols, protocol designers would have to consider the security of passwords against attacks by malicious clients who can set up normal protocol sessions with other clients (see [5][6][7][8][9]).

Dictionary attacks can be classified into two types, online and offline. Unlike offline dictionary attacks where password guesses can be verified offline, online dictionary attacks are the ones where the attacker verifies each password guess via a new online transaction with the server. However, detectable online dictionary attacks are considered as insignificant since the server may lock out the problematic client after a certain number of invalid transactions. Informally, a three-party PAKE protocol is secure if detectable online dictionary attacks are the best possible attacks that an adversary can mount against the protocol. In other words, three-party PAKE protocols should be able to resist undetectable online dictionary attacks as well as offline dictionary attacks.

In this work, we revisit three existing three-party PAKE protocols, namely, Guo et al.'s (2008) protocol [10], Huang's (2009) protocol [11], and Lee and Hwang's (2010) protocol [12]. We demonstrate that all three protocols are insecure against dictionary attacks in the presence of a malicious client. More specifically, we mount an offline dictionary attack against Guo et al.'s protocol, a combined offline and online dictionary attack against Huang's protocol, and an undetectable online dictionary attack against Lee and Hwang's protocol. The offline dictionary attack mounted against Guo et al.'s protocol also applies to Lee and Hwang's protocol and the protocols of [13][14][6] (see Section 2.2). By identifying these vulnerabilities, we hope that similar security failures can be prevented in the future design of three-party PAKE protocols. We present simple countermeasures for Guo et al.'s protocol and Lee and Hwang's protocol, but the existence of a security proof for the modified protocols remains an open question. We also suggest ways in which designers of three-party PAKE protocols can reduce the possibility of dictionary attacks.

2. Revisiting Guo et al.'s Protocol

This section revisits the three-party PAKE protocol proposed by Guo, Lia, Mu and Zhang in 2008 [10], and demonstrates that this protocol is susceptible to an offline dictionary attack in the presence of a malicious client.

2.1 Protocol Description

In the three-party setting, a trusted server S provides its registered clients with a central authentication service. Let A and B be two registered clients who wish to establish a session key, and pw_A and pw_B denote the passwords of A and B respectively shared with S via a secure channel. The protocol's public parameters include:

- A finite cyclic group G of prime order q, and a generator g of G.
- Two elements M and N of group G.
- A two-party PAKE protocol 2PAKE.
- Two hash functions $F:\{0,1\}^* \to G$ and $H:\{0,1\}^* \to \{0,1\}^\ell$, where ℓ represents the bit length of session keys.
- A message authentication code (MAC) scheme consisting of two algorithms, namely a MAC generation algorithm Mac and a MAC verification algorithm Ver. Here, Ver outputs a bit, with 1 meaning `accept` and 0 meaning `reject`.

The protocol depicted in Fig. 1 works as follows:

1. A (and B) and S establish a shared secret key k_{AS} (k_{BS} respectively) by running the two-party protocol 2PAKE.
2. A chooses a random $x \in Z_q$, computes $X = g^x$, $X^* = X \cdot M^{pw_A}$ and $\delta_A = \mathsf{Mac}_{k_{AS}}(X^*)$, and sends $\langle A, X^*, \delta_A \rangle$ to B.
3. B selects a random $y \in Z_q$, computes $Y = g^y$, $Y^* = Y \cdot N^{pw_B}$ and $\delta_B = \mathsf{Mac}_{k_{BS}}(Y^*)$, and sends $\langle A, X^*, \delta_A, B, Y^*, \delta_B \rangle$ to S.
4. Using Ver, S verifies that δ_A and δ_B are both valid. If either verification fails, S aborts the protocol. Otherwise, S recovers $X = X^* / M^{pw_A}$ and $Y = Y^* / N^{pw_B}$, selects a random $z \in Z_q$, and computes

$$\begin{aligned}
\bar{X} &= X^z, \\
\bar{Y} &= Y^z, \\
pw_A^* &= F(A \| B \| S \| X)^{pw_A}, \\
pw_B^* &= F(B \| A \| S \| Y)^{pw_B}, \\
\bar{X}^* &= \bar{X} \cdot pw_B^*, \\
\bar{Y}^* &= \bar{Y} \cdot pw_A^*.
\end{aligned}$$

   Then S sends $\langle \bar{X}^*, \bar{Y}^* \rangle$ to B.

