An enhanced of Simple Three-Party Key Exchange Protocol

Fuw-Yi Yang Wen-Hui Li Cai-Ming Liao
Department of Computer Science and Information Engineering,
Chaoyang University of Technology
yangfy@cyut.edu.tw s9627641@cyut.edu.tw s9527645@cyut.edu.tw

Abstract

Three-party password authenticated key exchange (3PAKE) protocol is capable of reducing the number of required keys stored by users in a conference. In 2007, Lu and Cao proposed a simple three-party password authenticated key exchange (S-3PAKE) protocol for assisting users to complete business negotiations and agreements during a communication process without requiring a public server key, and the S-3PAKE protocol can resist all known attacks. However, in 2008, Cuo et al. pointed out the protocol proposed by Lu et al. still cannot resist man-in-the-middle attacks and online dictionary attacks thus further proposed an improved solution as follows. Two parties execute the 2-PAKE protocol to obtain an agreement key first, so that when they execute the 3PAKE protocol, they can authenticate the real identity with each other. An effective three-party password authenticated key exchange (3PAKE) protocol was proposed for authenticating the real identity of both parties without requiring an execution of the 2-PAKE protocol to obtain the agreement key first, but simply adding the real identity of the opposite party in the communication process, so as to achieve the same security effect.

Keywords: Three-party key exchange, Password-authenticated key exchange, Man-in-the-middle attack, Dictionary attack.

1. Introduction

In recent years, network technology becomes well developed and popular, and transactions such as network trading and ATM transfer are used extensively thus finding a way to achieve a safe communication and a secured authentication of user's real identity becomes an important issue in public network applications.

In 1992, Bellovin and Merrit [1] proposed a password authenticated key exchange protocol, such that users can share a session key through a safe communication mechanism. If a user wants to create a session key to be shared with several other users, the user has to remember several passwords, but those passwords is composed of numerals that can be guessed easily, and users are exposed to the risk of being attacked by password guessing attacks

during the communication with others. Therefore, scholars and experts in the related field attempted to propose improvements and solutions to fix the security loophole by extending the 2PAKE protocol to the 3PAKE protocol. In 1995, Steiner et al. [2] proposed a 3PAKE protocol, such that if two users want to share a session key, each user must share a set of passwords issued by an eligible organization whose server stores the user passwords, and provides the passwords for another user to authenticate their identities. Although this protocol allows users to authenticate the identities with each other, yet the protocol still faces a high risk of being attacked by the password guessing attacks. Therefore, Ding and Horster [3] in the same year pointed out that the protocol proposed by Steiner et al. still cannot resist the online password guessing attacks. Invaders can intercept an authorized user's password and guess the password through a publicly transmitted message. In 2000, Lin et al. [4] analyzed the protocol proposed by Steiner et al. and concluded that such protocol is unable to resist online password guessing attacks, and thus proposed an improved solution. Lin et al. encrypt messages transmitted through a server key, and users must verify the message. However, users must confirm their identity with a server in advance when they use the server key mechanism, and thus such arrangement will incur a higher consumption cost to users. To make further improvements, Lin et al. [5] proposed an improved solution in 2001, and users simply need to hold a personal password for participating in the protocol negotiation without requiring a server key, so as to reduce the user's cost. In complicated calculations or transmissions with a high frequency, scholars and experts keep proposing different solutions [6-8].

In 2005, Abdalla et al. [9] proposed a simple password-authenticated key exchange protocol, and the security of this protocol is built according to the chosen-basis computational Diffie-Hellman (CCDH) assumption. If an invader attempts to calculate a user's randomly selected numbers, the invader will encounter the difficult discrete logarithm problem.

In 2007, Lu and Cao [10] proposed a simple protocol by integrating the protocol proposed by Abdalla et al. with the 3PAKE protocol, such that if two users want to communication with each other, the users must negotiate for a common session key, and the users can verify the opposite party's real identity without the need of providing a sever key to users for identity verification, as to prevent known attacks.

