The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.75 e-ISSN: 2251-7545

Cryptanalysis ofthree Provably Secure Password Authenticated Key Exchange Protocols in the Three-party Setting

Maryam Saeed
Iran University of Sceince & Technology
Tehran, Iran
m_saeed@vu.iust.ir

Hadi Shahriar Shahhoseini
Iran University of Sceince & Technology
Tehran, Iran
hshsh@iust.ac.ir

Ali Mackvandi
R&D of Pishgaman Kavir Company
Yazd, Iran
Mackvandi@pishgaman.com

Mohammad Reza Rezaeinezhad
Pishgaman Kavir Company
Yazd, Iran
rezaei@pishgaman.com

Mansour Naddafiun
Pishgaman Kavir Company
Yazd, Iran
addafiun@pishgaman.com

***Abstract*—Three-party Password Authenticated Key Exchange (3PAKE) protocols play a key role in providing security goals in communications. They enable two entities to share a common session key in an authentic manner based on a low entropy human-memorable password. In 2010, Lee and Hwang proposed S-IA-3PAKE and S-EA-3PAKE protocols based on the SPAKE protocol developed by Abdalla and Pointcheval. In 2011, Chang et al. presented an efficient three-party Password Authenticated Key Exchange Protocol and its parallel version based on LHL-3PAKE protocol proposed by Lee et al. In this paper, it is shown that both supposedly provably secure S-IA-3PAKE and S-EA-3PAKE protocols are vulnerable to serious threats such as Unknown Key Share (UKS) and password compromise impersonation attacks. It is also shown that the provably secure protocol of Chang et al. and its parallel version suffer from password compromise impersonation and ephemeral key compromise impersonation attacks. Indeed, our results highlight the need of more attention and precision during defining the provable security models and constructing proofs in this method, because there are still considerable gaps between what can be proven based on formal security models and what are actually secure in use.**

Keywords-Password Authenticated Key Exchange; Cryptanalysis; Unknown Key Share attack (UKS); ephemeral key compromise impersonation attack; password compromise impersonation attack.

I. INTRODUCTION

Securing communications over the Internet has been a considerable concern for protocol designers over the past years. One of the prevalent approaches for solving this problem is applying two-party Password-based Authenticated Key Exchange (2PAKE) protocol. In 2PAKE protocols, two entities try to share a common symmetric session key based on a low-entropy human-memorable password. Axiomatically, the first proposed 2PAKE protocol is referred to Bellovin and Merritt in 1992 [1]. Afterwards, 2PAKEs considerably have been expanded to 3PAKEs with slightly different in comparison with 2PAKEs. In a 3PAKE protocol, participants share their secret passwords with a trusted server or Key Distribution Center (KDC). Consequently, the server or KDC authenticates users with their pre-shared passwords and every user can exchange his/her session key with the intended user securely via the trusted server or KDC. Over the recent years, many 3PAKEs have been devised and proposed in which the seminal ones can be referred to [2-10]. The security analysis of 3PAKEs in formal model can be referred to Abdalla et al. [2] who extended the work of Bellare and Rogaway [11-12] that was designed for 2PAKE protocols.

It is perspicuously conspicuous that the protocol designers should intelligibly pinpoint what attacks their proposed protocol must resist and what kind of desirable security properties it must possess because there are several protocols that are proved to be insecure while their designers believed in security proof of their protocols in a formal security model [7], [13-16]. Consequently, based on the [17-24], it is essential for 3PAKEs to provide the following desirable security attributes:

- **Forward secrecy:** The forward secrecy is provided if the secrecy of previously established session keys are not divulged by compromising of any entity's the password or long-term private keys.
- **Known session key security:** Compromising of one session key should not jeopardize the security of other session keys.
- **Resilience to Unknown Key Share attack (UKS):** User *A* should not be compelled into sharing a session key with an adversary *E* after completion of a protocol run while *A* falsely thinks that his/her key is shared with another user *B*.
- **Resilience to password compromise impersonation attack:** Disclosure of any user *A*'s password should not

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.75 e-ISSN: 2251-7545

allow an adversary to share any session key with *A* by masquerading him- or herself as any other entity.

- **Resilience to ephemeral key compromise impersonation attack:** Some protocols deploy some random parameters as the ephemeral keys. Disclosure of any user *A*'s ephemeral key should not enable an adversary to establish a session key with *A* by impersonating him- or herself as any other participant.

In 2010, Lee and Hwang [25] proposed S-IA-3PAKE and S-EA-3PAKE protocols, which are based on the SPAKE protocol developed by Abdalla and Pointcheval [26], and also claimed that the security and efficiency of the proposed protocols are proved in the random oracle model and the security and efficiency of their protocols have great improvement in comparison with other 3PAKEs. In 2011, Chang et al. [27] presented a communication-efficient three-party Password Authenticated Key Exchange Protocol and its parallel version, which are based on LHL-3PAKE protocol proposed by Lee et al [4]. For the sake of simplicity, in this paper, we refer to Chang et al.'s protocol as CHY-3PAKEv1 and its parallel version as CHY-3PAKEv2. It is notable that the security and efficiency of CHY-3PAKEv1 and CHY-3PAKEv2 protocols are grounded on the computational Diffie–Hellman assumption in the random oracle model.

In this paper, it is shown that both provably secure S-IA-3PAKE and S-EA-3PAKE protocols suffer from serious threats such as Unknown Key Share (UKS) and password compromise impersonation attacks. It is also shown that the so-called secure CHY-3PAKEv1 and CHY-3PAKEv2 protocols are vulnerable to password compromise impersonation and ephemeral key compromise impersonation attacks. In fact, defining a proper provable security model is not a simple task since not considering some kinds of queries, e.g. the Corrupt query [28],[29], or incorrectly defining the adversarial game [30] may cause a proof of security that fails to take into account some important attacks (for more details, see [30],[28],[29],[31],[7], [16].

The rest of the paper is organized as follows. Section II explicates a brief review on S-IA-3PAKE, S-EA-3PAKE and CHY-3PAKE protocols and the notation used hereinafter, whereas the security vulnerabilities of S-IA-3PAKE and S-EA-3PAKE protocols are pinpointed in Section III. Section IV also elucidates the security flaws of CHY-3PAKE protocols. Finally, Section V concludes the paper.

II. A Brief Review On CHY-3pake , S-IA-3pake And S-EA-3PAKE Protocols

This section briefly presents the S-IA-3PAKE, S-EA-3PAKE and CHY-3PAKEv1 and CHY-3PAKEv2 protocols [25], [27] which need neither the server public keys nor symmetric encryption/decryption system. The S-IA-3PAKE and S-EA-3PAKE protocols, which are based on the SPAKE protocol developed by Abdalla and Pointcheval [26], use implicit server authentication and explicit server authentication, respectively, and CHY-3PAKEv1 and its parallel version, CHY-3PAKEv2, are based on LHL-3PAKE proposed by Lee et al [4]. It is assumed that there are three parties involved in the mentioned protocols: a trusted authentication server S, and two users A and B who share low-entropy passwords pw_1 and pw_2 in $\mathbb{Z}_p^*$ with server S, respectively and want to establish a common secret session key. Figures 1, 2 , 3 and 4 illustrate the S-IA-3PAKE, S-EA-3PAKE and CHY-3PAKEv1 and CHY-3PAKEv2 protocols, respectively, in which the deployed notations are explained in Table I.