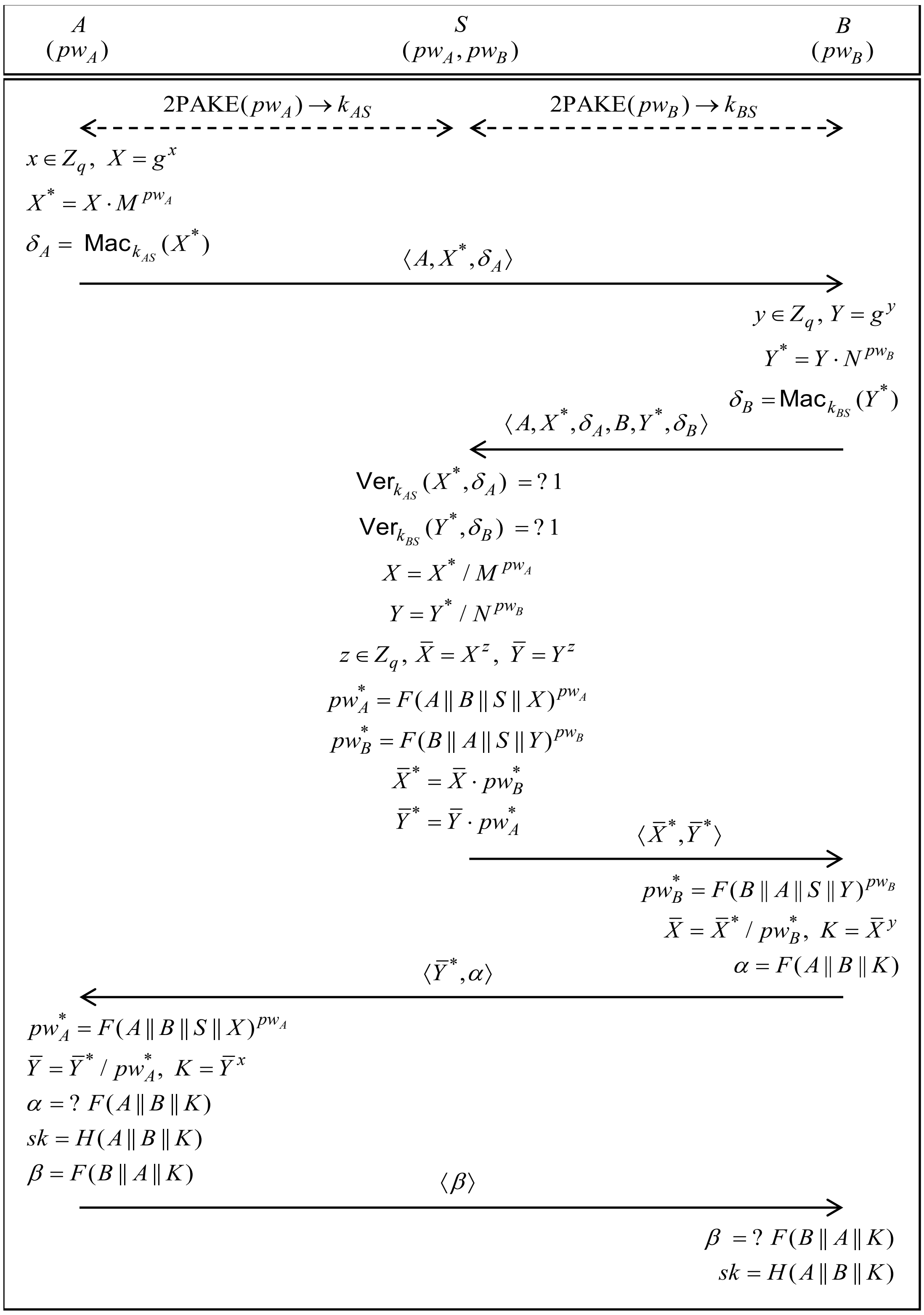
A (pw_A)
S (pw_A, pw_B)
B (pw_B)
$2\text{PAKE}(pw_A) \rightarrow k_{AS}$
$2\text{PAKE}(pw_B) \rightarrow k_{BS}$
$x \in Z_q,\ X = g^x$
$X^* = X \cdot M^{pw_A}$
$\delta_A = \mathsf{Mac}_{k_{AS}}(X^*)$
$\langle A, X^*, \delta_A \rangle$
$y \in Z_q, Y = g^y$
$Y^* = Y \cdot N^{pw_B}$
$\delta_B = \mathsf{Mac}_{k_{BS}}(Y^*)$
$\langle A, X^*, \delta_A, B, Y^*, \delta_B \rangle$
$\mathsf{Ver}_{k_{AS}}(X^*, \delta_A) = ?\ 1$
$\mathsf{Ver}_{k_{BS}}(Y^*, \delta_B) = ?\ 1$
$X = X^* / M^{pw_A}$
$Y = Y^* / N^{pw_B}$
$z \in Z_q, \bar{X} = X^z,\ \bar{Y} = Y^z$
$pw_A^* = F(A \| B \| S \| X)^{pw_A}$
$pw_B^* = F(B \| A \| S \| Y)^{pw_B}$
$\bar{X}^* = \bar{X} \cdot pw_B^*$
$\bar{Y}^* = \bar{Y} \cdot pw_A^*$
$\langle \bar{X}^*, \bar{Y}^* \rangle$
$pw_B^* = F(B \| A \| S \| Y)^{pw_B}$
$\bar{X} = \bar{X}^* / pw_B^*,\ K = \bar{X}^y$
$\alpha = F(A \| B \| K)$
$\langle \bar{Y}^*, \alpha \rangle$
$pw_A^* = F(A \| B \| S \| X)^{pw_A}$
$\bar{Y} = \bar{Y}^* / pw_A^*,\ K = \bar{Y}^x$
$\alpha = ?\ F(A \| B \| K)$
$sk = H(A \| B \| K)$
$\beta = F(B \| A \| K)$
$\langle \beta \rangle$
$\beta = ?\ F(B \| A \| K)$
$sk = H(A \| B \| K)$

Fig. 1. Guo et al.'s three-party PAKE protocol [10]

5. After receiving $\langle \overline{X}^*, \overline{Y}^* \rangle$, B computes

$$\begin{aligned} pw_B^* &= F(B \| A \| S \| Y)^{pw_B}, \\ \overline{X} &= \overline{X}^* / pw_B^*, \\ K &= \overline{X}^y, \\ \alpha &= F(A \| B \| K). \end{aligned}$$

   B then sends $\langle \overline{Y}^*, \alpha \rangle$ to A.

6. Upon receiving $\langle \overline{Y}^*, \alpha \rangle$, A computes

$$\begin{aligned} pw_A^* &= F(A \| B \| S \| X)^{pw_A}, \\ \overline{Y} &= \overline{Y}^* / pw_A^*, \\ K &= \overline{Y}^x. \end{aligned}$$

   A then checks if the equation $\alpha = F(A \| B \| K)$ holds. If it does not hold, A aborts the protocol. Otherwise, A computes the session key $sk = H(A \| B \| K)$ and $\beta = F(B \| A \| K)$, and sends $\langle \beta \rangle$ to B.

7. B checks if the equation $\beta = F(B \| A \| K)$ holds. If it holds, B computes the session key $sk = H(A \| B \| K)$. Otherwise, B aborts the protocol.

The correctness of the protocol is straightforward to verify, as shown below.

$$\begin{aligned} K &= \left(\frac{\overline{X}^*}{pw_B^*}\right)^y \\ &= \overline{X}^y \\ &= g^{xyz} \end{aligned}$$

and

$$\begin{aligned} K &= \left(\frac{\overline{Y}^*}{pw_A^*}\right)^x \\ &= \overline{Y}^x \\ &= g^{xyz}. \end{aligned}$$

2.2 A Previously Unpublished Offline Dictionary Attack, and a Simple Fix

Guo et al.'s protocol described above is vulnerable to the following dictionary attack where a malicious client A is able to verify all guesses on the password of client B in an offline manner.

Step 1. The attacker A initiates the protocol with the targeted client B, establishes a shared secret key k_{AS} with S, and then sends the message $\langle A, X^*, \delta_A \rangle$ to B.

Step 2. A eavesdrops on the message $\langle A, X^*, \delta_A, B, Y^*, \delta_B \rangle$ sent by B to S.