However, Guo et al. [11] pointed out in 2008 that the protocol proposed by Lu and Cao still has loopholes and attacked by man-in-the-middle attacks and undetectable online dictionary attacks. If an invader is a eligibly registed user, the invader can share passwords from a server and create a common session key with other eligible users, becuase no confirmation on the real messages returned from a server has been made. On the other hand, if an invader existts between two communicating parties, the invader can intercept messages to find

out the passwords of eligible users from the undetectable online dictionary attacks. Therefore, Guo et al. further proposed an improved protocol, such that users must execute a 2-PAKE protocol with a sever to generate a message authentication code (*MAC*) in advance before the server can authenticate the real identity of both parties. In addition, the IDs of both parties are added to execute a Hash function for setting the identity to prevent man-in-the-middle attacks and online dictionary attacks. In view of the foregoing issues, we propose an improved solution based on the protocol provided by Guo et al. without executing the 2-PAKE protocol first. In the origianl S-3PAKE protocol, the opposite party's identity is attached to the two users' messages, and the Hash function is computed. In the authentication process, it is not necessary to involve the computation of personal passwords and thus will not have any influence to the outcome, and also can prevent the man-in-the-middle attacks and the online password guessing attacks to improve the protocol provided by Guo et al. having too many computations, and achieve the same level of security. Our protocol also fits applications in a public network environment.

In this paper, reviews Lu and Cao protocol in Section 2. Next, we analyze the S-3PAKE protocol security hole and Guo et al. improvement protocol in Section 3. And then we propose an improvement protocol and security analysis in Section 4 and Section 5, respectively. Finally, conclusions are given in Section 6.

2. Review of the S-3PAKE protocol

We provide some definitions and notations and then describe the S-3PAKE protocol.

- (G, g, p): A finite cyclic group G generated by an element g of prime order p.
- M, N: Two elements in G.
- S: A trusted server.
- A, B: Two clients.
- pw_1: The password shared between A and S.
- pw_2: The password shared between B and S.
- H_1, H_2, H_3: Three secure one-way hash functions.

Under a condition of which without authorized message sharing, two users are required to negotiate an authenticated key exchange protocol concerned as an important cryptographic technique. The authenticated key exchange protocol may through login the trusted third party to process the procedure. The communication processes expressed as following and shown in Fig. 1.

(1) A selects a random number $x \in Z_p^*$ and computes $X = g^x \cdot M^{pw_1} \in G$, for convenience writing will calculate in the future leaves out of G, and then sends (A, X) to B.

(2) B selects a random number $y \in Z_p^*$ and computes $Y = g^y \cdot N^{pw_2}$, and then

sends (A, X, B, Y) to *S*.

(3) Upon receiving (A, X, B, Y), the server *S* first uses the passwords pw_1 and pw_2 to compute $g^x = X / M^{pw_1}$ and $g^y = Y / N^{pw_2}$, respectively. Then, the server *S* selects random number $z \in Z_p^*$ and computes $g^{xz} = (g^x)^z$、$g^{yz} = (g^y)^z$、$X' = (g^y)^z \cdot H_1(A, S, g^x)^{pw_1}$ and $Y' = (g^x)^z \cdot H_1(B, S, g^y)^{pw_2}$, and then sends (X', Y') to *B*.

(4) Upon receiving (A, X), *B* computes $g^{xz} = Y' / H_1(B, S, g^y)^{pw_2}$ with the password pw_2 and computes $\alpha = H_1(A, B, g^{xyz})$, and then sends (X', α) to *A*.

(5) Upon receiving (X', α), *A* computes $g^{yz} = X' / H_1(A, S, g^x)^{pw_1}$ and checks whether $H_1(A, B, g^{xyz}) = \alpha$ holds or not. If it does not hold, *A* stops executing the protocol. Otherwise, *A* believes that client *B* is valid and computes the session key $SK_A = H_2(A, B, g^{xyz})$; Then, *A* computes $\beta = H_1(B, A, g^{xyz})$, and then sends β to *B*.

(6) Upon receiving β, *B* checks whether $H_1(B, A, g^{xyz})$ hold or not. If it does hold, *B* believes that client *A* is valid and computes the session key $SK_B = H_2(A, B, g^{xyz})$.

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

3. Cryptanalysis of S-3PAKE and improved protocol

A simple three-party password based key exchange protocol proposed by Lu and Cao without server public key provides to verify users, of which no need to assist the two-party to certificated reality and able to resist the known attacks. However, Guo et al. argues that the protocol proposed has a security hole because S-3PAKE consists in man-in-the-middle attack and implicit dictionary attack. In Fig. 2, listed attacks foucs on the vulnerability on protocol.

3.1 Man-in-the-middle attack

If attacker *C* is a legal user, it means that user *C* is sharing the protocol pw_3 while at certified server *S*. Then, the attacker *C* can process the following attack whilst users *A* and *B* in communication an eavesdropper can detect who the receiver is, and makes a prospective attack target.

(1) *A*, *B* operate as specified in the protocol in the first two steps.