TABLE I. Deployed Notations

Notations	Definition
A	The user *A*'s identifier.
B	The user *B*'s identifier.
C	The malicious user *C*'s identifier.
S	The trusted server *S*'s identifier.
pw_1	The shared password between A and S.
pw_2	The shared password between B and S.
pw_3	The shared password between C and S.
x	A random number generated by A.
y	A random number generated by B.
z, z_1, z_2	Random numbers generated by S.
$H(.)$	Collision-resistant one-way hash function.
p	Sufficiently large prime.
g	The generator of $GF(p)$.
M, N	Two elements in G, a finite cyclic group generated by an element g of prime order p.
SK_{AB}	Session key

A. Description of the S-IA-3PAKE protocol

The S-IA-3PAKE protocol [25] is a tripartite password authenticated key exchange protocol that does not supply the server authentication of user A and B during the protocol run. The structure of the S-IA-3PAKE protocol is depicted in Figure 1, and the detailed steps are described as follows:

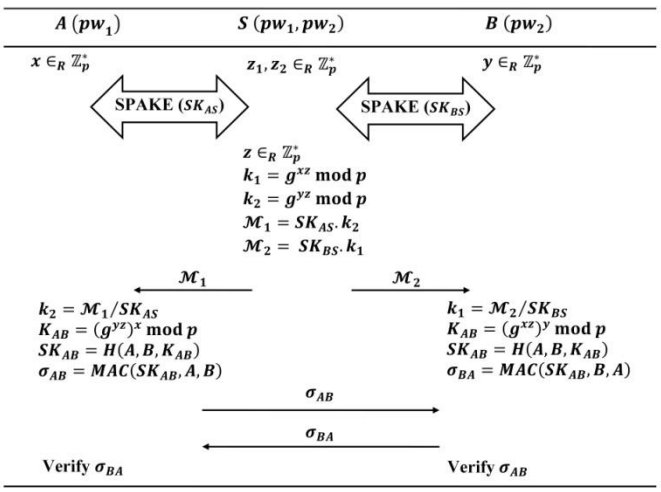

Figure 1. S-IA-3PAKE protocol [25]

Step 1:

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.75 e-ISSN: 2251-7545

The user A and server S first select two random numbers $x \in_R \mathbb{Z}_p^*$ and $z_1 \in_R \mathbb{Z}_p^*$, respectively, run SPAKE protocol and derive the same session key $SK_{AS} = g^{xz_1} \bmod p$. A more detailed description of the Step 1 is as follows:

(1) The user A chooses a random number $x \in_R \mathbb{Z}^*{}_p$, computes $X = g^x \bmod p$ and $X^* = X.M^{pw_1}$ and sends X^* to the server S.
(2) After receiving X^*, S selects a random number $z_1 \in_R \mathbb{Z}_p^*$ and calculates $X = X^*/M^{pw_1}$, $SK_{AS} = (X)^{Z_1} = (g^x)^{Z_1} \bmod p$, $Z_1 = g^{z_1} \bmod p$ and $Z_1^* = Z_1.N^{pw_1}$, then sends $Z_1{}^*$ to A.
(3) When A receives Z_1^*, s/he computes $Z_1 = Z_1^*/N^{pw_1}$, $SK_{AS} = (Z_1)^x = (g^{Z_1})^x \bmod p$.

Step 2:
In the same way, the user B and server S choose two random numbers $y \in_R \mathbb{Z}_p^*$ and $z_2 \in_R \mathbb{Z}_p^*$, respectively, run the SPAKE protocol and establish the common session key $SK_{BS} = g^{yz_2} \bmod p$.

Step 3:
S selects another random number $z \in_R \mathbb{Z}_p^*$, computes $\mathcal{M}_1 = SK_{AS}.k_2$ where $k_{2=}g^{yz} \bmod p$ and $\mathcal{M}_2 = SK_{BS}.k_1$ where $k_1 = g^{xz} \bmod p$ and distributes $\mathcal{M}_1$ and $\mathcal{M}_2$ to A and B, respectively.

Step 4:
A and B calculate $k_2 = \mathcal{M}_1/SK_{AS}, K_{AB} = (k_2)^x = (g^{yz})^x \bmod p$ and $k_1 = \mathcal{M}_2/SK_{BS}, K_{BA} = (k_1)^y = (g^{xz})^y \bmod p$, respectively. Consequently, they obtain a common value $K_{AB} = g^{xyz} \bmod p$ and derive the same session key $SK_{AB} = H(A, B, K_{AB})$.

Step 5:
A and B separately calculate their parameters of authenticator $\sigma_{AB} = MAC(SK_{AB}, A, B)$ and $\sigma_{BA} = MAC(SK_{AB}, B, A)$, respectively and send them to the intended user. Then, A and B check the validity of the verifier messages σ_{BA} and σ_{AB}, which are used for confirming that users A and B have possession of a same session key, respectively.

B. Description of the S-EA-3PAKE protocol

The S-EA-3PAKE protocol [25] is a 3PAKE protocol that provides explicit server authentication. Therefore, the server directly authenticates the users A and B. The detailed steps of the S-EA-3PAKE protocol, as illustrated in Figure 2, are explained as follows:

Step 1:
The user A and server S first choose two random numbers $x \in_R \mathbb{Z}_p^*$ and $z_1 \in_R \mathbb{Z}_p^*$, respectively, run the SPAKE protocol and share the common session key $SK_{AS} = g^{xz_1} \bmod p$. The more details of this step is the same as what described in Step 1 of S-IA-3PAKE protocol.

Step 2:
Similarly, the user B and server S select two random numbers $y \in_R \mathbb{Z}_p^*$ and $z_2 \in_R \mathbb{Z}_p^*$, respectively, execute the SPAKE protocol and compute the same session key $SK_{BS} = g^{yz_2} \bmod p$.

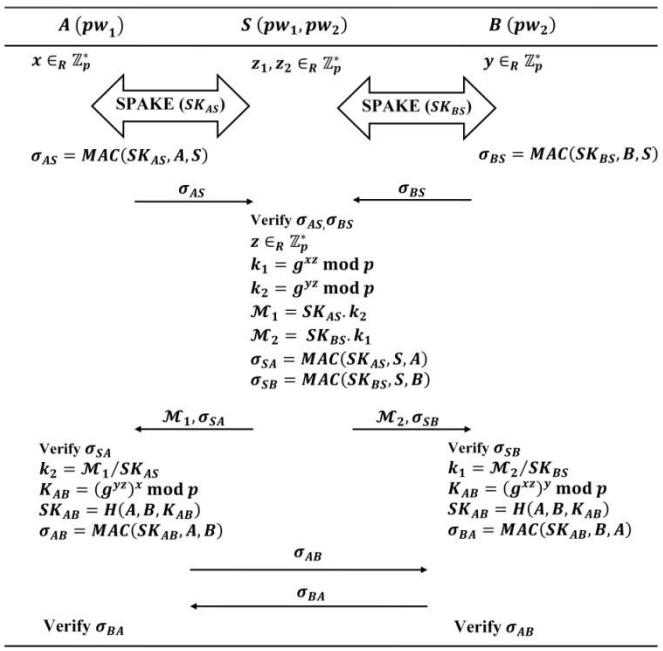

Figure 2. S-EA-3PAKE protocol [25]

Step 3:
The users A and B compute and issue the values of authenticator $\sigma_{AS} = MAC(SK_{AS}, A, S)$ and $\sigma_{BS} = MAC(SK_{BS}, B, S)$ to S, respectively.