Step 3. When S sends the message $\langle \bar{X}^*, \bar{Y}^* \rangle$ to B, A replaces it with the forged message $\langle \hat{X}^*, \bar{Y}^* \rangle$ where

$$\hat{X}^* = \bar{X}^* \cdot X.$$

Since $\bar{X}^*$ was replaced with $\hat{X}^*$, B will compute α as $\alpha = F(A \| B \| \hat{K})$ where

$$\begin{aligned}\hat{K} &= \left(\frac{\hat{X}^*}{pw_B^*}\right)^y \\ &= \left(\frac{\bar{X}^* \cdot X}{pw_B^*}\right)^y \\ &= (\bar{X} \cdot X)^y \\ &= g^{xyz} \cdot g^{xy}.\end{aligned}$$

Step 4. Once the message $\langle \bar{Y}^*, \alpha \rangle$ is received from B, A aborts the protocol indicating that the session-key computation has failed due to an unexpected error, and then computes

$$\begin{aligned}K &= \left(\frac{\bar{Y}^*}{pw_A^*}\right)^x \\ &= \bar{Y}^x \\ &= g^{xyz}.\end{aligned}$$

Step 5. A makes a guess pw_B' on the password pw_B and computes

$$\begin{aligned}Y' &= Y^* / N^{pw_B'}, \\ \hat{K}' &= K \cdot Y'^x, \\ \alpha' &= F(A \| B \| \hat{K}').\end{aligned}$$

Step 6. A verifies the correctness of pw_B' by checking that α is equal to α'. If they are equal, then pw_B' is the correct password with an overwhelming probability.

Step 7. A repeats Steps 5 & 6 until the correct password is found.

This offline dictionary attack can have devastating implications for all clients registered with the server since the attack is likely to go undetected and the victim could be any of the clients. A possible countermeasure against the attack is to modify the server's message from $\langle \bar{X}^*, \bar{Y}^* \rangle$ to $\langle \bar{X}^*, \sigma_B, \bar{Y}^*, \sigma_A \rangle$, where $\sigma_B = \mathsf{Mac}_{k_{BS}}(S \| B \| A \| \bar{X}^*)$ and $\sigma_A = \mathsf{Mac}_{k_{AS}}(S \| A \| B \| \bar{Y}^*)$.

Guo et al.'s protocol was proposed as a fix to the flaws they found on the protocol of Lu and Cao (2007) [13]. We note that the offline dictionary attack above also applies to Lu and Cao's protocol [13] - see Appendix A - as well as its successors [14] [6].

3. Revisiting Huang's Protocol

In 2009, Huang [11] proposed a three-party PAKE protocol, claiming that the proposed protocol provides both security and efficiency without recourse to the use of server's public keys. However in 2011, Yoon and Yoo [8] pointed out that Huang's protocol is vulnerable not only to undetectable online dictionary attacks but also to offline dictionary attacks. In the same year, Lin and Hwang [9] also presented an undetectable online dictionary attack against Huang's protocol. In this section, we present a different (previously unpublished) dictionary attack against Huang's protocol, which is a combination of offline dictionary attacks and (undetectable) online dictionary attacks.

3.1 Protocol Description

Let A and B be two clients who wish to establish a session key, and pw_A and pw_B denote the passwords of A and B respectively shared with a trusted server S. Let p be a large prime number such that $p-1$ has a large prime factor q. Let G be a cyclic multiplicative subgroup of Z_p^* that has a prime order q, and g be a random generator of G (and the original protocol specification requires $q \geq 2^{256}$).

The protocol depicted in Fig. 2 works as follows:

1. A chooses a random number $x \in Z_q$ and computes

$$\begin{aligned} X &= g^x \bmod p, \\ V_A &= h(pw_A \| A \| B), \\ R_A &= X \oplus V_A, \end{aligned}$$

   where h is a cryptographic hash function and the symbol $\oplus$ denotes the bitwise XOR operation. A sends $\langle A, R_A \rangle$ to B.
2. B selects a random number $y \in Z_q$ and computes

$$\begin{aligned} Y &= g^y \bmod p, \\ V_B &= h(pw_B \| A \| B), \\ R_B &= Y \oplus V_B. \end{aligned}$$

   B then sends $\langle A, R_A, B, R_B \rangle$ to S.
3. After receiving $\langle A, R_A, B, R_B \rangle$ from B, S recovers X and Y by computing

$$\begin{aligned} V_A &= h(pw_A \| A \| B), \\ V_B &= h(pw_B \| A \| B), \\ X &= R_A \oplus V_A, \\ Y &= R_B \oplus V_B. \end{aligned}$$