(2) When *B* sends the message (A, X, B, Y), *C* chooses random numbers $e \in Z_p^*$, $f \in Z_p^*$ and $Q \in G$ and computes $E = g^e \cdot Q^{pw_3}$ and $F = g^f \cdot Q^{pw_3}$. Then, adversary *C* intercepts the incoming message and forges two separate messages $msg_{m_1} = (A, X, C, E)$ and $msg_{m_2} = (C, F, B, Y)$. *C* then sends to

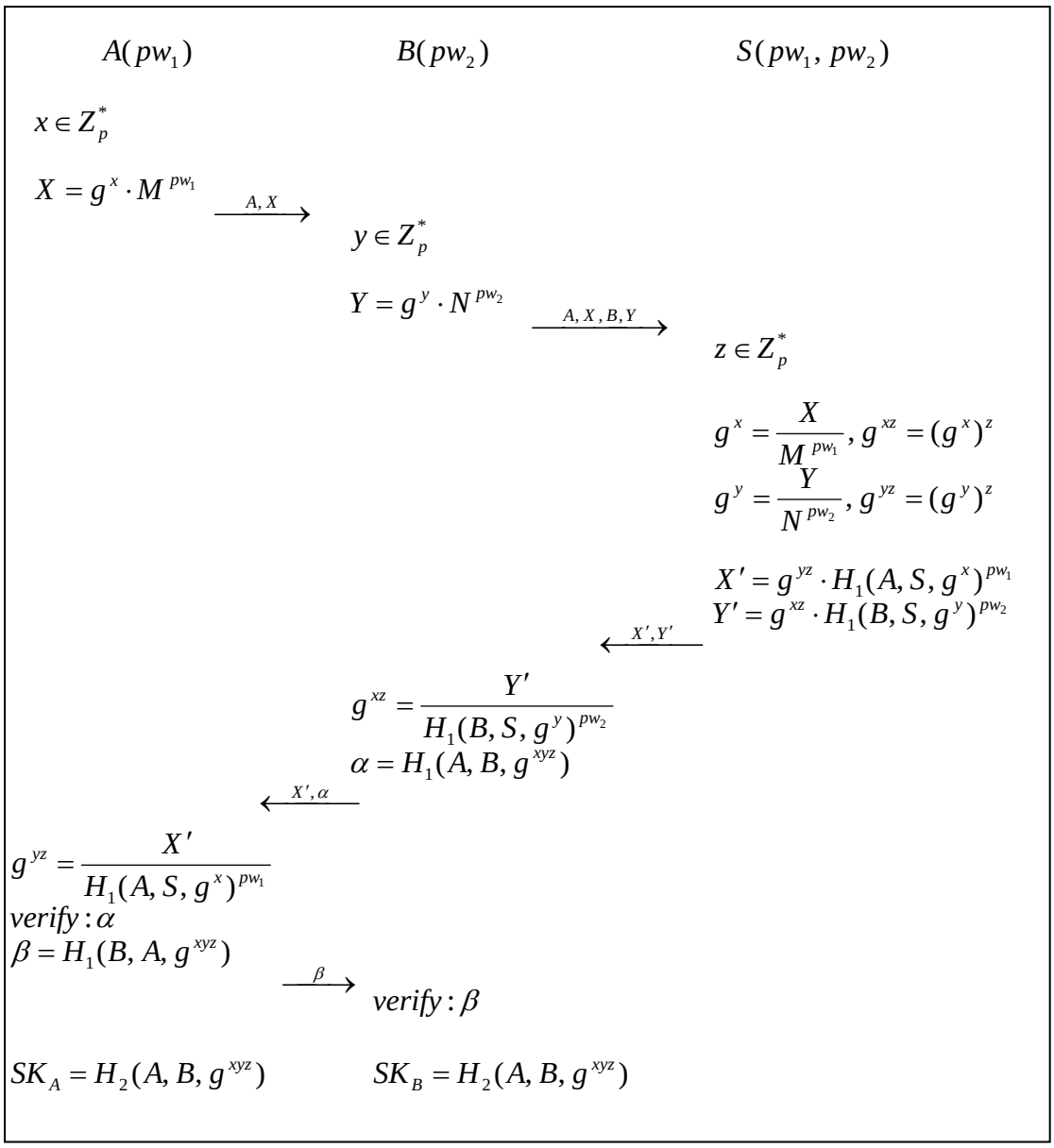

Fig. 1 S-3PAKE protocol

S the first forged message msg_{m_1} alleging that it is for establishing a session key between *A* and *C*. C then sends to *S* the second forged message msg_{m_2} alleging that it is for establishing a session key between *B* and *C*.