Step 4:
Upon receiving σ_{AS} and σ_{BS}, if S successfully confirm that σ_{AS} and σ_{BS} are valid, then S selects another random number $z \in_R \mathbb{Z}_p^*$ and calculates $\sigma_{SA} = MAC(SK_{AS}, S, A)$, $\sigma_{SB} = MAC(SK_{BS}, S, B)$, $\mathcal{M}_1 = SK_{AS}.k_2$ in which $k_{2=}g^{yz} \bmod p$ and $\mathcal{M}_2 = SK_{BS}.k_1$ in which $k_1 = g^{xz} \bmod p$. Finally, S sends σ_{SA}, $\mathcal{M}_1$ and σ_{SB}, $\mathcal{M}_2$ to A and B, respectively.

Step 5:
After receiving the messages from S, if A and B verify the validity of authenticators σ_{SA} and σ_{SB}, respectively, then A and B calculate $K_{AB} = (k_2)^x = (g^{yz})^x \bmod p$ where $k_2 = \mathcal{M}_1/SK_{AS}$ and $K_{BA} = (k_1)^y = (g^{xz})^y \bmod p$ where $k_1 = \mathcal{M}_2/SK_{BS}$, respectively. Therefore, the corresponding users derive the same value $K_{AB} = g^{xyz} \bmod p$ and establish the common session key $SK_{AB} = H(A, B, K_{AB})$.

Step 6:
A and B compute and issue their parameters of authenticator $\sigma_{AB} = MAC(SK_{AB}, A, B)$ and $\sigma_{BA} = MAC(SK_{AB}, B, A)$ to the opposite side, respectively. At last, A

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.75 e-ISSN: 2251-7545

and B check the validity of the authenticator's values σ_{BA} and σ_{AB}, respectively.

C. Description of the CHY-3PAKE protocols

CHY-3PAKE protocols consist of two versions: CHY-3PAKEv1 and CHY-3PAKEv2 [27]. CHY-3PAKEv2 is the parallel version of CHY-3PAKEv1that its steps are reordered , but its fundamental structure and the contents of transmitted messages are similar to those in CHY-3PAKEv1 protocol. CHY-3PAKEv1 and CHY-3PAKEv2 are illustrated in Figures 3 and 4. The steps of the CHY-3PAKEv1 protocol are described as follows:

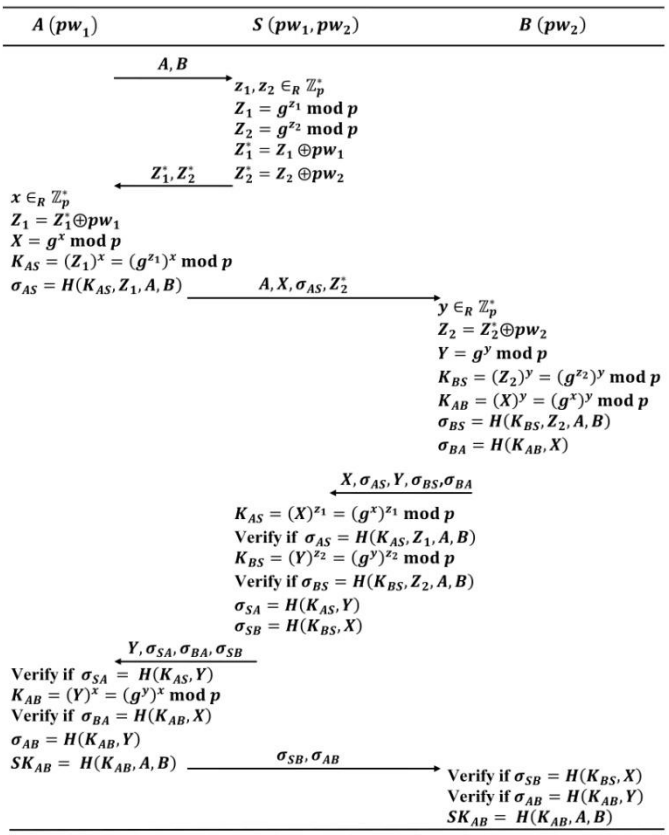

Figure 3. CHY-3PAKEv1 protocol [27]

Step 1:
The user A initiates the protocol by sending A and B to the server S.

Step 2:
Based on identifiers A and B from the incoming message, S can recall the passwords pw_1 and pw_2 from its password database, then selects two random numbers $z_1 \in_R \mathbb{Z}_p^*$ and $z_2 \in_R \mathbb{Z}_p^*$, computes $Z_1 = g^{z_1} \bmod p$, $Z_2 = g^{z_2} \bmod p$, $Z_1^* = Z_1 \oplus pw_1$ and $Z_2^* = Z_2 \oplus pw_2$. Finally, S sends Z_1^* and Z_2^* to A.

Step 3:
The user A selects a random number x $\in_R \mathbb{Z}_p^*$, obtains Z_1 as $Z_1 = Z_1^* \oplus pw_1$, calculates $X = g^x \bmod p$, $K_{AS} = (Z_1)^x = (g^{z_1})^x \bmod p$ and $\sigma_{AS} = H(SK_{AS}, Z_1, A, B)$ and then sends A, X, σ_{AS} and Z_2^* to B.

Step 4:
Upon receiving the message from A, the user B selects a random number y $\in_R \mathbb{Z}_p^*$ and acquires Z_2 as $Z_2 = Z_2^* \oplus pw_2$, computes $Y = g^y \bmod p$, $K_{BS} = (Z_2)^y = (g^{z_2})^y \bmod p$ and $K_{AB} = (X)^y = (g^x)^y \bmod p$, $\sigma_{BS} = (K_{BS}, Z_2, A, B)$ and $\sigma_{BA} = H(K_{AB}, X)$. Finally, B sends $X, \sigma_{AS}, Y, \sigma_{BS}$ and σ_{BA} to S.

Step 5:
To separately authenticate A and B, S first constructs $K_{AS} = (X)^{z_1} = (g^x)^{z_1} \bmod p$ and $K_{BS} = (g^y)^{z_2} \bmod p$, then verifies if $\sigma_{AS} = H(K_{AS}, Z_1, A, B)$ and $\sigma_{BS} = H(K_{BS}, Z_2, A, B)$ or not, . If the both equalities are satisfied, the authentication of users A and B is done successfully and S calculates his/her authenticators $\sigma_{SA} = H(K_{AS}, Y)$ and $\sigma_{SB} = H(K_{BS}, X)$ and issues $Y, \sigma_{SA}, \sigma_{BA}$ and σ_{SB} to A.

Step 6:
A first checks if $\sigma_{SA} = H(K_{AS}, Y)$ or not. If it is not verified, the authentication fails. Otherwise, A computes $K_{AB} = (Y)^x = (g^y)^x \bmod p$, then verifies if $\sigma_{BA} = H(K_{AB}, X)$ or not, if it is hold, A calculates $\sigma_{AB} = H(K_{AB}, Y)$ and generates the session key as $SK_{AB} = H(K_{AB}, A, B)$ in which $K_{AB} = (g^y)^x \bmod p$. Finally, A sends σ_{SB} and σ_{AB} to B.