   Next, S selects a random number $z \in Z_q$ and computes

$$\bar{X} = X^z \bmod p,$$
$$\bar{Y} = Y^z \bmod p,$$

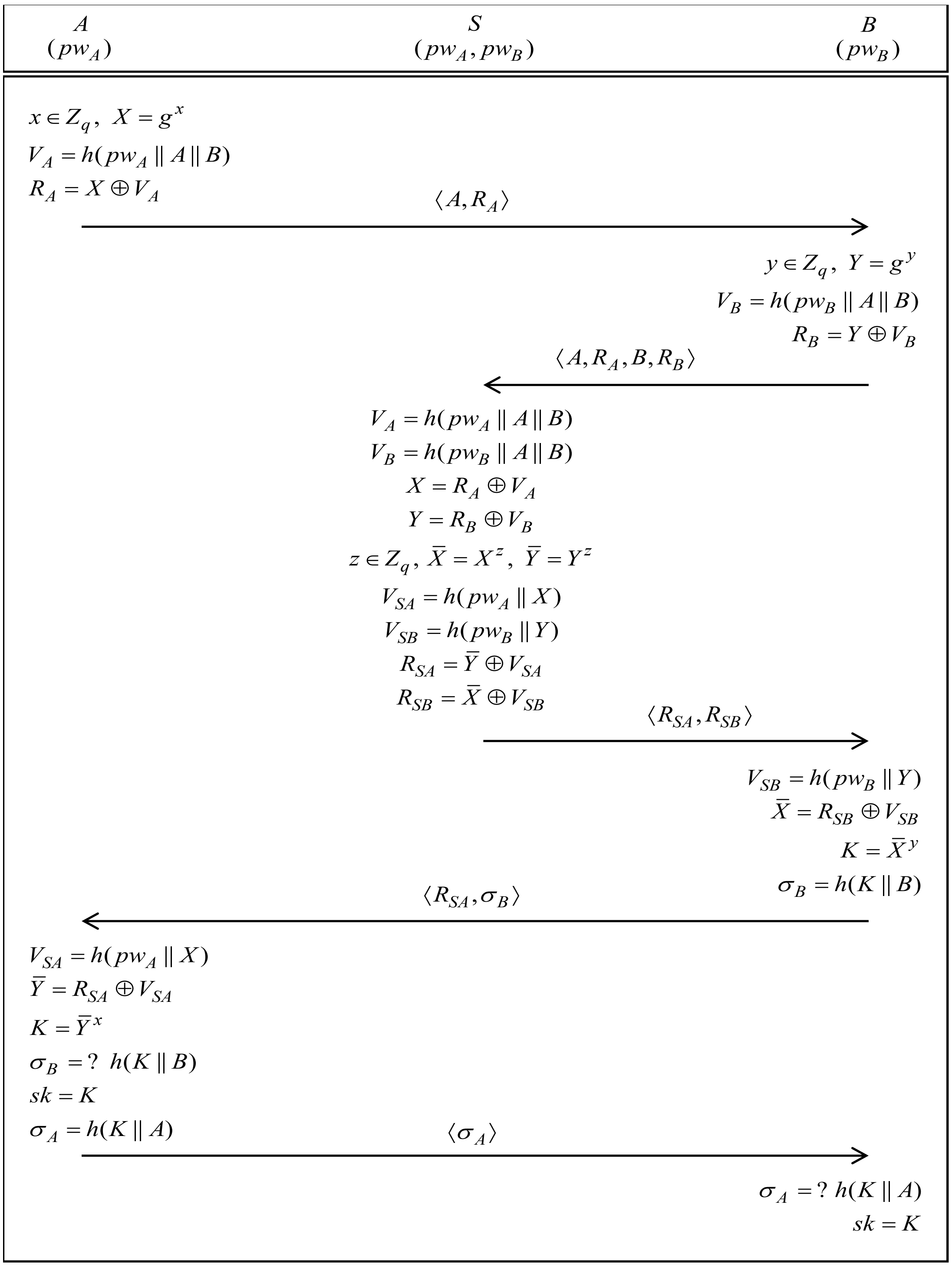
A (pw_A)
S (pw_A, pw_B)
B (pw_B)
$x \in Z_q,\ X = g^x$
$V_A = h(pw_A \| A \| B)$
$R_A = X \oplus V_A$
$\langle A, R_A \rangle$
$y \in Z_q,\ Y = g^y$
$V_B = h(pw_B \| A \| B)$
$R_B = Y \oplus V_B$
$\langle A, R_A, B, R_B \rangle$
$V_A = h(pw_A \| A \| B)$
$V_B = h(pw_B \| A \| B)$
$X = R_A \oplus V_A$
$Y = R_B \oplus V_B$
$z \in Z_q,\ \bar{X} = X^z,\ \bar{Y} = Y^z$
$V_{SA} = h(pw_A \| X)$
$V_{SB} = h(pw_B \| Y)$
$R_{SA} = \bar{Y} \oplus V_{SA}$
$R_{SB} = \bar{X} \oplus V_{SB}$
$\langle R_{SA}, R_{SB} \rangle$
$V_{SB} = h(pw_B \| Y)$
$\bar{X} = R_{SB} \oplus V_{SB}$
$K = \bar{X}^y$
$\sigma_B = h(K \| B)$
$\langle R_{SA}, \sigma_B \rangle$
$V_{SA} = h(pw_A \| X)$
$\bar{Y} = R_{SA} \oplus V_{SA}$
$K = \bar{Y}^x$
$\sigma_B = ?\ h(K \| B)$
$sk = K$
$\sigma_A = h(K \| A)$
$\langle \sigma_A \rangle$
$\sigma_A = ?\ h(K \| A)$
$sk = K$

Fig. 2. Huang's three-party PAKE protocol [11]

$$\begin{aligned} V_{SA} &= h(pw_A \| X), \\ V_{SB} &= h(pw_B \| Y), \\ R_{SA} &= \bar{Y} \oplus V_{SA}, \\ R_{SB} &= \bar{X} \oplus V_{SB}. \end{aligned}$$

S then sends $\langle R_{SA}, R_{SB} \rangle$ to B.

4. Upon receiving $\langle R_{SA}, R_{SB} \rangle$ from S, B computes

$$\begin{aligned} V_{SB} &= h(pw_B \| Y), \\ \bar{X} &= R_{SB} \oplus V_{SB}, \\ K &= \bar{X}^y \bmod p, \\ \sigma_B &= h(K \| B). \end{aligned}$$

Then B sends $\langle R_{SA}, \sigma_B \rangle$ to A.

5. After receiving $\langle R_{SA}, \sigma_B \rangle$ from B, A computes

$$\begin{aligned} V_{SA} &= h(pw_A \| X), \\ \bar{Y} &= R_{SA} \oplus V_{SA}, \\ K &= \bar{Y}^x \bmod p. \end{aligned}$$

Then, A checks whether the equation $\sigma_B = h(K \| B)$ holds or not. If it does not hold, A aborts the protocol. Otherwise, A sets the session key sk equal to K, computes $\sigma_A = h(K \| A)$, and sends $\langle \sigma_A \rangle$ to B.

6. B checks whether the equation $\sigma_A = h(K \| A)$ holds or not. If it does not hold, B aborts the protocol. Otherwise, B sets the session key to $sk = K$.

The correctness of the protocol can be easily verified as shown below.

$$\begin{aligned} sk &= (R_{SB} \oplus V_{SB})^y \\ &= \bar{X}^y \\ &= g^{xyz} \end{aligned}$$

and

$$\begin{aligned} sk &= (R_{SA} \oplus V_{SA})^x \\ &= \bar{Y}^x \\ &= g^{xyz}. \end{aligned}$$

3.2 A Previously Unpublished Combined Offline and Online Dictionary Attack

Our dictionary attack against Huang's protocol exploits two flaws in the design of the protocol: (1) the server does not authenticate any message from the clients and (2) the publicly transmitted keying materials (i.e., R_A, R_B, R_{SA} and R_{SB}) are computed using the bitwise

XOR operation when the multiplicative subgroup G is not closed under the XOR operation.

Let D be the set of all possible passwords. Assume that B is a malicious client who wants to discover the password of client A. The attack works as follows:

Step 1. The attacker B runs the protocol with client A and stores the first message $\langle A, R_A \rangle$ received from A.