(3) *S* uses the passwords M^{pw_1}, N^{pw_2}, and Q^{pw_3} to compute X', Y', E', F', respectively. *S* believes that the forged messages are valid.

(4) Since msg_{m_1} and msg_{m_2} are both valid, in response to msg_{m_1}, *S* computes $g^x = X / M^{pw_1}$、$g^{xz} = (g^x)^z$、$g^e = E / Q^{pw_3}$、$g^{ez} = (g^e)^z$、$X' = g^{ez} \cdot H_1(A, S, g^x)^{pw_1}$、$E' = g^{xz} \cdot H_1(C, S, g^e)^{pw_3}$, and sends forged message $msg_{S_1} = (X', E')$ to *C*; then, in response to msg_{m_2}, *S*

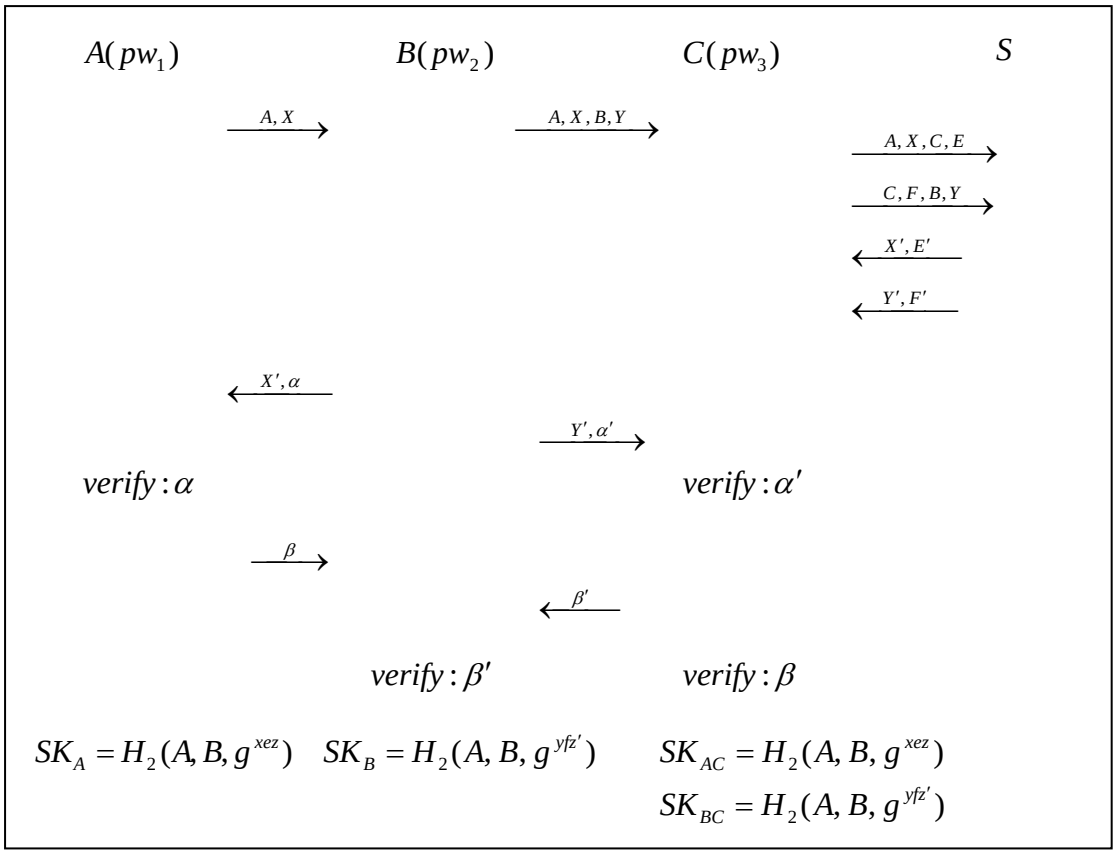

Fig. 2 The man-in-the-middle attack on S-3PAKE protocol

computes $g^x = X / M^{pw_1}$ 、 $g^{xz} = (g^x)^z$ 、 $g^e = E / Q^{pw_3}$ 、 $g^{ez} = (g^e)^z$ 、 $X' = g^{ez} \cdot H_1(A, S, g^x)^{pw_1}$ 、 $E' = g^{xz} \cdot H_1(C, S, g^e)^{pw_3}$, and sends forged message $msg_{S_1} = (X', E')$ to *C*.; then, in response to msg_{m_2} , *S* computes $g^y = Y / N^{pw_2}$ 、 $g^{yz'} = (g^y)^{z'}$ 、 $g^f = F / Q^{pw_3}$ 、 $g^{fz'} = (g^f)^{z'}$ 、 $Y' = g^{fz'} \cdot H_1(B, S, g^y)^{pw_2}$ 、 $F' = g^{yz'} \cdot H_1(C, S, g^f)^{pw_3}$, and sends forged message $msg_{S_2} = (Y', F')$ to *B*.

