Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 479534, 7 pages
http://dx.doi.org/10.1155/2014/479534

Research Article

On the Security of a Simple Three-Party Key Exchange Protocol without Server's Public Keys

Junghyun Nam,[1] Kim-Kwang Raymond Choo,[2] Minkyu Park,[1] Juryon Paik,[3] and Dongho Won[3]

[1] *Department of Computer Engineering, Konkuk University, 268 Chungwondaero, Chungju, Chungcheongbuk-do 380-701, Republic of Korea*
[2] *Information Assurance Research Group, Advanced Computing Research Centre, University of South Australia, Mawson Lakes, SA 5095, Australia*
[3] *Department of Computer Engineering, Sungkyunkwan University, 2066 Seobu-ro, Suwon, Gyeonggi-do 440-746, Republic of Korea*

Correspondence should be addressed to Dongho Won; dhwon@security.re.kr

Received 19 June 2014; Accepted 16 July 2014; Published 1 September 2014

Academic Editor: Moonseong Kim

Authenticated key exchange protocols are of fundamental importance in securing communications and are now extensively deployed for use in various real-world network applications. In this work, we reveal major previously unpublished security vulnerabilities in the password-based authenticated three-party key exchange protocol according to Lee and Hwang (2010): (1) the Lee-Hwang protocol is susceptible to a man-in-the-middle attack and thus fails to achieve implicit key authentication; (2) the protocol cannot protect clients' passwords against an offline dictionary attack; and (3) the indistinguishability-based security of the protocol can be easily broken even in the presence of a passive adversary. We also propose an improved password-based authenticated three-party key exchange protocol that addresses the security vulnerabilities identified in the Lee-Hwang protocol.

1. Introduction

One of the fundamental problems in the areas of cryptography and communication security is to enable two parties communicating over a public network to establish a high-entropy secret key (known as a *session key*) from their low-entropy passwords which are easy for humans to remember. Password-based authenticated key exchange (PAKE) protocols are designed to solve this problem and often assume the three-party setting, in which each party (commonly called a client) needs to remember only a single password shared with a trusted server [1–11]. The design of secure yet efficient three-party PAKE protocols is notoriously hard and continues to be a subject of active research. A key challenge in designing such protocols is to prevent potential attacks by a malicious client, who is registered with the server, and thus is able to set up normal protocol sessions with other clients.

In this work, we present previously unpublished flaws in the S-EA-3PAKE protocol, a three-party PAKE protocol proposed by Lee and Hwang [4]. The design of the S-EA-3PAKE protocol is relatively simple and efficient and carries a claimed proof of security in the ROR model according to Abdalla et al. [1]. However, despite the claim of provable security, this protocol exhibits major security weaknesses. First, the protocol fails to achieve *implicit key authentication*, which is the fundamental security property that any given key exchange protocol is expected to provide. We demonstrate this by mounting a man-in-the-middle attack against the protocol. The attacker could be any malicious client. Second, the protocol is vulnerable to an offline dictionary attack by a malicious client and thus other clients cannot be guaranteed of the security of their passwords. Third, the protocol does not achieve semantic security of session keys; that is, session keys established by S-EA-3PAKE are distinguishable from random

keys. We show this by mounting a passive attack in the ROR model, thereby invalidating the existing proof of security for S-EA-3PAKE. In addition to reporting the security vulnerabilities, we will also show how to fix the S-EA-3PAKE protocol so that it can achieve implicit key authentication as well as password security and semantic security.

Throughout the paper, we make the following assumptions on the capabilities of the adversary $\mathscr{A}$ in order to properly analyze the security properties of three-party PAKE protocols.

(i) $\mathscr{A}$ is either an outsider or an insider who runs in a probabilistic polynomial time.

(ii) $\mathscr{A}$ has the complete control of all message exchanges between the server and clients. That is, $\mathscr{A}$ can eavesdrop, insert, modify, intercept, and delete messages exchanged among the protocol participants at will.

This assumption is the standard one [12, 13] and is consistent with Dolev-Yao model.