Step 2. For each $pw'_A \in D$, B computes

$$V'_A = h(pw'_A \,||\, A \,||\, B),$$
$$X' = R_A \oplus V'_A,$$

and checks whether X' is an element of G or not. If $X' \notin G$, B deletes pw'_A from the dictionary D (i.e., $D = D ,\ \{pw'_A\}$). If $X' \notin G$, then $pw'_A \neq pw_A$. If we assume that p is a safe prime (i.e., $p = 2q + 1$), then this step would cut the size of D about in half.

Step 3. B generates R_B as specified in the protocol and sends $\langle A, R_A, B, R_B \rangle$ to S, indicating that A and B want to establish a session key. After receiving $\langle R_{SA}, R_{SB} \rangle$ from S, B proceeds to Step 4.

Step 4. For each $pw'_A \in D$, B computes

$$V'_A = h(pw'_A \,||\, A \,||\, B),$$
$$X' = R_A \oplus V'_A,$$
$$V'_{SA} = h(pw'_A \,||\, X'),$$
$$\bar{Y}' = R_{SA} \oplus V'_{SA},$$

and checks whether $\bar{Y}' \in G$ or not. If $\bar{Y}' \notin G$ (i.e., $pw'_A \neq pw_A$), B sets $D = D ,\ \{pw'_A\}$.

Step 5. B repeats Steps 3 & 4 until the correct password is found (i.e., until $|\mathrm{D}| = 1$).

The number of iterations of Steps 3 & 4 required to determine the correct password is bounded by $\log_2 |\mathrm{D}|$ in the case of $p = 2q + 1$. If p is much greater than q (e.g., $\log_2 p = 1024$ and $\log_2 q = 512$), performing Step 2 once will be sufficient to determine the correct password (with an overwhelming probability) and thus, no iterative pruning is needed.

It appears that there is no quick tweak we can apply to make Huang's protocol resistant to dictionary attacks such as the above. Note that simply replacing the bitwise XOR operation with the multiplicative operation would make the protocol vulnerable to such an attack as the one we presented against Guo et al.'s protocol in Section 2.2.

4. Revisiting Lee and Hwang's Protocol

We now revisit the last of the three protocols, namely Lee and Hwang's three-party PAKE protocol [12] – S-IA-3PAKE.

4.1 Protocol Description

Let S be the trusted server, and A and B be two registered clients of S who wish to establish a shared session key. We denote the passwords of A and B by pw_A and pw_B respectively. The S-IA-3PAKE protocol uses the following public parameters: (1) a large prime p and a generator g of Z_p, (2) two random elements M and N of Z_p, (3) a cryptographic hash function H used as a key derivation function, and (4) a pair of MAC generation/verification algorithms $(\mathsf{Mac}, \mathsf{Ver})$, where Ver outputs a bit, with 1 meaning `accept` and 0 meaning `reject`.

S-IA-3PAKE (see Fig. 3) works as follows:

Step 1. A chooses a random $x \in Z_p$, computes $X = g^x$ and $X^* = X \cdot M^{pw_A}$, and sends $\langle X^* \rangle$ to S. At the same time, S chooses a random $u \in Z_p$, computes $U = g^u$ and $U^* = U \cdot N^{pw_A}$, and sends $\langle U^* \rangle$ to A. A and S then recover U and X respectively, and establish a shared secret key $k_{AS} = g^{xu}$.

Step 2. B chooses a random $y \in Z_p$, computes $Y = g^y$ and $Y^* = Y \cdot M^{pw_B}$, and sends $\langle Y^* \rangle$ to S. At the same time, S chooses a random $v \in Z_p$, computes $V = g^v$ and $V^* = V \cdot N^{pw_B}$, and sends $\langle V^* \rangle$ to B. B and S recover V and Y respectively, and compute the shared secret key $k_{BS} = g^{yv}$.

Step 3. S chooses a random $w \in Z_p$ and computes

$$\bar{X} = X^w,\ \bar{Y} = Y^w,$$
$$\bar{X}^* = \bar{X} \cdot k_{BS},\ \bar{Y}^* = \bar{Y} \cdot k_{AS}.$$

S then sends $\langle \bar{Y}^* \rangle$ and $\langle \bar{X}^* \rangle$ to A and B respectively.

Step 4. A computes the key derivation secret, $K_A = (\bar{Y}^* / k_{AS})^x$, and the session key, $sk_A = H(A \| B \| K_A)$. Meanwhile, B computes $K_B = (\bar{X}^* / k_{BS})^y$ and $sk_B = H(A \| B \|$ $K_B)$.

Step 5. A and B perform key confirmation by exchanging $\sigma_{AB} = \mathsf{Mac}_{sk_A}(A \| B)$ and $\sigma_{BA} = \mathsf{Mac}_{sk_B}(B \| A)$ and verifying them in the straightforward way.

The correctness of S-IA-3PAKE can be easily verified from $K_A = K_B = g^{xyw}$.

4.2 A Previously Unpublished Undetectable Online Dictionary Attack, and a Simple Fix

We now demonstrate that S-IA-3PAKE is susceptible to a previously unpublished undetectable online dictionary attack. Suppose that A is a malicious client who wants to discover the password of client B. The attack works as follows:

Step 1. The attacker A notifies the server S that she wants to establish a session key with B.

Step 2. A chooses a random $x \in Z_p$, computes $X = g^x$ and $X^* = X \cdot M^{pw_A}$, and sends S the message $\langle X^* \rangle$ with its true identity.

Step 3. A makes a guess pw'_B on the password pw_B, computes Y^* as $Y^* = X \cdot M^{pw'_B}$, and sends S the message $\langle Y^* \rangle$ as if it is from B.