(5) After receiving message msg_{m_1} , *C* computes $g^{xz} = E' / H_1(C, S, g^e)^{pw_3}$ 、 $\alpha = H_1(A, B, g^{xez})$, and sends $msg_{C_1} = (X', \alpha)$ to *A*; similarly, *B* receives message msg_{m_2} , computes $g^{fz'} = Y' / H_1(B, S, g^y)^{pw_2}$ 、 $\alpha' = H_1(A, B, g^{yfz'})$, and sends $msg_{B_1} = (Y', \alpha')$ to *C*.

(6) *A* uses the password pw_1 to compute $g^{ez} = X' / H_1(A, S, g^x)^{pw_1}$ and checks whether $H_1(A, B, g^{xez}) = \alpha$ holds or not. If it does hold, *A* computes $\beta = H_1(B, A, g^{xez})$ and sends it to *B*. But, this message is intercepted by *C*. *C* computes the session key $SK_{AC} = H_2(A, B, g^{xez})$. Finally, *A* computes the session key $SK_A = H_2(A, B, g^{xez})$.

(7) Meanwhile, upon receiving msg_{B_1} from *B*, *C* computes $F' = H_1(C, S, g^e)^{pw_3}$ and checks $H_1(A, B, g^{yfz'}) = \alpha'$, and

sends $\beta' = H_1(B, A, g^{yfz'})$ to *B* as if it originated from *A*. Then, *C* computes the session key $SK_{BC} = H_2(A, B, g^{yfz'})$. When *B* receives β' from *C*, can verify it successfully using g^{yfz}. Finally, *B* will compute the session key $SK_B = H_2(A, B, g^{yfz'})$.

In this protocol, attacker *C* can read any transmission message and makes an attack and user *A* believe is sharing the session key with user *B*. Instead, the user *A* is sharing the session key with attacker *C*. In fact, the user *B* is sharing the session key with *C*. As a result, Lu and Cao's protocol cannot accomplish the mutual authentication.

3.2 Guo et al. improved protocol

Guo et al. indicates that the cryptosystem of Lu and Cao is unsafe because it is easy caused man-in-the-middle attack and dictionary attack. Assum the attacker is a legal registreated user who can shares message encryption with server and establish a sharing session key between each other. The reason is caused by the server's feedback message cannot confirm the reality of message. Beside, if attacker subsists between two communication sides, then attacker can intercept message through undetectable dictionary attack to find user's password.

Hence, to solve the protocol security problem that Guo et al. proposes a improve program by which two-side users individually implement 2-PAKE protocol to obtain Message Authentication Codes (*MAC*) prior to creating a sharing session key. In which, *S* can identify the reality of *X* and *Y* and implements the Hash function possessed $X' = g^{yz} \cdot H(A, B, S, g^x)^{pw_1}$ and $Y' = g^{xz} \cdot H(B, A, S, g^y)^{pw_2}$. If X' is not correct data, *A* may through α value to identify error message and discontinue protocol accessing by which improvement can efficiently prevents man-in-the-middle attack and indetectable dictionary attack. Fig. 3 is Guo et al. proposed protocol.

4. Our protocol

As to the protocol proposed by Guo et al., it is necessary to compute the Diffie-Hellman key to obtain the message authentication code (*MAC*) *first* in order to let user authenticate real identity with each other and achieve the desired security. During the process of executing the protocol, the server can be used for authenticating the real identity of both parties, and it will increase the volume of calculations and the level of complexity. Therefore, we propose an improved solutoin by attaching the messages of the users *A* and *B* (*X* and *Y*) to another party's identity in the original S-3PAKE protocol, executing the computation of a Hash function, and performing the computation by a server *S* without substituting the user passwords pw_1 and pw_2 into X' and Y' of the equations, because the equation of the Hash function can be offset with each other when *A* and *B* authenticate X' and Y' respectively, and the outcome will not be affected at all. Compared with the protocol