2. The S-EA-3PAKE Protocol

The S-EA-3PAKE protocol [4] is built upon Abdalla and Pointcheval's 2-party PAKE protocol called SPAKE [14]. 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. The public parameters required by S-EA-3PAKE include

(i) a large prime p and a generator g of $\mathbb{Z}_p^*$,

(ii) two random elements M and N of $\mathbb{Z}_p^*$,

(iii) a cryptographic hash function H used as a key derivation function and

(iv) a pair of message authentication code (MAC) generation/verification algorithms $(\mathsf{Mac}, \mathsf{Ver})$, where Ver outputs a bit, with 1 meaning accept and 0 meaning reject.

S-EA-3PAKE is shown in Figure 1 and proceeds as follows.

Step 1. A sends S and B a protocol initiation message $M_{\text{init}} = \langle A, B\rangle$ which states "A wants to establish a session key with B."

Step 2. A and S establish a shared secret key k_{AS} by running the 2-party protocol SPAKE. Likewise, B and S establish a shared secret key k_{BS}. More precisely, k_{AS} and k_{BS} are established as shown in Table 1.

Step 3. A (resp. B) computes the authenticator. $\sigma_{AS} = \mathsf{Mac}_{k_{AS}}(A \parallel S)$ (resp. $\sigma_{BS} = \mathsf{Mac}_{k_{BS}}(B \parallel S)$) and sends it to S.

Step 4. S aborts if either $\mathsf{Ver}_{k_{AS}}(A \parallel S, \sigma_{AS}) = 1$ or $\mathsf{Ver}_{k_{BS}}(B \parallel S, \sigma_{BS}) = 1$ is untrue. Otherwise, S selects a random $s \in \mathbb{Z}_p^*$, computes

$$\begin{aligned} \overline{X} &= X^s, \\ \overline{Y} &= Y^s, \\ \overline{X}^* &= \overline{Y} \cdot k_{AS}, \\ \overline{Y}^* &= \overline{X} \cdot k_{BS}, \\ \sigma_{SA} &= \mathsf{Mac}_{k_{AS}}(s \parallel A)(s \parallel A), \\ \sigma_{SB} &= \mathsf{Mac}_{k_{BS}}(s \parallel B), \end{aligned} \tag{1}$$

and sends $\langle \overline{X}^*, \sigma_{SA}\rangle$ and $\langle \overline{Y}^*, \sigma_{SB}\rangle$ to A and B, respectively.

Step 5. A checks if $\mathsf{Ver}_{k_{AS}}(S \parallel A, \sigma_{SA}) = 1$ and aborts if the check fails. Otherwise, A computes the key derivation secret, $K_A = (\overline{X}^*/k_{AS})^x$, and the session key, $sk_A = H(A \parallel B \parallel K_A)$. Meanwhile, B checks if $\mathsf{Ver}_{k_{BS}}(S \parallel B, \sigma_{SB}) = 1$ and aborts if the check fails. Otherwise, B computes $K_B = (\overline{Y}^*/k_{BS})^y$ and $sk_B = H(A \parallel B \parallel K_B)$.

Step 6. A and B perform key confirmation by exchanging $\sigma_{AB} = \mathsf{Mac}_{sk_A}(A \parallel B)$ and $\sigma_{BA} = \mathsf{Mac}_{sk_B}(B \parallel A)$ and verifying them in a straightforward way.

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

3. Previously Unpublished Flaws

3.1. No Implicit Key Authentication. Implicit key authentication of S-EA-3PAKE can be violated via a man-in-the-middle attack by a malicious (registered) client C. A possible attack scenario is as follows.

(1) The attacker C blocks the protocol initiation message $M_{\text{init}} = \langle A, B\rangle$ from reaching S and instead sends (to S) two forged initiation messages $M'_{\text{init}} = \langle A, C\rangle$ and $M''_{\text{init}} = \langle C, B\rangle$ which state, respectively, "A wants to establish a session key with C" and "C wants to establish a session key with B." As a result, S will think that there are two protocol sessions running concurrently; let $\Pi_{A,C}$ denote the session between A and C and let $\Pi_{C,B}$ denote the session between C and B.