Step 7:
Upon receiving the message from A, B checks if $\sigma_{SB} = H(K_{BS}, X)$ and $\sigma_{AB} = H(K_{AB}, Y)$ or not. If both equalities are hold, B constructs the session key as $SK_{AB} = H(K_{AB}, A, B)$ in which $K_{AB} = (g^x)^y \bmod p$ and also believes that A has obtained the common session key.

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.75 e-ISSN: 2251-7545

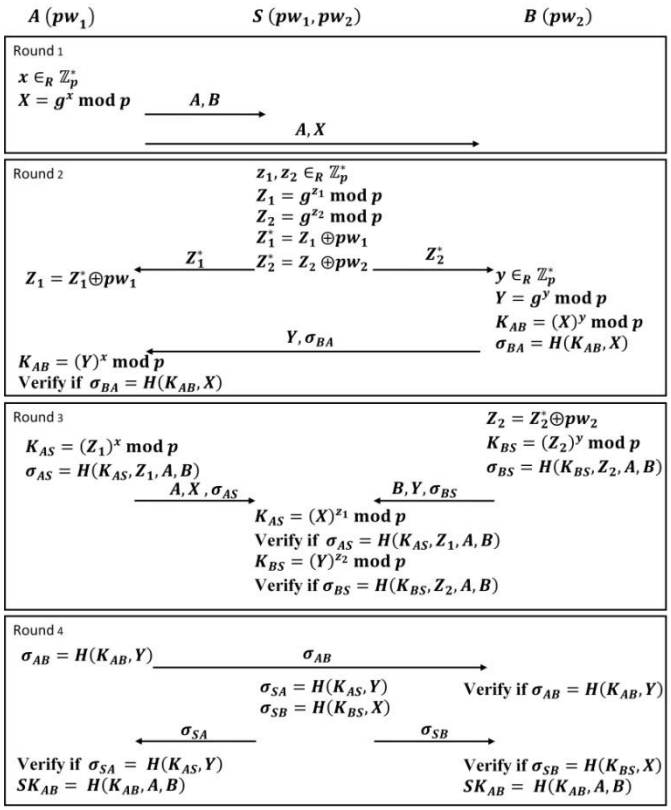

Figure 4. CHY-3PAKEv2 protocol [27]

III. CRYPTANALYSIS OF THE S-IA-3PAKE AND S-EA-3PAKE PROTOCOLS

This section shows that S-EA-3PAKE and S-IA-3PAKE protocols are vulnerable to Unknown Key Share (UKS) and password compromise impersonation attacks. S-EA-3PAKE and S-IA-3PAKE protocols are similar to a great extent, and the only difference between them is in their server authentication, which is explicit in S-EA-3PAKE and implicit in S-IA-3PAKE. As a result, the detailed descriptions of the mentioned attacks on S-EA-3PAKE protocol are described as follows. It is noteworthy that the S-IA-3PAKE protocol, which does not provide explicit server authentication, is also subject to the same kind of attacks.

(1) Vulnerability to Unknown Key Share (UKS) attack

In this section, it is shown that the S-EA-3PAKE protocol is susceptible to an Unknown Key Share (UKS) attack [7], [32-34], which is always hard to be detected. In particular, the user C, who is a valid user registered with the authentication server S and not supposedly involved in the execution of protocol, can share a session key with user B by masquerading him- or herself as A, but all the while with B thinking it is sharing a key with A, who is not sharing any session key with B or C. More precisely, the UKS attack against B, which runs in a straightforward fashion, is explained the following and illustrated in Figure 5.

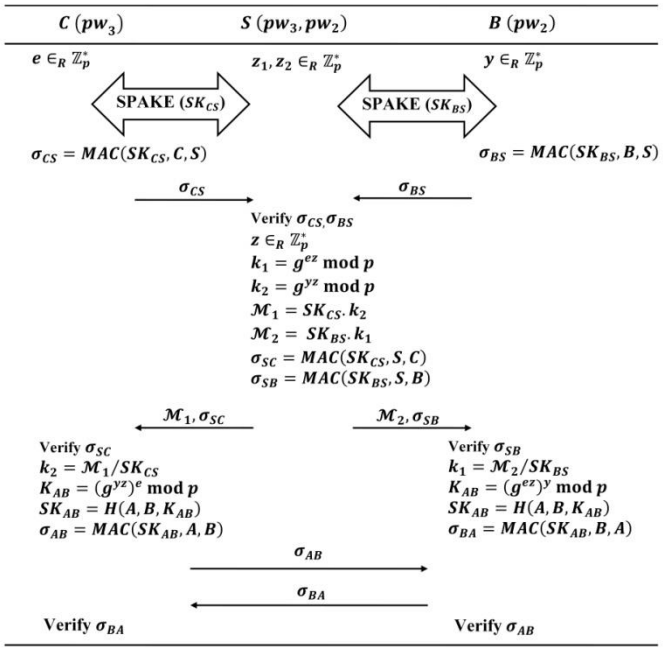

Figure 5. The Unknown Key Share (UKS) attack, against user B, on S-EA-3PAKE protocol

(1) In Step 1 of the S-EA-3PAKE protocol, the malicious user C initiates the UKS attack against B by choosing a random number $e \in_R \mathbb{Z}_p^*$, executes the SPAKE protocol with server S. Thus, C and S share the common session key $SK_{CS} = g^{ez_1} \bmod p$ as follows:

- C randomly selects a number $e \in_R \mathbb{Z}_p^*$, calculates $E = g^e \bmod p$ and $E^* = E.M^{pw_3}$, pw_3 is the password pre-shared by C with S, and sends E^* to the server S.
- Upon receiving E^*, S randomly chooses $z_1 \in_R Z_p^*$ and computes $E = E^*/M^{pw_3}$, $SK_{CS} = (E)^{z_1} = (g^e)^{z_1} \bmod p$, $Z_1 = g^{z_1} \bmod p$ and $Z_1^* = Z_1.N^{pw_3}$ and then sends Z_1^* to C.
- When C receives Z_1^*, s/he computes $Z_1 = Z_1^*/N^{pw_3}$, $SK_{CS} = (Z_1)^e = (g^{z_1})^e \bmod p$.

(2) In Step 2 of the protocol, similarly, the user B and server S randomly choose two numbers $y \in_R \mathbb{Z}^*{}_p$ and $z_2 \in_R \mathbb{Z}_p^*$, execute the SPAKE protocol and derive the common session key $SK_{BS} = g^{yz_2} \bmod p$.

(3) In Step 3 of the protocol, The users C and B compute and issue the values of authenticator $\sigma_{CS} = MAC(SK_{CS}, C, S)$ and $\sigma_{BS} = MAC(SK_{BS}, B, S)$ to S, respectively, causing S to think that users C and B want to set up a protocol session.