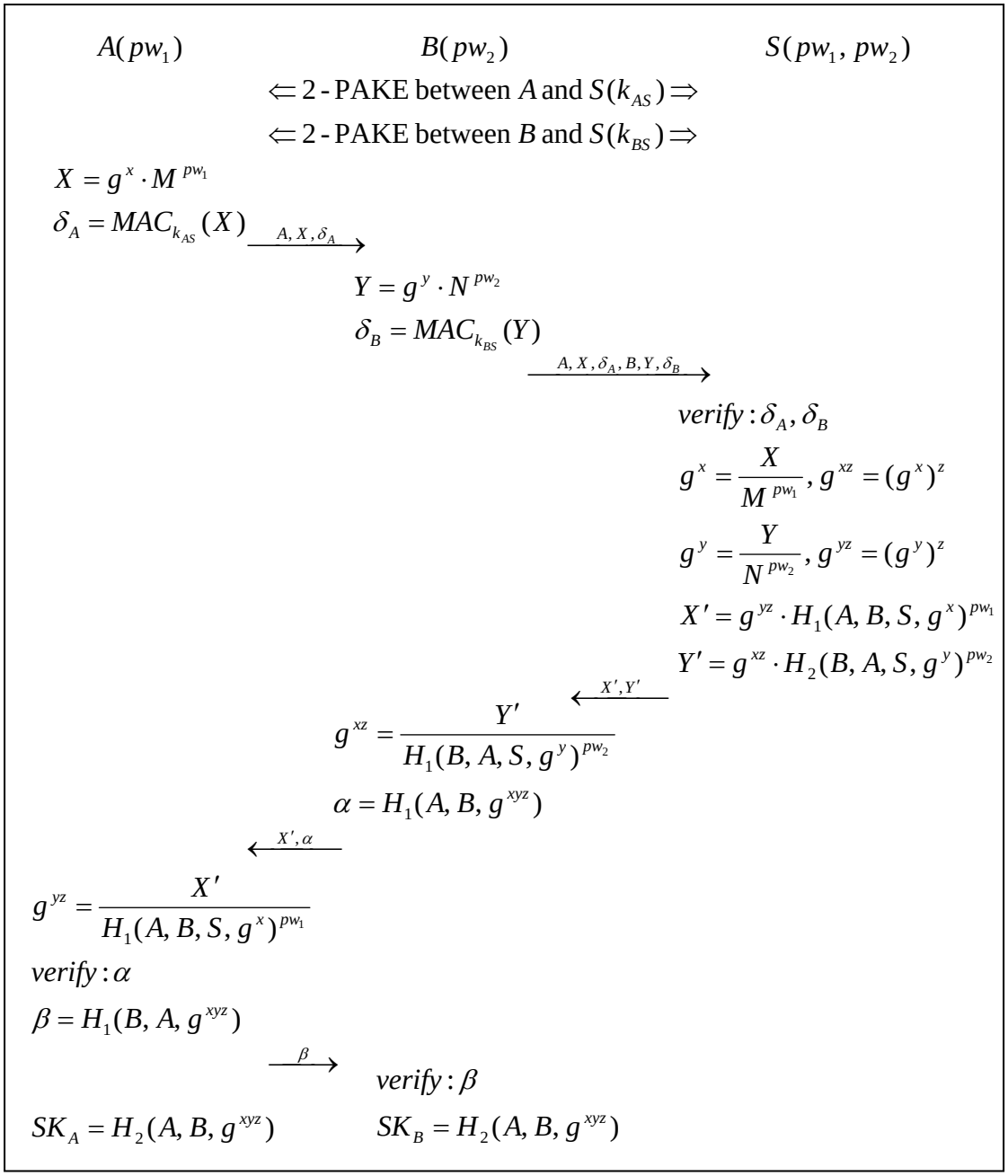

Fig. 3 Guo et al. improved protocol

proposed by Guo et al., our protocol reduces the volume of calculations without requiring the execution of the 2-PAKE protocol in advance, but maintaining the existing advantages of the protocol provided by the Guo et al. With reference to Fig. 4 for our improved protocol, $H_3(\cdot)$ is the collision-free one-way Hash function defined as $H_3(*) \to G$.

5. Security analysis

We are proposing an efficient approach that is secured, and is suitable to external public internet network.