(2) In both sessions $\Pi_{A,C}$ and $\Pi_{C,B}$, C performs Step 2 through 5 as per the protocol specification with its true identity. This can go undetected since none of the authenticators, σ_{AS}, σ_{BS}, σ_{SA}, and σ_{SB}, can confirm who the actual protocol participants are. As a result, C will share a session key, $sk_{A,C}$, with A and another session key, $sk_{C,B}$, with B.

(3) With $sk_{A,C}$ and $sk_{C,B}$ in hand, C can perform Step 6 (of both sessions) in the straightforward way without being detected; C simply replaces $\sigma_{AB} = \mathsf{Mac}_{sk_{A,C}}(A \parallel B)$ and $\sigma_{BA} = \mathsf{Mac}_{sk_{C,B}}(B \parallel A)$ with $\sigma'_{AB} = \mathsf{Mac}_{sk_{C,B}}(A \parallel B)$ and $\sigma'_{BA} = \mathsf{Mac}_{sk_{A,C}}(B \parallel A)$, respectively.

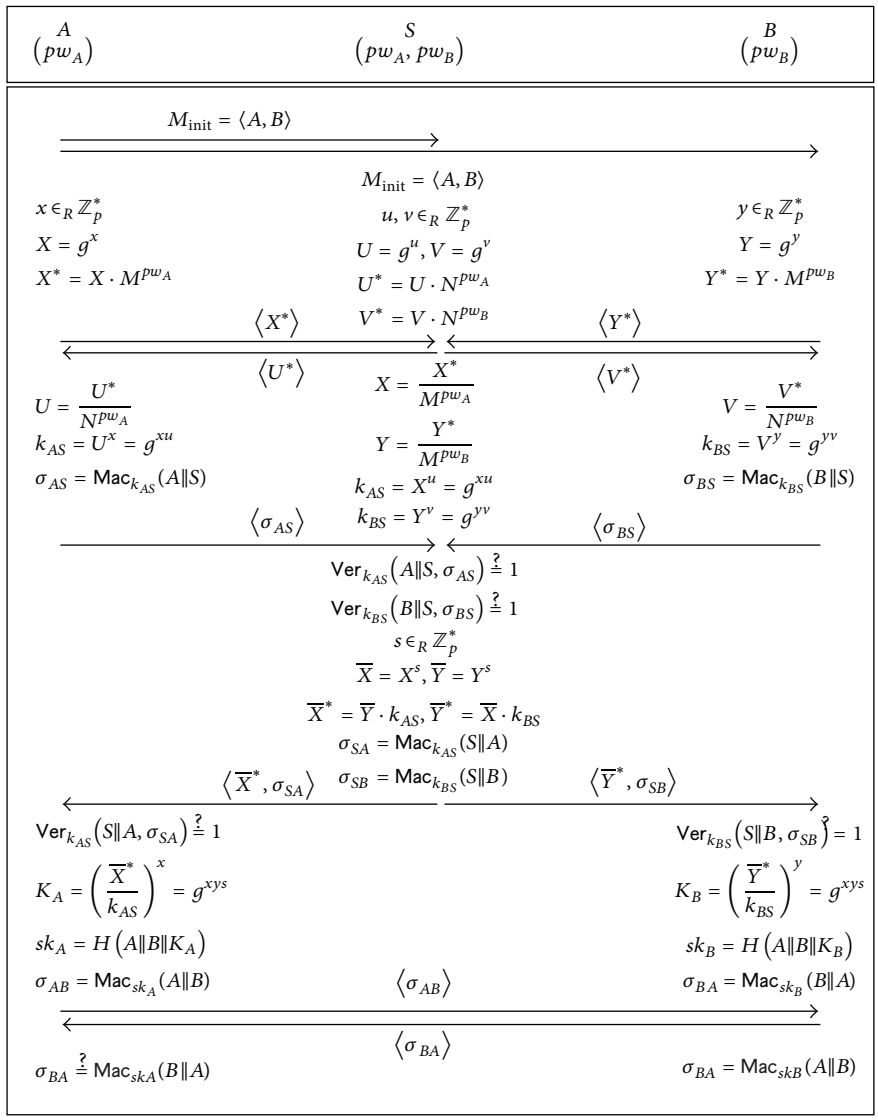

Figure 1: The S-EA-3PAKE protocol according to Lee and Hwang [4].