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.75 e-ISSN: 2251-7545

(4) In Step 4 of the protocol, when the received authenticators σ_{CS} and σ_{BS} are verified by S successfully, S chooses another random number $z \in_R \mathbb{Z}_p^*$ and computes $\sigma_{SC} = MAC(SK_{CS}, S, C)$, $\sigma_{SB} = MAC(SK_{BS}, S, B)$, $\mathcal{M}_1 = SK_{CS}.k_2$ in which $k_2 = g^{yz} \bmod p$ and $\mathcal{M}_2 = SK_{BS}.k_1$ in which $k_1 = g^{ez} \bmod p$. Then, S sends σ_{SC}, $\mathcal{M}_1$ and σ_{SB}, $\mathcal{M}_2$ to C and B, respectively .

(5) After receiving the messages from the server S, C and B verify the validity of authenticators σ_{SC} and σ_{SB}, respectively. Afterward, C and B calculate $K_{AB} = (k_2)^e = (g^{yz})^e \bmod p$ where $k_2 = \mathcal{M}_1/SK_{CS}$ and $K_{AB} = (k_1)^y = (g^{ez})^y \bmod p$ where $k_1 = \mathcal{M}_2/SK_{BS}$, respectively. Consequently, C and B derive the same value $K_{AB} = g^{eyz} \bmod p$ and establish the common session key $SK_{AB} = H(A, B, K_{AB})$.

(6) In Step 6 of the protocol, B calculates and sends his/her authenticator $\sigma_{BA} = MAC(SK_{AB}, B, A)$ to A , which is intercepted by C. At the same time, C computes and issues the forged message $\sigma_{AB} = MAC(SK_{AB}, A, B)$ to B . Finally, B successfully verifies σ_{AB} and believes that the another party is legitimate user A.

At the end of protocol, B believes that s/he has shared the session key with A, but indeed s/he has established the session key with C and C knows that s/he and B have agreed the same session key $SK_{AB} = H(A, B, K_{AB})$. Meanwhile, it is not necessary for B to be present at all.

- In a similar way, the malicious user C can apply the Unknown Key Share (UKS) attack, against user A, to S-EA-3PAKE protocol.

(2) Vulnerability to password compromise impersonation attack

In the S-EA-3PAKE protocol, if the user A's password, pw_1, is compromised, the adversary E can easily share the common session key with A and masquerade him- or herself as B successfully as follows:

(1) In Step 1 of the S-EA-3PAKE protocol, E resides at the place of server S, masquerade him- or herself as S, runs the SPAKE protocol with user A. Therefore, E and A derive the common session key $SK_{AS} = g^{xe_1} \bmod p$ as follows:

- The adversary E intercepts message X^* from A . Afterward, E disconnects the communication between A and S , retrieves X by computing $X = X^*/M^{pw_1}$, randomly selects an exponent $e_1 \in_R \mathbb{Z}_p^*$, computes the session key $SK_{AS} = (X)^{e_1} = (g^x)^{e_1} \bmod p$, $E_1 = g^{e_1} \bmod p$ and $E_1^* = E_1.N^{pw_1}$ and then sends the first forged message E_1^* to A.
- Upon receiving E_1^* , A computes $E_1 = E_1^*/N^{pw_1}$, $SK_{AS} = (E_1)^x = (g^{e_1})^x \bmod p$.

- At last, the common session key $SK_{AS} = g^{xe_1} \bmod p$ is established by E with A successfully but all the while with A thinking s/he is agreeing the session key SK_{AS} with the legal server S.

(2) In Step 3 of the protocol, A computes and issues his/her authenticator $\sigma_{AS} = MAC(SK_{AS}, A, S)$ to S , which is intercepted by E.

(3) In Step 4 of the protocol, E verifies σ_{AS}. E selects another random exponent $e_2 \in_R \mathbb{Z}_p^*$ and computes $\sigma_{SA} = MAC(SK_{AS}, S, A)$, $\mathcal{M}_1 = SK_{AS}.k_2$ in which $k_2 = g^{e_2} \bmod p$. Finally, E sends the forged messages σ_{SA} and $\mathcal{M}_1$ to A.

(4) Upon receiving the messages σ_{SA} and $\mathcal{M}_1$ from E , A verifies the validity of authenticator σ_{SA} by using SK_{AS} successfully, then A computes $K_{AB} = (k_2)^x = (g^{e_2})^x \bmod p$ where $k_2 = \mathcal{M}_1/SK_{AS}$ and generates the session key $SK_{AB} = H(A, B, K_{AB})$. Finally, A computes and issues his/her authenticator $\sigma_{AB} = MAC(SK_{AB}, A, B)$ to B, which is captured by E. At this time, E derives the same value $K_{AB} = (X)^{e_2} = (g^x)^{e_2} \bmod p$ and shares the session key $SK_{AB} = H(A, B, K_{AB})$ with A . Then, E calculates $\sigma_{BA} = MAC(SK_{AB}, B, A)$ and impersonates him- or herself as B by sending the last forged message σ_{BA} to A.

(5) When A receives σ_{BA} from E, A validates σ_{BA} successfully by using SK_{AB}. Thus, A believes that the opposite side is the benign user B, but it is in fact E who could also share the common session key $SK_{AB} = H(A, B, K_{AB})$ with A and masquerade him- or herself as the legitimate user B.
- On the other hand, if the password of B, pw_2, is revealed, the adversary E can easily establish the common session key with B and impersonate him- or herself as A successfully as follows:

(1) In Step 2 of the S-EA-3PAKE protocol, E first resides at the place of server S, executes the SPAKE protocol with user A . Then, E can share the common session key $SK_{BS} = g^{ye_1} \bmod p$ with B by masquerading him- or herself as the legitimate server S as follows:

- The adversary E captures message Y^* from B. Then, E can disconnect the communication between S and B, randomly choose an exponent $e_1 \in_R \mathbb{Z}_p^*$, retrieves Y by calculating $Y = Y^*/M^{pw_2}$, constructs the secure session key $SK_{BS} = (Y)^{e_1} = (g^y)^{e_1} \bmod p$, $E_2 = g^{e_1} \bmod p$ and $E_2^* = E_2.N^{pw_2}$, then sends the first forged message E_2^* to A.
- After receiving the message E_2^* , B calculates $E_2 = E_2^*/N^{pw_2}$, $SK_{BS} = (E_2)^y = (g^{e_1})^y \bmod p$.

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.75 e-ISSN: 2251-7545

- Finally, the adversary E could easily share the common session key $SK_{BS} = (g^{y})^{e_1} \bmod p$ with B successfully but all the while B thinks that s/he has established the secure session key $SK_{BS} = (g^{e_1})^{y} \bmod p$ with the benign server S.

(2) In Step 3, B calculates and sends his/her authenticator $\sigma_{BS} = MAC(SK_{BS}, B, S)$ to S, which is captured by E.

(3) Upon receiving the message σ_{BS} from B in Step 4, E validates σ_{BS}, chooses another random exponent $e_2 \in_R \mathbb{Z}_p^*$ and calculates $\sigma_{SB} = MAC(SK_{BS}, S, B)$, $\mathcal{M}_2 = SK_{BS}.k_1$ in which $k_{1=} g^{e_2} \bmod p$. At last, E impersonates him-or herself as the valid server S by sending the forged messages σ_{SB} and $\mathcal{M}_2$ to B.