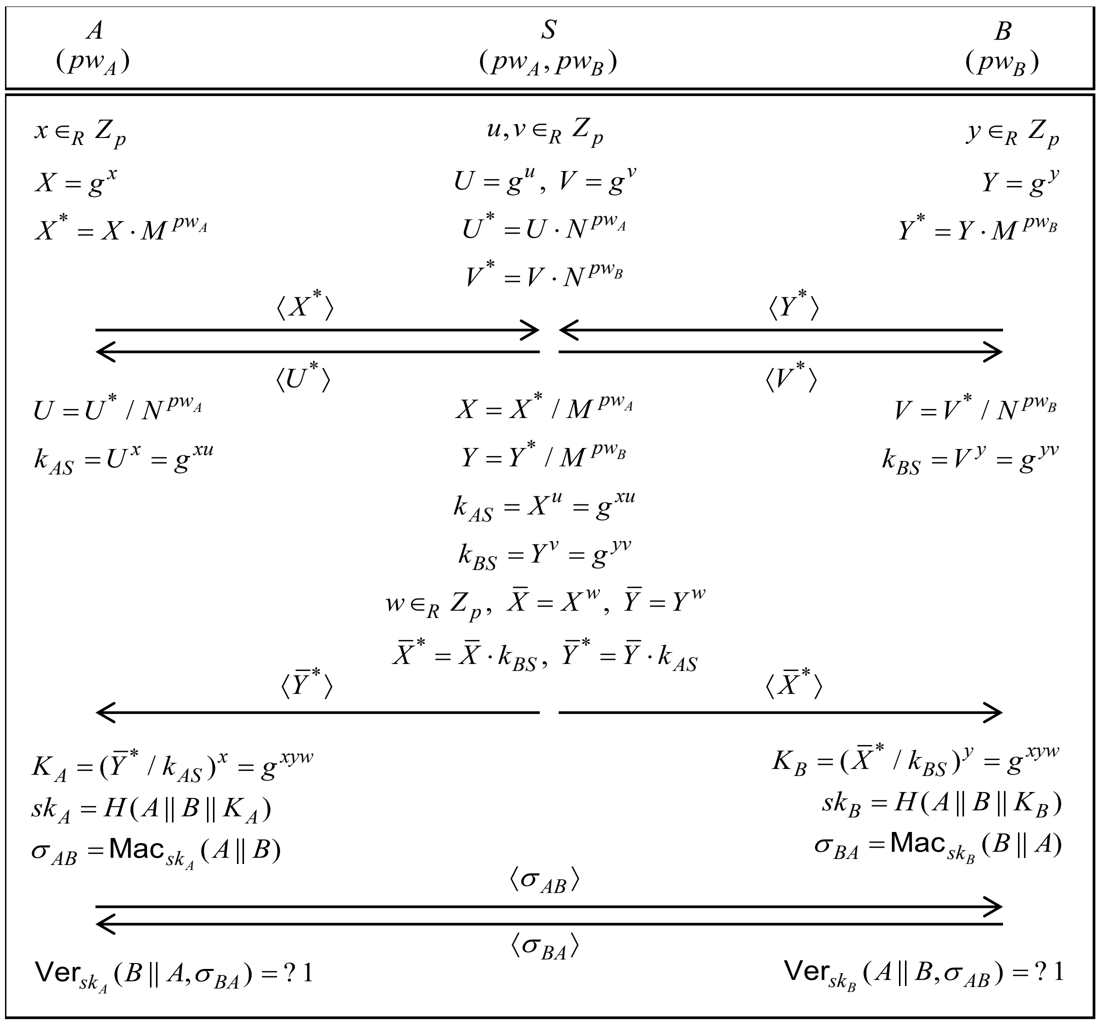
A (pw_A)
S (pw_A, pw_B)
B (pw_B)
$x \in_R Z_p$
$X = g^x$
$X^* = X \cdot M^{pw_A}$
$u, v \in_R Z_p$
$U = g^u,\ V = g^v$
$U^* = U \cdot N^{pw_A}$
$V^* = V \cdot N^{pw_B}$
$y \in_R Z_p$
$Y = g^y$
$Y^* = Y \cdot M^{pw_B}$
$\langle X^* \rangle$
$\langle Y^* \rangle$
$\langle U^* \rangle$
$\langle V^* \rangle$
$U = U^* / N^{pw_A}$
$k_{AS} = U^x = g^{xu}$
$X = X^* / M^{pw_A}$
$Y = Y^* / M^{pw_B}$
$k_{AS} = X^u = g^{xu}$
$k_{BS} = Y^v = g^{yv}$
$w \in_R Z_p,\ \bar{X} = X^w,\ \bar{Y} = Y^w$
$\bar{X}^* = \bar{X} \cdot k_{BS},\ \bar{Y}^* = \bar{Y} \cdot k_{AS}$
$V = V^* / N^{pw_B}$
$k_{BS} = V^y = g^{yv}$
$\langle \bar{Y}^* \rangle$
$\langle \bar{X}^* \rangle$
$K_A = (\bar{Y}^* / k_{AS})^x = g^{xyw}$
$sk_A = H(A \| B \| K_A)$
$\sigma_{AB} = \mathsf{Mac}_{sk_A}(A \| B)$
$K_B = (\bar{X}^* / k_{BS})^y = g^{xyw}$
$sk_B = H(A \| B \| K_B)$
$\sigma_{BA} = \mathsf{Mac}_{sk_B}(B \| A)$
$\langle \sigma_{AB} \rangle$
$\langle \sigma_{BA} \rangle$
$\mathsf{Ver}_{sk_A}(B \| A, \sigma_{BA}) = ?\ 1$
$\mathsf{Ver}_{sk_B}(A \| B, \sigma_{AB}) = ?\ 1$

Fig. 3. S-IA-3PAKE: Lee and Hwang's three-party PAKE protocol [12]

Step 4. After receiving $U^* = U \cdot N^{pw_A}$ from S, A computes the secret key $k_{AS} = g^{xu}$ as per the protocol specification.

Step 5. When S sends $V^* = V \cdot N^{pw_B}$ to B, A intercepts it and computes $V' = V^* / N^{pw'_B}$ and $k'_{BS} = V'^x$.

Step 6. If pw'_B is the correct password, then the value $\bar{Y}$ computed by S would be equal to $\bar{X}$ $(= g^{xw})$. After receiving $\bar{Y}^*$ and intercepting $\bar{X}^*$, A computes

$$K'_A = (\bar{Y}^* / k_{AS})^x \text{ and } K'_B = (\bar{X}^* / k'_{BS})^x,$$

and verifies the correctness of pw'_B by checking that K'_A is equal to K'_B. Note that if $pw'_B = pw_B$, then it must hold that $K'_A = K'_B$.

This online dictionary attack is undetectable and can be mounted repeatedly until the correct password is found. An obvious fix is to add client-to-server authentication, where both clients

A and B send the authenticators $\sigma_{AS} = \mathsf{Mac}_{k_{AS}}(A \| B \| S)$ and $\sigma_{BS} = \mathsf{Mac}_{k_{BS}}(B \| A \| S)$ to the server S respectively.