Table 1: Establishing the secret keys k_{AS} and k_{BS} in the S-EA-3PAKE protocol.

$pw_A \rightarrow k_{AS}$	$pw_B \rightarrow k_{BS}$
A chooses a random $x \in \mathbb{Z}_p^*$, computes $X = g^x$ and $X^* = X \cdot M^{pw_A}$, and sends X^* to S. At the same time, S chooses a random $u \in \mathbb{Z}_p^*$, computes $U = g^u$ and $U^* = U \cdot N^{pw_A}$, and sends U^* to A. Then, A and S set $k_{AS} = g^{xu}$.	B chooses a random $y \in \mathbb{Z}_p^*$, computes $Y = g^y$ and $Y^* = Y \cdot M^{pw_B}$, and sends Y^* to S. At the same time, S chooses a random $v \in \mathbb{Z}_p^*$, computes $V = g^v$ and $V^* = V \cdot N^{pw_B}$, and sends V^* to B. Then, B and S set $k_{BS} = g^{yv}$.

At the end of the attack scenario, A and B believe that they have established a secure session with each other sharing a key, while in fact they have shared their keys with the attacker C. Consequently, S-EA-3PAKE fails to achieve implicit key authentication.

3.2. No Password Security. We now show that S-EA-3PAKE cannot protect clients' passwords against an offline dictionary attack. Assume a malicious client C who wants to find out the passwords of A and B. Let pw_C be the password of C. Then, an offline dictionary attack by C against both A and B can be mounted as follows.

Phase 1 (gathering password verifiers online). C conducts a type of man-in-the-middle attack to obtain information required to verify password guesses.

(1) C blocks the initiation message $M_{\text{init}} = \langle A, B \rangle$ from reaching S and instead sends two forged initiation messages $M'_{\text{init}} = \langle A, C \rangle$ and $M''_{\text{init}} = \langle C, B \rangle$, thereby

deceiving S into thinking that there are two protocol sessions, $\Pi_{A,C}$ and $\Pi_{C,B}$, running concurrently.

(2) C then performs Step 2 through 5 of both sessions as specified by the protocol except for the following.

(i) When A and B send $X^* = X \cdot M^{pw_A}$ and $Y^* = Y \cdot M^{pw_B}$ in Step 2, C makes a copy of these messages for later use.

(ii) C sends the same Step 2 message $Z^* = g^z \cdot M^{pw_C}$ of its own for both sessions, where $z \in_R \mathbb{Z}_p^*$.

(iii) When S sends $\langle \overline{X}^*, \sigma_{SA} \rangle$ and $\langle \overline{Y}^*, \sigma_{SB} \rangle$, respectively, to A and B in Step 4 of the sessions, C replaces $\overline{X}^*$ and $\overline{Y}^*$ with $\widehat{X}^* = \overline{X}^* \cdot g^z$ and $\widehat{Y}^* = \overline{Y}^* \cdot g^z$, respectively.

Let $\langle \overline{Z}^*, \sigma_{SC} \rangle$ and $\langle \overline{Z'}^*, \sigma'_{SC} \rangle$ denote the two messages sent by S to C in Step 4 of $\Pi_{A,C}$ and $\Pi_{C,B}$, respectively.

(3) Now when A and B exchange the key confirmation messages $\sigma_{AB} = \mathsf{Mac}_{sk_A}(A \parallel B)$ and $\sigma_{BA} = \mathsf{Mac}_{sk_B}(B \parallel A)$, C intercepts these messages and instead sends the clients "a failure message" to trick them into believing that, due to an unexpected error, their partner has failed to compute the session key and thus has aborted the protocol.

Phase 2 (verifying password guesses offline). C can now verify password guesses both on pw_A and pw_B using the obtained information $(X^*, \overline{Z}^*, \sigma_{AB})$ and $(Y^*, \overline{Z'}^*, \sigma_{BA})$, respectively. (For simplicity, we here describe this verification phase only for pw_A; the case for pw_B proceeds correspondingly).