Man-in-the-middle attack

In order for server to certify the user's authentication, proposed protocol contains another end user's authentication in

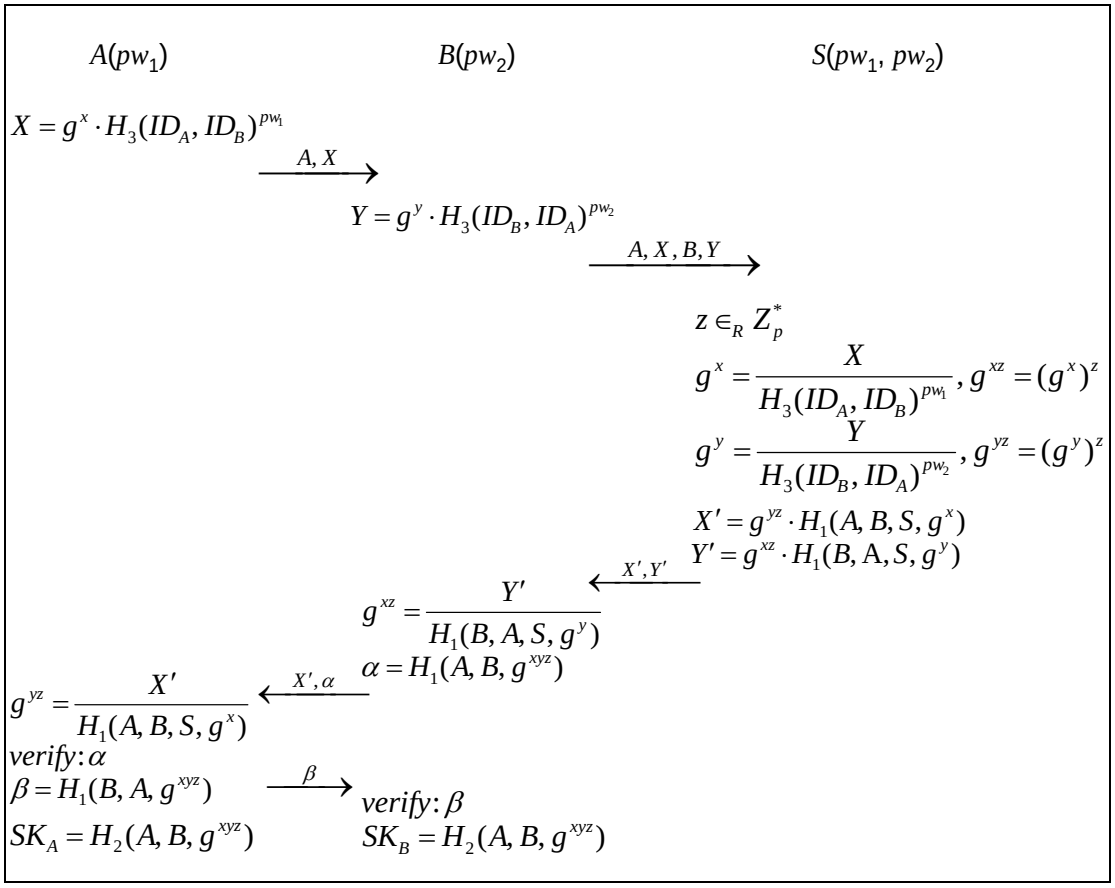

Fig. 4 Our protocol

information transmission. Assuming an attacker intercepts request as End User *A*, and processes session key with an identified End User *B* through S-3PAKE protocol which cannot authenticate the true identification of two users. Yet, our proposed protocol includes a two-party authentication and random number generated to perplex the user's password of which implements one-way hash function to identify the end user. The main purpose not only makes the server authenticate for each other by which attacker will not be able to calculate the feedback data X' or Y' and confirm data α or β as tending to process man-in-the-middle attack to obtain common session key as this proposed protocol can against man-in-the-middle attack as identified user can detect the legislative information of α or β.

Dictionary Attack

This proposed protocol works against dictionary attack because attacker process dictionary attack after intercepted messages whereas the information (X',Y') of this protocol responded does not include passwords of PW_1 and PW_2 for users. Nevertheless, the attacker cannot try any guessing password.

Off-line password guessing attack

This protocol we proposed includes two-party authentication and random number *x* and *y* as for perplex the user's password by which one-way hash function implement to identified two-side end users. Assume the attacker tries to guess and confirm the password through off-line after

intercepting messages firstly, the attacker must get the random number *x* or *y*, and then process password guessing and confirming. Secondly, the probability of value obtain are $1/2^{|x|}$ and $1/2^{|y|}$. In summary, attacker cannot obtain the password of a legislative user in polynomial time.

Replay attack

Assuming the attacker intercepts and records legislative information each session between user *A* and user *B*, and performs replay attacks subsequently. Since each session key would change according to the reselected random number *x* and *y'* that will prevent the attacker from obtaining previous record between user *A* and user *B* as α or β becomes inaccessible. In a word, the protocol we proposed can against replay attack.