The S-IA-3PAKE protocol is also vulnerable to an offline dictionary attack similar to the one we presented against Guo et al.'s protocol in Section 2.2. Due to similarity, we omit the details of the attack scenario. To address this vulnerability, we recommend to modify the server's messages $\langle \overline{Y}^* \rangle$ and $\langle \overline{X}^* \rangle$ respectively to $\langle \overline{Y}^*, \sigma_{SA} \rangle$ and $\langle \overline{X}^*, \sigma_{SB} \rangle$, where $\sigma_{SA} = \mathsf{Mac}_{k_{AS}}(S \| A \| B \| \overline{Y}^*)$ and $\sigma_{SB} = \mathsf{Mac}_{k_{BS}}(S \| B \| A \| \overline{X}^*)$.

5. Concluding Remarks

We have examined several existing three-party PAKE protocols, including Guo et al.'s (2008) protocol [10], Huang's (2009) protocol [11], and Lee and Hwang's (2010) protocol [12], and demonstrated that they are vulnerable to previously unpublished offline and/or online dictionary attacks by a malicious client. This research confirms that achieving password security in the presence of a malicious client remains a challenging task in designing an efficient three-party PAKE protocol. Based on our findings, we propose that designers of three-party PAKE protocols should consider the following principles to mitigate dictionary attacks:

- Authenticate all the keying materials sent from the server to the clients, as this measure will increase the protocol's resilience against offline dictionary attacks.
- Do not use an operation under which the underlying group is not closed, when generating a password-entangled protocol message; the use of such an operation may result in the protocol being vulnerable to a combined offline and online dictionary attack similar to our attack against Huang's protocol (see Section 3).
- Ensure that all clients send at least one message to the server in an authenticated manner. Otherwise, the protocol is likely to be susceptible to an undetectable online dictionary attack.

Guo et al.'s protocol and Huang's protocol do not have accompanying proofs of security. Although Lee and Hwang's protocol carries a proof of security, the proof model used does not allow the adversary to corrupt protocol participants and thus cannot capture any kind of insider attacks, in particular, offline and online dictionary attacks by a malicious client. In other words, our dictionary attacks do not invalidate the existing proof of security for Lee and Hwang's protocol. As such, we recommend that protocol designers choose an appropriate proof model that adequately captures all the security requirements, so that protocol implementers can be assured of the security properties of protocols.

References

[1] C. Boyd and KKR. Choo, "Security of Two-Party Identity-Based Key Agreement," *Progress in Cryptology – Mycrypt 2005*, LNCS vol. 3715, pp. 229-243, 2005. Article (CrossRef Link)

[2] KKR. Choo, C. Boyd and Y. Hitchcock, "Errors in Computational Complexity Proofs for Protocols," *Advances in Cryptology – Asiacrypt 2005*, LNCS vol. 3788, pp. 624-643, 2005. Article (CrossRef Link)

[3] KKR. Choo, C. Boyd and Y. Hitchcock, "The Importance of Proofs of Security for Key Establishment Protocols: Formal Analysis of Jan-Chen, Yang-Shen-Shieh, Kim-Huh-Hwang-Lee, Lin-Sun-Hwang, and Yeh-Sun Protocols," *Computer Communications*, vol. 29, no. 15, pp.

2788-2797, 2006. Article (CrossRef Link)
[4] M. Gorantla, C. Boyd, J. Nieto and M. Manulis. "Modeling Key Compromise Impersonation Attacks on Group Key Exchange Protocols," *ACM Transactions on Information and System Security*, vol. 14, no. 4, Article 28, 2011. Article (CrossRef Link)
[5] H. Chen, T. Chen, W. Lee and C. Chang, "Security Enhancement for a Three-Party Encrypted Key Exchange Protocol against Undetectable On-Line Password Guessing Attacks," *Computer Standards & Interfaces*, vol. 30, no. 1-2, pp. 95-99, 2008. Article (CrossRef Link)
[6] J. Nam, J. Paik, H. Kang, U. Kim and D. Won, "An Off-Line Dictionary Attack on a Simple Three-Party Key Exchange Protocol," *IEEE Communications Letters*, vol. 13, no. 3, pp. 205-207, 2009. Article (CrossRef Link)
[7] N. Lo and K. Yeh, "Cryptanalysis of Two Three-Party Encrypted Key Exchange Protocols," *Computer Standards & Interfaces*, vol. 31, no. 6, pp. 1167-1174, 2009. Article (CrossRef Link)
[8] E. Yoon and K. Yoo, "Cryptanalysis of a Simple Three-Party Password-Based Key Exchange Protocol," *International Journal of Communication Systems*, vol. 24, no. 4, pp.532-542, 2011. Article (CrossRef Link)
[9] C. Lin and T. Hwang, "On 'a Simple Three-Party Password-Based Key Exchange Protocol'," *International Journal of Communication Systems*, vol. 24, no. 11, pp. 1520-1532, 2011. Article (CrossRef Link)
[10] H. Guo, Z. Li, Y. Mu and X. Zhang, "Cryptanalysis of Simple Three-Party Key Exchange Protocol," *Computers & Security*, vol. 27, no. 1, pp. 16-21, 2008. Article (CrossRef Link)
[11] H. Huang, "A Simple Three-Party Password-Based Key Exchange Protocol," *International Journal of Communication Systems*, vol. 22, no. 7, pp. 857-862, 2009. Article (CrossRef Link)
[12] T. Lee and T. Hwang, "Simple Password-Based Three-Party Authenticated Key Exchange without Server Public Keys," *Information Sciences*, vol. 180, no. 9, pp.1702-1714, 2010. Article (CrossRef Link)
[13] R. Lu and Z. Cao, "Simple Three-Party Key Exchange Protocol," *Computers & Security*, vol. 26, no. 1, pp. 94-97, 2007. Article (CrossRef Link)
[14] H. Chung and W. Ku, "Three Weaknesses in a Simple Three-Party Key Exchange Protocol," *Information Sciences*, vol. 178, no. 1, pp. 220-229, 2008. Article (CrossRef Link)

A. A Previously Unpublished Offline Dictionary Attack against Lu and Cao's Protocol

In this section, we revisit Lu and Cao's three-party PAKE protocol [13] and reveal a previously unpublished offline dictionary attack against the protocol. Unless stated otherwise, all notations used here are the same as those used in Section 2.