(4) When B receives the messages σ_{SB} and $\mathcal{M}_2$ from E, B validates the authenticator σ_{SB} by using SK_{BS}successfully, then B calculates $K_{AB} = (k_1)^{y} = (g^{e_2})^{y} \bmod p$ where $k_1 = \mathcal{M}_2 / SK_{BS}$ and constructs the session key $SK_{AB} = H(A, B, K_{AB})$. Then, B computes and sends his/her authenticator $\sigma_{BA} = MAC(SK_{AB}, B, A)$ to A, which is intercepted by E. Meanwhile, E generates the same value $K_{AB} = (g^{y})^{e_2} \bmod p$ and establishes the common session key $SK_{AB} = H(A, B, K_{AB})$ with B. Finally, E computes and issues the last forged message $\sigma_{AB} = MAC(SK_{AB}, A, B)$ to B.

(5) Upon receiving the message σ_{AB} from E, B verifies σ_{AB} successfully by using SK_{AB}. Therefore, B thinks that the opposite side is the legitimate user A, but E could easily impersonate him- or herself as the benign user A and establish the common session key SK_{AB} with B successfully.

- Based on the upper steps, it is shown that if the passwords of A or B is disclosed, the adversary E can simply share a session key with corresponding user, and impersonate him- or herself instead of opposite party. Therefore, the S-EA-3PAKE protocol suffers from password compromise impersonation attack.

IV. CRYPTANALYSIS OF THE CHY-3PAKE PROTOCOLS

In this section, it is shown that CHY-3PAKEv1 and CHY-3PAKEv2 protocols are subject to password compromise impersonation and ephemeral key compromise impersonation attacks. Because the CHY-3PAKEv2 is the parallel version of CHY-3PAKEv1 and the CHY-3PAKEv2's contents of the transmitted messages are also similar to those in CHY-3PAKEv1 so the details of the password compromise impersonation and ephemeral key compromise impersonation attacks on CHY-3PAKEv1 protocol are explained as follows. Similarly, the following attacks are also applicable to CHY-3PAKEv2 protocol.

(1) Vulnerability to password compromise impersonation attack

In the CHY-3PAKEv1 protocol, if the adversary E compromises the user A's password pw_1, E can easily establish the same session key with A and impersonate him- or herself as B successfully. More precisely, the attack works as follows:

(1) E mediates between A and S, initiates the attack against A by intercepting the messages Z_1^* and Z_2^* from S in Step 2, disconnecting the communication between A and S. Then, s/he impersonates him- or herself as S, randomly chooses a number $e_1 \in_R \mathbb{Z}_p^*$, calculates $E_1 = g^{e_1} \bmod p$ and $E_1^* = E_1 \oplus pw_1$ and sends the first forged message, which consists of E_1^* and Z_2^*, to A.

(2) After receiving the message from E, the user A obtains E_1 by computing $E_1 = E_1^* \oplus pw_1$, then randomly chooses a number $x \in_R \mathbb{Z}_p^*$, computes $X = g^{x} \bmod p$, $K_{AS} = (E_1)^{x} = (g^{e_1})^{x} \bmod p$ and $\sigma_{AS} = H(SK_{AS}, E_1, A, B)$ and then issues A, X, σ_{AS} and Z_2^* to B, which are intercepted by E.

(3) E first masquerades him- or herself as B, randomly selects another number $e_2 \in_R \mathbb{Z}_p^*$, computes $E_2 = g^{e_2} \bmod p$, $K_{AB} = (X)^{e_2} = (g^{x})^{e_2} \bmod p$ and $\sigma_{BA} = H(K_{AB}, X)$. S/He also directly skips to Step 5 of the protocol for impersonating him- or herself as the legitimate server S. Then, s/he constructs $K_{AS} = (X)^{e_1} = (g^{x})^{e_1} \bmod p$ and $\sigma_{SA} = H(K_{AS}, E_2)$ and values the authenticator σ_{SB} with a random value (e.g.$\sigma_{SB} = H(E_2, X)$) because the user A does not apply any verification on σ_{SB} and only forwards it to B. Finally, E sends the forged message, which includes $Y, \sigma_{SA}, \sigma_{BA}$ and σ_{SB}, to A.

(4) Upon receiving E's message, A first verifies $\sigma_{SA} = H(K_{AS}, E_2)$ successfully using $K_{AS} = (g^{e_1})^{x} \bmod p$. Then, s/he computes $K_{AB} = (E_2)^{x} = (g^{e_2})^{x} \bmod p$ and confirms that the received σ_{BA} is valid using K_{AB}. Therefore, A calculates $\sigma_{AB} = H(K_{AB}, Y)$, generates the session key as $SK_{AB} = H(K_{AB}, A, B)$ in which $K_{AB} = (g^{e_2})^{x} \bmod p$ and believes that B also has the ability to derive the same session key. Finally, A issues σ_{SB} and σ_{AB} to B, which is captured by E.

(5) E verifies σ_{AB} using K_{AB} and calculates the session key as $SK_{AB} = H(K_{AB}, A, B)$ in which $K_{AB} = (g^{x})^{e_2} \bmod p$. Therefore, E has established the common session key with A and impersonated him or herself as B successfully, while A erroneously thinks that s/he has shared a secure session key with B.

- On the other hand, it is assumed that the adversary E compromises pw_2 of the user B. Then E will be able to impersonate him- or herself as A and agree a common session key with B. In more detail, the attack is successfully applied to the CHY-3PAKEv1 protocol as follows:

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.75 e-ISSN: 2251-7545

(1) In Step 3 of the protocol, E poses him- or herself as A , selects two random numbers $e_1 \in_R \mathbb{Z}_p^*$ and $e_2 \in_R \mathbb{Z}_p^*$, calculates $E_1 = g^{e_1} \bmod p$, $E_2 = g^{e_2} \bmod p$, and $E_2^* = E_2 \oplus pw_2$ using pw_2 , which is compromised by E, and values the parameter σ_{AS} with a random value (e.g. $\sigma_{AS} = H(g^{e_1 e_2} \bmod p, E_1, A, B)$) since the user B can not verify the correctness of σ_{AS} and can only forward it to A. Then, E sends the first forged message, which includes A, E_1, σ_{AS} and E_2^*, to B.

(2) Upon receiving the message from E, B randomly chooses a number $y \in_R \mathbb{Z}_p^*$ and obtains E_2 by computing $E_2 = E_2^* \oplus pw_2$, calculates $Y = g^y \bmod p$, $K_{BS} = (\mathrm{E}_2)^y = (\mathrm{g}^{\mathrm{e}_2})^y \bmod p$ and $K_{AB} = (\mathrm{E}_1)^y = (g^{\mathrm{e}_1})^y \bmod p$, $\sigma_{BS} = H(K_{BS}, E_2, A, B)$ and $\sigma_{BA} = H(K_{AB}, \mathrm{E}_1)$. Finally, B issues $E_1, \sigma_{AS}, Y, \sigma_{BS}$ and σ_{BA} to S , which is intercepted by E.