Step 1. C computes

$$K_C = \left(\frac{\overline{Z}^*}{k_{CS}} \right)^z = \overline{X}^z = g^{xzs}, \tag{2}$$

where k_{CS} is the secret key shared between C and S in Step 2 of session $\Pi_{A,C}$.

Step 2. Note that, since $\overline{X}^*$ was replaced with $\widehat{X}^* = \overline{X}^* \cdot g^z$, A must have computed K_A as

$$\begin{aligned} K_A &= \left(\frac{\widehat{X}^*}{k_{AS}} \right)^x = \left(\frac{\overline{X}^* \cdot g^z}{k_{AS}} \right)^x = \left(g^{zs} \cdot g^z \right)^x \\ &= g^{xzs} \cdot g^{xz}. \end{aligned} \tag{3}$$

With this in mind, C makes a guess pw'_A on the password pw_A and computes

$$\begin{aligned} X' &= \frac{X^*}{M^{pw'_A}}, \\ K'_A &= K_C \cdot X'^z, \\ sk'_A &= H\left(A \| B \| K'_A \right), \\ \sigma'_{AB} &= \mathsf{Mac}_{sk'_A}(A \| B). \end{aligned} \tag{4}$$

Step 3. A verifies the correctness of pw'_A by checking that σ_{AB} is equal to σ'_{AB}. If they are equal, then pw'_A is the correct password with an overwhelming probability.

Step 4. C repeats Steps 2 and 3 (of this verification phase) until a correct password is found.

This offline dictionary attack can be trivially simplified to an insider-attacker version whereby one of the two clients, A and B, tries to discover the other client's password. After all, the S-EA-3PAKE protocol cannot prevent any (malicious) client from mounting an offline dictionary attack against any other clients.

3.3. No Semantic Security. Finally, we point out that the S-EA-3PAKE protocol does not achieve the semantic security of session keys. In S-EA-3PAKE, the session key sk_A (resp. sk_B) is used as the MAC key in generating the authenticator $\sigma_{AB} = \mathsf{Mac}_{sk_A}(A \parallel B)$ (resp. $\sigma_{BA} = \mathsf{Mac}_{sk_B}(B \parallel A)$). This oversight leaks some information about the session key and allows an adversary to distinguish the real session key from a random key chosen from the session key space. Indeed, S-EA-3PAKE can be easily broken even in the presence of a passive adversary who asks only a single Execute and Test query. A simple attack by such an adversary $\mathcal{A}$ can be described as follows.

(1) First, $\mathcal{A}$ makes an $\mathsf{Execute}(\Pi_A^*, \Pi_B^*, \Pi_S^*)$ query, where Π_A^*, Π_B^*, and Π_S^* denote any instance of A, B, and S, respectively. This query prompts an honest execution of the protocol between the three instances and will return the transcript of the protocol execution.

(2) Next, $\mathcal{A}$ makes a $\mathsf{Test}(\Pi_A^*)$ query and receives a key $\overline{sk}$ in response to the query.

(3) Then, $\mathcal{A}$ computes $\sigma'_{AB} = \mathsf{Mac}_{\overline{sk}}(A \parallel B)$ and checks if σ'_{AB} is equal to σ_{AB}. The key $\overline{sk}$ is real if they are equal and otherwise it is random.

This attack invalidates the existing proof of security for S-EA-3PAKE [4]. We refer the reader to the work of Bellare et al. [13] for a possible countermeasure.