A.1. Protocol Description

Lu and Cao's protocol works as follows:

1. A chooses a random number $x \in Z_q$, computes $X = g^x$ and $X^* = X \cdot M^{pw_A}$, and sends $A \| X^*$ to B.
2. B selects a random number $y \in Z_q$, computes $Y = g^y$ and $Y^* = Y \cdot N^{pw_B}$, and sends $A \| X^* \| B \| Y^*$ to S.
3. Upon receiving $A \| X^* \| B \| Y^*$, S first recovers X and Y by computing $X = X^* / M^{pw_A}$ and $Y = Y^* / N^{pw_B}$. Next, S selects a random number $z \in Z_q$ and computes

$$\begin{aligned} \overline{X} &= X^z, \\ \overline{Y} &= Y^z, \\ pw_A^* &= F(A \| S \| X)^{pw_A}, \\ pw_B^* &= F(B \| S \| Y)^{pw_B}, \\ \overline{X}^* &= \overline{X} \cdot pw_B^*, \\ \overline{Y}^* &= \overline{Y} \cdot pw_A^*. \end{aligned}$$

S then sends $\overline{X}^* \| \overline{Y}^*$ to B.

4. After having received $\overline{X}^* \| \overline{Y}^*$, B computes

$$\begin{aligned} pw_B^* &= F(B \| S \| Y)^{pw_B}, \\ K &= \left(\frac{\overline{X}^*}{pw_B^*}\right)^y, \\ \alpha &= F(A \| B \| K), \end{aligned}$$

and sends $\overline{Y}^* \| \alpha$ to A.

5. With $\overline{Y}^* \| \alpha$ from B, A computes

$$\begin{aligned} pw_A^* &= F(A \| S \| X)^{pw_A}, \\ K &= \left(\frac{\overline{Y}^*}{pw_A^*}\right)^x, \end{aligned}$$

and verifies that α is equal to $F(A \| B \| K)$. If the verification fails, then A aborts the protocol. Otherwise, A computes the session key $sk = H(A \| B \| K)$ and sends $\beta = F(B \| A \| K)$ to B.

6. B verifies the correctness of β by checking that the equation $\beta = F(B \| A \| K)$ holds. If it holds, then B computes the session key $sk = H(A \| B \| K)$. Otherwise, B aborts the protocol.

A.2. The Attack Scenario

A malicious client, A, can mount an offline dictionary attack against Lu and Cao's protocol to find out the password of any other client, B, as follows:

Step 1. The attacker A initiates the protocol with the targeted client B and sends the message $A \| X^*$ to B as per the protocol specification.

Step 2. A eavesdrops on the message $A \| X^* \| B \| Y^*$ sent by B to S.

Step 3. A replaces the message $\overline{X}^* \| \overline{Y}^*$ with $\hat{X}^* \| \overline{Y}^*$, where $\hat{X}^* = \overline{X}^* \cdot X$. Due to the message replacement, B will compute α as $\alpha = F(A \| B \| \hat{K})$ where

$$\begin{aligned}\hat{K} &= \left(\frac{\hat{X}^*}{pw_B^*}\right)^y \\ &= (\bar{X} \cdot X)^y \\ &= g^{xyz} \cdot g^{xy}.\end{aligned}$$

Step 4. After receiving $\bar{Y}^* \| \alpha$, A aborts the protocol and then computes

$$\begin{aligned}K &= \left(\frac{\bar{Y}^*}{pw_A^*}\right)^x \\ &= \bar{Y}^x \\ &= g^{xyz}.\end{aligned}$$

Step 5. A makes a guess pw'_B on the password pw_B and computes

$$\begin{aligned}Y' &= Y^* / N^{pw'_B}, \\ \hat{K}' &= K \cdot Y'^x, \\ \alpha' &= F(A \| B \| \hat{K}').\end{aligned}$$

A then checks that α is equal to α'. If they are equal, pw'_B is the correct password with an overwhelming probability. Otherwise, A repeats Step 5 until the correct password is found.

Junghyun Nam received the B.E. degree in Information Engineering from Sungkyunkwan University, Korea, in 1997. He received his M.S. degree in Computer Science from University of Louisiana, Lafayette, in 2002, and the Ph.D. degree in Computer Engineering from Sungkyunkwan University, Korea, in 2006. He is now an associate professor in Konkuk University, Korea. His research interests include cryptography and computer security.

Kim-Kwang Raymond Choo received the PhD in Information Security from Queensland University of Technology in 2006. He has (co-)authored a number of publications including a book published in Springer's "Advances in Information Security" book series, a book published by Elsevier (Forewords written by Australia's Chief Defence Scientist and Chair of the Electronic Evidence Specialist Advisory Group, Senior Managers of Australian and New Zealand Forensic Laboratories), and six refereed monographs; and is the recipient of the 2010 ACT Pearcey Award, 2009 Fulbright Scholarship, 2008 Australia Day Achievement Medallion, British Computer Society's Wilkes Award for the best paper published in the 2007 volume of The Computer Journal, 2007 Queensland University of Technology Faculty of Information Technology Executive Dean's outstanding Ph.D. thesis commendation, and the 2005 Australasian Conference on Information Security and Privacy's Best Student Paper Award.

Moonseong Kim received the M.S. degree in Mathematics, August 2002 and the Ph.D. degree in Electrical and Computer Engineering, February 2007 both from Sungkyunkwan University, Korea. He was a research professor at Sungkyunkwan University in 2007. From December 2007 to October 2009, he was a visiting scholar in ECE and CSE, Michigan State University, USA. Since October 2009, he has been a patent examiner in Information and Communication Examination Bureau, Korean Intellectual Property Office (KIPO), Korea. His research interests include wired/wireless networking, sensor networking, mobile computing, network security protocols, and simulations/numerical analysis.

Juryon Paik received the B.E. degree in Information Engineering from Sungkyunkwan University, Korea, in 1997. She received her M.E. and Ph.D. degrees in Computer Engineering from Sungkyunkwan University in 2005 and 2008, respectively. Currently, she is a research professor at the Department of Computer Engineering, Sungkyunkwan University. Her research interests include XML mining, semantic mining, and web search engines.

Dongho Won received his B.E., M.E., and Ph.D. degrees from Sungkyunkwan University in 1976, 1978, and 1988, respectively. After working at ETRI (Electronics & Telecommunications Research Institute) from 1978 to 1980, he joined Sungkyunkwan University in 1982, where he is currently Professor of School of Information and Communication Engineering. In the year 2002, he served as the President of KIISC (Korea Institute of Information Security & Cryptology). He was the Program Committee Chairman of the 8th International Conference on Information Security and Cryptology (ICISC 2005). His research interests are on cryptology and information security.