(3) E first impersonates him- or herself as S, calculates $K_{BS} = (Y)^{e_2} = (\mathrm{g}^{\mathrm{y}})^{\mathrm{e}_2} \bmod p$ and , then verifies if $\sigma_{BS} = H(K_{BS}, E_2, A, B)$ or not, . If the equality is satisfied, E computes the forged authenticator $\sigma_{SB} = H(K_{SB}, E_1)$, then skips to Step 6, masquerades him- or herself as A, calculates $K_{AB} = (Y)^{e_1} = (g^y)^{e_1} \bmod p$ and checks if $\sigma_{BA} = H(K_{AB}, E_1)$ or not. If the result is positive, s/he calculates the forged $\sigma_{AB} = H(K_{AB}, Y)$, generate the session key as $SK_{AB} = H(K_{AB}, A, B)$ and sends the last false message, which consists of σ_{SB} and σ_{AB} , to B .

(4) After receiving E's message, B validates $\sigma_{SB} = H(K_{BS}, E_1)$ and $\sigma_{AB} = H(K_{AB}, Y)$ successfully so s/he generates the session key as $SK_{AB} = H(K_{AB}, A, B)$ in which $K_{AB} = (E_1)^y = (g^{\mathrm{e}_1})^y \bmod p$ and also believes that the opposite party is the valid user A, has obtained the common session key SK_{AB}.

(5) At the end of the protocol, E succeeds by masquerading him- or herself as A as well as establishing the common session key $SK_{AB} = H(K_{AB}, A, B)$ with B.

- Consequently, CHY-3PAKEv1 protocol does not provide resilience to password compromise impersonation attack, which is an imperative security attribute that any proposed password authenticated key exchange (PAKE) protocol is required to hold.

(2) Vulnerability to ephemeral key compromise impersonation attack

Indeed, two random numbers $x \in_R \mathbb{Z}_p^*$ and $y \in_R \mathbb{Z}_p^*$ are the ephemeral keys of the user A and B, respectively, in every session. If an adversary E accesses any of the ephemeral keys, s/he will be able to compute the common session key $SK_{AB} = H(K_{AB}, A, B)$ and completes the authentication. Therefore, CHY-3PAKEv1 protocol is insecure against ephemeral key compromise impersonation attack. The following descriptions show how the attack works in detail.

- If E obtains the user A's ephemeral key x, s/he can set up the protocol and establish the common session key SK_{AB} with A as follows:

(1) in Step 4 of the protocol, E first intercepts B's message, which includes $X, \sigma_{AS}, Y, \sigma_{BS}$ and σ_{BA}. Then, s/he disconnects the communication between B and S, calculates $K_{AB} = (Y)^x = (g^y)^x \bmod p$ using A's ephemeral key x, which is revealed by E , and generates the session key $SK_{AB} = H(K_{AB}, A, B)$ successfully. Finally, E impersonates him- or herself as B by forwarding $X, \sigma_{AS}, Y, \sigma_{BS}$ and σ_{BA} to S.

(2) After receiving the message from E, S computes $K_{AS} = (X)^{z_1} = (g^x)^{z_1} \bmod p$ and $K_{BS} = (g^y)^{z_2} \bmod p$, validates the authenticators $\sigma_{AS} = H(K_{AS}, Z_1, A, B)$ and $\sigma_{BS} = H(K_{BS}, Z_2, A, B)$, calculates his/her authenticators $\sigma_{SA} = H(K_{AS}, Y)$ and $\sigma_{SB} = H(K_{BS}, X)$ and issues $Y, \sigma_{SA}, \sigma_{BA}$ and σ_{SB} to A.

(3) A computes $K_{AB} = (Y)^x = (g^y)^x \bmod p$, then verifies the validity of $\sigma_{SA} = H(K_{AS}, Y)$ and $\sigma_{BA} = H(K_{AB}, X)$ successfully so s/he calculates $\sigma_{AB} = H(K_{AB}, Y)$, generates the session key as $SK_{AB} = H(K_{AB}, A, B)$. Finally, A issues σ_{SB} and σ_{AB} to B, which is captured by E.

(4) E verifies the correctness of the received authenticator $\sigma_{AB} = H(K_{AB}, Y)$ using SK_{AB} successfully. Consequently, A believes that the opposite side is the valid user B and E is assured that s/he shares the common session key SK_{AB} with A and impersonates him- or herself as the benign user B.

- On the other hand, if the user B's ephemeral key y, is divulged, the adversary E proceeds the protocol steps and shares the common session key $SK_{AB} = H(K_{AB}, A, B)$ with B as follows:

(1) E initiates the attack against B by intercepting A 's message, which consists of A, X, σ_{as} and Z_2^*, in Step 3 of the protocol. Afterward, s/he disconnects the communication between A and B, computes $K_{AB} = (X)^y = (g^x)^y \bmod p$ using B's ephemeral key y, which is compromised by E, constructs the session key as $SK_{AB} = H(K_{AB}, A, B)$ successfully and simultaneously poses him- or herself as A by forwarding A's message unaltered to B.

(2) After receiving the message from E, B chooses a random number $y \in_R \mathbb{Z}_p^*$ and obtained Z_2 as $Z_2 = Z_2^* \oplus pw_2$, computes $Y = g^y \bmod p$, $K_{BS} = (Z_2)^y = (g^{z_2})^y \bmod p$ and $K_{AB} = (X)^y = (g^x)^y \bmod p$, $\sigma_{BS} = (K_{BS}, Z_2, A, B)$ and $\sigma_{BA} = H(K_{AB}, X)$. Finally, B sends $X, \sigma_{AS}, Y, \sigma_{BS}$ and σ_{BA} to S.

(3) In Step 5 of the protocol, S first calculates $K_{AS} = (X)^{z_1} = (g^x)^{z_1} \bmod p$ and $K_{BS} = (g^y)^{z_2} \bmod p$, then validates $\sigma_{AS} = H(K_{AS}, Z_1, A, B)$ and $\sigma_{BS} = H(K_{BS}, Z_2, A, B)$ successfully,

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.75 e-ISSN: 2251-7545

calculates his/her authenticators $\sigma_{SA} = H(K_{AS}, Y)$ and $\sigma_{SB} = H(K_{BS}, X)$ and issues $Y, \sigma_{SA}, \sigma_{BA}$ and σ_{SB} to A, which are intercepted by E.

(4) E first checks the correctness of its computed session key SK_{AB} by verifying received authenticator $\sigma_{BA} = H(K_{AB}, X)$, then calculates its forged authenticator$\sigma_{AB} = H(K_{AB}, Y)$ using SK_{AB}. At last, E masquerades him- or herself as A by sending the forged message, which includes σ_{SB} and σ_{AB}, to B.

(5) Upon receiving E's message, B verifies the authenticators $\sigma_{SB} = H(K_{BS}, E_1)$ and $\sigma_{AB} = H(K_{AB}, Y)$ successfully. Thus, s/he computes the session key as $SK_{AB} = H(K_{AB}, A, B)$ in which $K_{AB} = (X)^y = (g^x)^y \bmod p$ and also believes that the opposite party is the benign user A, has derived the same session key SK_{AB}.

(6) Finally, E succeeds with masquerading him- or herself as A as well as establishing the common session key $SK_{AB} = H(K_{AB}, A, B)$ with B.

- From the above steps, the authentication infrastructure of the protocol is completely failed and the adversary E can simply mount ephemeral key compromise impersonation attack on CHY-3PAKEv1 protocol.