4. An Improved Three-Party PAKE Protocol

In this section, we propose an improved three-party PAKE protocol which achieves semantic security and is secure against man-in-the-middle attacks as well as offline dictionary attacks. Let S be the trusted server and let 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. Our improved protocol uses the following public parameters:

(i) a finite cyclic group $\mathbb{G}$ of prime order q and a generator g of $\mathbb{G}$;

(ii) two random elements M and N of $\mathbb{G}$;

(iii) a cryptographic hash function $H : \{0,1\}^* \to \{0,1\}^\ell$, where ℓ represents the bit length of session keys;

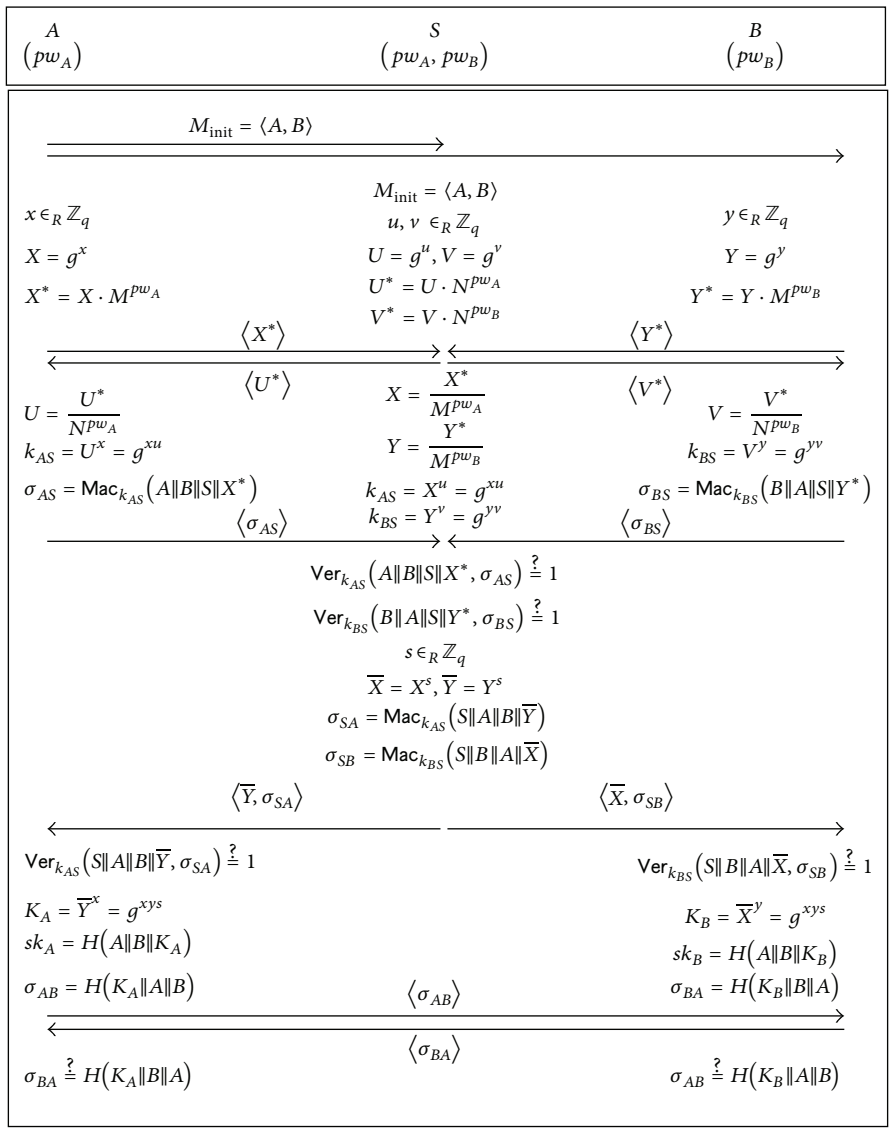

Figure 2: Our improved three-party PAKE protocol.

Table 2: Establishing the secret keys k_{AS} and k_{BS} in our improved protocol.

$pw_A \rightarrow k_{AS}$	$pw_B \rightarrow k_{BS}$
A chooses a random $x \in \mathbb{Z}_q$, computes $X = g^x$ and $X^* = X \cdot M^{pw_A}$, and sends X^* to S. At the same time, S chooses a random $u \in \mathbb{Z}_q$, computes $U = g^u$ and $U^* = U \cdot N^{pw_A}$, and sends U^* to A. Then, A and S set $k_{AS} = g^{xu}$.	B chooses a random $y \in \mathbb{Z}_q$, computes $Y = g^y$ and $Y^* = Y \cdot M^{pw_B}$, and sends Y^* to S. At the same time, S chooses a random $v \in \mathbb{Z}_q$, computes $V = g^v$ and $V^* = V \cdot N^{pw_B}$, and sends V^* to B. Then, B and S set $k_{BS} = g^{yv}$.