Forward secrecy

Our protocol secures forward secrecy as the accession keys are produced by random number (x, y, z). Even though attacker was able to obtain accession key from intercepting communication, attacker still cannot decrypt previous accession information because the accession key is calculated by three-party random number (g^x, g^y, g^z) and (X, Y, Z) restored by perspective *A*, *B*, and *C* users. Not only attacker cannot obtain the random number (x, y, z) but also g^{xyz} is not being able to be decrypted. Therefore, this is a forward secrecy protocol.

6. Performance analysis

In this section, we will show that our proposed protocol is also an efficient one. For security consideration, let *p* of 1024 bits, the output size of secure one-way hash functions be 160 bits, and the individual identity of 32 bits.

In computation cost, both Lu and Cao protocol and Cuo et al. protocol needs twelve exponential operation. In our protocol have only eight exponential operations. In communication cost, to solve the Lu and Cao proposed protocol security problem that Guo et al. proposes an improve program by which two-side users individually implement 2-PAKE protocol to obtain Message Authentication Codes. However, we proposed an improved solution based on the protocol provided by Guo et al. without executing the 2-PAKE protocol first. Compared with the protocol proposed by Guo et al., our protocol reduces the volume of calculations, but maintaining the existing advantages of the protocol provided by the Guo et al., and achieves the same level of security. We evaluate the efficiency of our protocol and related protocols in Table 1.

7. Conclusions

S-3PAKE Protocol proposed by Lu and Cao declares itself can resist diversity known attacks, however, Guo et al. argues the protocol has security holes that easily attacked by man-in-the-middle attack and undetectable dictionary attack. In order to achieve security requirement Guo et al. proposes an improved program by which two-side users individually implement

Table 1 Efficiency comparison among our protocol and related protocols

	Lu and Cao protocol	Guo et al. protocol	Our protocol
Computation cost (exponential operation)	12	12	8
Number of communication	3	3	3
Communication cost (bits)	6560	7040	6560

2-PAKE protocol to obtain Message Authentication Codes (*MAC*) prior to creating a sharing session key. Through the communication can access to certificate the reality of server users and raise the computational complexity as well.

In this paper, proposed improvement for Guo et al. by processing a built-in data attached to other party for identity authentication to individual data. Not only Hash function computation makes each other processes identity authentication without accessing individual password but also it will not cause any influence to the result and keep the advantage of Guo et al. protocol. Compare to the Guo et al. improvement protocol, this paper its computation burden reduced and without processing 2-PAKE protocol but achieves safety demand and performing in updated network environment.

References

[1] R. Anand, M. Kumar and A. Jhingran, "Distributing E-Coupon on the Internet", ***Proceedings of the 9th Annual Conference of the Internet Society*** (INET'99), 1999.

[2] Steiner M, Tsudik G, Waidner M. Refinement and extension of encrypted key exchange. ***ACM Operating Systems Review*** 1995; 29(3):22-30.

[3] Ding Y, Horster P. Undetectable on-line password guessing attacks. ***ACM Operating Systems Review*** 1995; 29(4):77-86.

[4] Lin CL, Sun HM, Hwang T. Three party-encrypted key exchanges: attacks and a solution. ***ACM Operating Systems Review*** 2000; 34(4):12-20.

[5] Lin CL, Sun HM, Steiner M, Hwang T. Three-party encrypted key exchange without server public-keys. ***IEEE Communication Letters*** 2001; 5(12):497-9.

[6] Law L, Menezes A, Qu M, Solinas J, Vanstone S. An efficient protocol for authenticated key agreement. ***Designs, Codes and Cryptography*** March 2003; 28(2):119-34.

[7] Chang CC, Chang YF. A novel three-party encrypted key exchange protocol. ***Computer Standards and Interfaces*** 2004; 26(5):471-6.

[8] Lee TF, Hwang T, Lin CL. Enhanced three-party encrypted key exchange without server public keys. ***Computers and Security*** 2004; 23(7):571-7.

[9] Abdalla M, Pointcheval D. Simple password-based encrypted key exchange protocols, ***Topics in cryptology – CT-RSA 2005.*** In: LNCS. Springer-Verlag; 2005. pp. 191-208.

[10] R. Lu Z. Cao. Simple Three-party Key Exchange Protocol. ***Computers & Security***, Vol. 26, pp. 94-97, 2007.

[11] Guo H, Li Z, Mu Y, Zhang X. Cryptanalysis of Simple Three-Party Key Exchange Protocol, ***Computers & Security*** (2008), doi: 10.1016/j.cose.2008.03.001.