V. Conclusion

In this paper, the security vulnerabilities of S-IA-3PAKE, S-EA-3PAKE, CHY-3PAKEv1 and CHY-3PAKEv2 protocols [25], [27] are analyzed and it is proved that both provably secure S-IA-3PAKE and S-EA-3PAKE protocols, which are based on the SPAKE protocol developed by Abdalla and Pointcheval [26], are subject to Unknown Key Share (UKS) and password compromise impersonation attacks. It is also shown that the so-called secure CHY-3PAKEv1 and CHY-3PAKEv2 protocols, which are based on LHL-3PAKE proposed by Lee et al. [4], are vulnerable to password compromise impersonation and ephemeral key compromise impersonation attacks. Our results show that the proof of security for a protocol is a baffling task to analyze and should be managed carefully utilizing the provable security model by the designers.

References

[1] S. Bellovin and M. Merritt, "Encrypted key exchange: Password based protocols secure against dictionary attacks," in: Proceedings of the 1992 IEEE Symposium on Security and Privacy, pages 72–84, 1992.

[2] M. Abdalla, P.-A. Fouque, and D. Pointcheval, "Password-Based Authenticated Key Exchange in the Three-Party Setting," Proc. PKC '05, LNCS 3386, pp. 65-84, 2005.

[3] M. Abdalla and D. Pointcheval, "Interactive Diffie-Hellman Assumptions with Applications to Password-Based Authentication," Proc. FC '05, LNCS 3570, pp. 341-356, 2005.

[4] T-F. Lee, T. Hwang, and C-L. Lin, "A novel three-party encrypted key exchange protocol," Computers and Security Vol 23, No. 7, pp.571-577, 2004.

[5] M. Abdalla, P.-A. Fouque, D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," IEE Proceedings – Information Security 153 (1) (2006) 27–39.

[6] R. Lu, Z. u Cao, "Simple three-party key exchange protocol," Computers and Security 26 (1) (2007) 94–97.

[7] R.C.-W. Phan, W.-C. Yau, B.-M. Goi, "Cryptanalysis of simple three-party key exchange protocol (S-3PAKE)," Information Sciences 178 (13) (2008) 2849– 2856.

[8] H.-R. Chung, W.-C. Ku, "Three weaknesses in a simple three-party key exchange protocol," Information Sciences 178 (2008) 220–229.

[9] Y. Ding, P. Horster, "Undetectable on-line password guessing attacks," ACM Operating Systems Review 29 (4) (1995) 77–86.

[10] C.C. Chang, Y.F. Chang, "A novel three-party encrypted key exchange protocol," Computer Standards & Interfaces 26 (5) (2004) 471–476.

[11] M. Bellare, P. Rogaway, "Entity authentication and key distribution," in: Advances in Cryptology – Crypto'93, LNCS, vol. 773, 1993, pp. 232–249.

[12] Young M. Bellare, P. Rogaway, "Provably secure session key distribution: the three party case," in: Proceedings of the ACM Symposium on the Theory of Computing (STOC'95), 1995, pp. 57–66.

[13] B. Kaliski, "An unknown key-share attack on the mqv key agreement protocol," ACM Transactions on Information and System Security (TISSEC), 4(3):275 – 288, 2001.

[14] Young 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, 177(6):13641375, 2007.

[15] P. Nose, "Security weaknesses of authenticated key agreement protocols," Information Processing Letters, 111:687696, 2011.

[16] R.-W. Phan and B.-M. Goi, "Cryptanalysis of two provably secure cross-realm c2c-pake protocols," in: Proceedings of the Indocrypt06 LNCS, 4329:104117, 2006.

[17] M. Saeed and H. Shahhoseini, "Security Analysis and Improvement of Smart Card-based Authenticated Key Exchange Protocol with CAPTCHAs for Wireless Mobile Network," Proceedings of the 2011 IEEE 16th Symposium on Computers and Communications (ISCC2011), July, 2011.

[18] T. Clancy, "Eap password authenticated exchange," Draft archive.http://www.cs.umd.edu/ clancy/eap-pax/, 2005.

[19] C. J. Cremers, "Session-state reveal is stronger than ephemeral key reveal: Attacking the naxos authenticated key exchange protocol," In: http://eprint.iacr.org/2008/376

[20] M. Gouda, A. Liu, L. Leung, and M. Alam, "Spp: An anti-phishing single password protocol," Computer Networks, 51(13):3715–3726, 2007.

[21] S. Halevi and H. Krawczyk, "Public-key cryptography and password protocols," ACM Transactions on Information and System Security (TISSEC), 2(3):230 – 268, 1999.

[22] I.-R. Jeong, J.-O. Kwon, and D.-H. Lee, "Strong Di_e-Hellman- DSA Key Exchange," IEEE Communications Letters, 11(5):432 – 433, May, 2007.

[23] J. Katz, R. Ostrovsky, and M. Yung, "Efficient password authenticated key exchange using human-memorable passwords," in: Proc. Eurocrypt01, LNCS, 2045:475–494, 2001.

[24] M. Saeed, H. Shahhoseini, and A. Mackvandi, "An Improved two-party Password Authenticated Key Exchange Protocol without Server's Public

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],

San Francisco State University, CA, U.S.A., March 2013

Doi: 10.7321/jscse.v3.n3.75 e-ISSN: 2251-7545

Key," In: Proceedings of the 2011 IEEE 3th International Conference on Information and Computer Networks (ICICN 2011), pages :90–95, January, 2011.

[25] T-F. Lee and T. Hwang, "Simple Password-based three-party Autheticated Key Exchage without Server Public Keys," Information Sciences 180 (9) (2010) 1702– 1714.

[26] M. Abdalla, D. Pointcheval, "Simple password-based authenticated key protocols," in: Topics in Cryptology – CT-RSA 2005, Lecture Notes in Computer Science, vol. 3376, 2005, pp. 191–208.

[27] T-Y. Chang, M-S. Hwang, W-P. Yang, "A Communication-efficient three-party Password Authenticated Key Exchange Protocol," Information Sciences 181 (1) (2011) 217– 226.

[28] K.-K. Choo, C. Boyd, and Y. Hitchcock, "Errors in computational complexity proofs for protocols," in: Advances in Cryptology Asiacrypt05, LNCS, 3788:624–643, 2005.

[29] K.-K. Choo, C. Boyd, and Y. Hitchcock, "Examining indistinguishability-based proof models for key establishment protocols," in: Advances in Cryptology Asiacrypt05, LNCS, 3788:585604, 2005.

[30] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in: Proceedings of the 19th international conference on Theory and application of cryptographic techniques, LNCS, 1807:139–155, 2000.

[31] R.-W. Phan and B.-M. Goi, "Cryptanalysis of the n-party encrypted diffiehellman key exchange using different passwords," in: Proceedings of the ACNS06, LNCS, 3989:226238, 2006.

[32] K.-K.R. Choo, C. Boyd, Y. Hitchcock, "Examining indistinguishability-based proof models for key establishment protocols," in: Advances in Cryptology – Asiacrypt'05, LNCS, vol. 3788, 2005, pp. 585–604.

[33] W. Diffie, P.C. van Oorschot, M.J. Wiener, "Authentication and authenticated key exchanges," Design, Codes and Cryptography 2 (2) (1992) 107–125.

[34] B.S. Kaliski Jr, "An unknown key-share attack on the MQV key agreement protocol," ACM Transactions on Information and System Security (TISSEC) 4 (3) (2001) 275–288.