(iv) a pair of message authentication code (MAC) generation/verification algorithms (Mac, Ver), where Ver outputs a bit, with 1 meaning accept and 0 meaning reject.

The improved protocol is illustrated in Figure 2 and its description is as follows.

Step 1. A sends S and B a protocol initiation message $M_{\text{init}} = \langle A, B\rangle$ which states "A wants to establish a session key with B."

Step 2. A and S establish a shared secret key k_{AS} by running the 2-party protocol SPAKE. Likewise, B and S establish a shared secret key k_{BS}. More precisely, k_{AS} and k_{BS} are

established as shown in Table 2. (Note in Table 2 that all the random exponents are selected from $\mathbb{Z}_q$ as our protocol works in a group of prime order q.)

Step 3. A (resp. B) computes the authenticator $\sigma_{AS} = \mathsf{Mac}_{k_{AS}}(A \parallel B \parallel S \parallel X^*)$ (resp. $\sigma_{BS} = \mathsf{Mac}_{k_{BS}}(B \parallel A \parallel S \parallel Y^*)$) and sends it to S.

Step 4. S aborts if either $\mathsf{Ver}_{k_{AS}}(A \parallel B \parallel S \parallel X^*, \sigma_{AS}) = 1$ or $\mathsf{Ver}_{k_{BS}}(B \parallel A \parallel S \parallel Y^*, \sigma_{BS}) = 1$ is untrue. Otherwise, S selects a random $s \in \mathbb{Z}_q$, computes

$$\begin{aligned} \overline{X} &= X^s, \\ \overline{Y} &= Y^s, \\ \sigma_{SA} &= \mathsf{Mac}_{k_{AS}}\left(S \| A \| B \| \overline{Y}\right), \\ \sigma_{SB} &= \mathsf{Mac}_{k_{BS}}\left(S \| A \| B \| \overline{X}\right), \end{aligned} \tag{5}$$

and sends $\langle \overline{Y}, \sigma_{SA} \rangle$ and $\langle \overline{X}, \sigma_{SB} \rangle$ to A and B, respectively.

Step 5. A checks if $\mathsf{Ver}_{k_{AS}}(S \parallel A \parallel B \parallel \overline{Y}, \sigma_{SA}) = 1$ and aborts if the check fails. Otherwise, A computes the key derivation secret, $K_A = \overline{Y}^x$, and the session key, $sk_A = H(A \parallel B \parallel K_A)$. Meanwhile, B checks if $\mathsf{Ver}_{k_{BS}}(S \parallel B \parallel A \parallel \overline{X}, \sigma_{SB}) = 1$ and aborts if the check fails. Otherwise, B computes $K_B = \overline{X}^y$ and $sk_B = H(A \parallel B \parallel K_B)$.

Step 6. A and B perform key confirmation by exchanging $\sigma_{AB} = H(K_A \parallel A \parallel B)$ and $\sigma_{BA} = H(K_B \parallel B \parallel A)$ and verifying them in a straightforward way.

It can be easily verified that A and B compute session keys of the same value since $K_A = K_B = g^{xys}$. Compared with the S-EA-3PAKE protocol, our improved protocol does not require the computations of $\overline{X}^*$ and $\overline{Y}^*$ and simplifies the computations of K_A and K_B. Therefore, it is fair to say that our protocol performs slightly better than the S-EA-3PAKE protocol.

Man-in-the-middle attacks and offline dictionary attacks such as the ones we mounted against the S-EA-3PAKE protocol are no longer valid against our improved protocol since the authenticators, σ_{AS}, σ_{BS}, σ_{SA}, and σ_{SB}, can now confirm who the actual protocol participants are. Moreover, our protocol achieves semantic security as the key derivation secrets K_A and K_B instead of the session keys sk_A and sk_B are used in generating the authenticators $\sigma_{AB} = H(K_A \parallel A \parallel B)$ and $\sigma_{BA} = H(K_B \parallel B \parallel A)$.

5. Concluding Remarks

The model where S-EA-3PAKE was claimed to be provably secure does not allow the adversary to ask Corrupt queries and thus cannot capture any kind of attacks that can be mounted by malicious clients. Accordingly, neither the man-in-the-middle attack nor the offline dictionary attack described in this work can be captured in the proof model. This situation is clearly unacceptable, from both theoretic and practical perspectives, and highlights the importance of considering Corrupt queries when proving security of three-party PAKE protocols. Although both the man-in-the-middle attack and the dictionary attack can be easily prevented by modifying the computations of the authenticators σ_{AS}, σ_{BS}, σ_{SA}, and σ_{SB}, the existence of a security proof for the S-EA-3PAKE protocol in a stronger model remains an open question. We finally note that all the three attacks presented in this work against S-EA-3PAKE also apply to the S-IA-3PAKE protocol [4], a simplified variant of S-EA-3PAKE. This becomes clear as soon as we notice that S-IA-3PAKE is different from S-EA-3PAKE only in the fact that it does not require the transmission of the authenticators σ_{AS}, σ_{BS}, σ_{SA}, and σ_{SB}.

Conflict of Interests

The authors of the paper do not have a direct financial relation with any institution or organization mentioned in their paper that might lead to a conflict of interests for any of the them.

Acknowledgment

This work was supported by Konkuk University.

References

[1] M. Abdalla, P. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Public Key Cryptography-PKC 2005*, vol. 3386 of *Lecture Notes in Computer Science*, pp. 65–84, Springer, Berlin, Germany, 2005.

[2] J. Nam, Y. Lee, S. Kim, and D. Won, "Security weakness in a three-party pairing-based protocol for password authenticated key exchange," *Information Sciences*, vol. 177, no. 6, pp. 1364–1375, 2007.

[3] J. Nam, J. Paik, H. Kang, U. M. 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.

[4] 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.

[5] T. Chang, M. Hwang, and W. Yang, "A communication-efficient three-party password authenticated key exchange protocol," *Information Sciences*, vol. 181, no. 1, pp. 217–226, 2011.

[6] J. Zhao and D. Gu, "Provably secure three-party password-based authenticated key exchange protocol," *Information Sciences*, vol. 184, pp. 310–323, 2012.

[7] S. Wu, Q. Pu, S. Wang, and D. He, "Cryptanalysis of a communication-efficient three-party password authenticated key exchange protocol," *Information Sciences*, vol. 215, pp. 83–96, 2012.

[8] C. Lee, S. Chen, and C. Chen, "A computation-efficient three-party encrypted key exchange protocol," *Applied Mathematics & Information Sciences*, vol. 6, no. 3, pp. 573–579, 2012.

[9] S. Wu, K. Chen, Q. Pu, and Y. Zhu, "Cryptanalysis and enhancements of efficient three-party password-based key exchange scheme," *International Journal of Communication Systems*, vol. 26, no. 5, pp. 674–686, 2013.

[10] J. Nam, K. K. R. Choo, M. Kim, J. Paik, and D. Won, "Dictionary attacks against password-based authenticated three-party key exchange protocols," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 12, pp. 3244–3260, 2013.

[11] J. Nam, K. K. R. Choo, J. Kim et al., "Password-only authenticated three-party key exchange with provable security in the standard model," *The Scientific World Journal*, vol. 2014, Article ID 825072, 11 pages, 2014.

[12] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Advances in Cryptology—(CRYPTO '93)*, vol. 773 of *Lecture Notes in Computer Science*, pp. 232–249, Springer, Berlin, Germany, 1994.

[13] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology—EUROCRYPT 2000*, vol. 1807 of *Lecture Notes in Computer Science*, pp. 139–155, Springer, Berlin, Germany, 2000.

[14] M. Abdalla and D. Pointcheval, "Simple password-based encrypted key exchange protocols," in *Topics in Cryptology—CT-RSA 2005*, vol. 3376 of *Lecture Notes in Computer Science*, pp. 191–208, Springer, Berlin, Germany, 2005.