Applied Mathematics & Information Sciences
An International Journal

A Computation-Efficient Three-Party Encrypted Key Exchange Protocol

Cheng-Chi Lee[1], *Shun-Der Chen*[1] *and Chin-Ling Chen*[2]

[1] Department of Library and Information Science, Fu Jen Catholic University at Taipei, Taiwan, R.O.C.
[2] Department of Computer Science and Information Engineering, Chaoyang University of Technology at Taichung, Taiwan, R.O.C.

Received: Aug 28, 2011; Revised Dec 27, 2011; Accepted Apr 7, 2012
Published online: 1 Sep. 2012

Abstract: Recently, Chen et al. proposed a three-party encrypted key exchange (3PEKE) protocol with password authentication which is called CCLC-3PEKE. The protocol simultaneously possesses round and computation efficiencies. However, the protocol is vulnerable to replaying attacks. Since the protocol is currently one of the most superior of all 3PEKE protocols, it seems valuable to remedy the security weakness and enhance their efficiency. Hence, we shall propose an efficient 3PEKE (L-3PEKE) scheme. Compare with other 3PEKE protocols, our proposed L-3PEKE is more secure and efficient.

Keywords: Authentication, key exchange, password, password guessing attacks, three-party encrypted key exchange.

1. Introduction

In Internet, two communicating parties can communicate each other securely by using conventional symmetric-key cryptosystems such as the AES [19]. The two parties have a common session key to encrypt and decrypt their communicated messages by using symmetric-key cryptosystem. However, how do two parties securely obtain the common session key between them? This can be solved by using Diffie-Hellman key exchange protocol [7]. In 1992, Bellovin and Merritt firstly proposed an encrypted key exchange (EKE) family of key exchange protocols [2]. It is a password-based authentication and key agreement protocol. Two advantages of EKE are: (1) the communicating parties can use an easy-to-remember password to authenticate each other without being threatened by dictionary attacks [17]; (2) the communicating parties can share a common session key to encrypt and decrypt confidential messages.

In a large communication environment, EKE is unpractical because every two parties should share a password previously. If there are one thousand parties to communicate in this environment, each party should hold 999 passwords for EKE. Hence, an extension to EKE is proposed to enhance its practicality. The extension is called three-party encrypted key exchange protocol (3PEKE) in which a participant is allowed to share only one easy-to-remember password with a trusted server such that two participants can negotiate a common session key to communicate with each other secretly [3, 5, 10]. It can provide confidential communications between two participants over an insecure network. In 3PEKE, each party only holds himself/herself password.

1.1. Related Work

Since the 3PEKE is based on password authentication, protecting the low-entropy password from guessing attacks is crucial for password-based authentication schemes [15, 23]. Ding and Horster introduced three possible types of guessing attacks as follows: (1) detectable on-line password guessing attacks, (2) undetectable on-line password guessing attacks, and (3) off-line password guessing attacks. Among the three classes, off-line password guessing attacks is the most critical ones [8]. The proposed 3PEKE should protect against the three classes of password guessing attacks, off-line password guessing attacks especially.

In 1995, Steiner et al. proposed a 3PEKE protocol (STW-3PEKE) based on EKE protocols [21]. However, Lin et al. showed that STW-3PEKE is vulnerable to undetectable

* Corresponding author: e-mail: ryanchen@lins.fju.edu.tw

on-line password guessing attacks and proposed a new 3PEKE protocol (LSH-3PEKE) [13]. A solution of STW-3PEKE is also founded in SCH-3PEKE [22]. In both LSH-3PEKE and SCH-3PEKE, they used public key cryptosystems to resist the password guessing attacks. The communicating parties can encrypt his/her password and one-time messages with the server's public key as a request. Only the server can decrypt these messages with its own private key. This scheme can protect password against the three attacks. Unfortunately, employing public key cryptosystems involves time-consuming communication cost and the certificate infrastructure is needed. As a result, Lin et al. proposed an efficient protocol (LSSH-3PEKE) without using the public key cryptosystems [14]. However, the number of communicating rounds in LSSH-3PEKE is two more than that in LSH-3PEKE. Lee et al. proposed an enhanced scheme (LHL-3PEKE) [12]. The number of communicating rounds in LHL-3PEKE is one less than that in LSSH-3PEKE. After analyzing 3PEKE, Chang and Chang proposed a novel 3PEKE (CC-3PEKE) and mentioned the followings [4]: (1) The session key should be agreed by the communication parties instead of being assigned by the server directly. (2) Except the password, no extra secret information should be needed - the public key for example. (3) The server has to authenticate both communication parties. (4) Computation and round efficiencies should be provided at the same time. In 2005, Wen et al. proposed a provably secure 3PEKE using weil pairing [24]. Next, Nam et al. showed that their protocol is completely insecure [18].

In 2007, Lu et al. proposed a new 3PEKE (LC-3PEKE) to meet all the above mentioned [16]. Unfortunately, Chang [3] and Chung et al. [6], respectively, showed that LC-3PEKE suffers from undetectable on-line password guessing attacks. Chang proposed a practical 3PEKE (C-3PEKE) to remedy the security weakness [3]. Recently, Lee and Chang had founded that C-3PEKE suffers from off-line password guessing attacks [11]. In 2008, Chen et al. showed that CC-3PEKE suffers from undetectable on-line password guessing attacks and proposed an enhancement (CCLC-3PEKE) against undetectable on-line password guessing attacks [5]. The protocol uses super-poly-to-one trapdoor function which requires no certificate and can be efficiently constructed from one-way hash functions [1]. Until to now, CCLC-3PEKE is currently one of the most superior of all 3PEKE protocols. They claimed that the protocol is not only secure and efficient, but also meet the all requirements as follows: mutual authentication, resistance to three classes of password guessing attacks, round and computation efficiencies, and practicality. However, we shall show that CCLC-3PEKE is vulnerable to replaying attacks. Since their protocol is currently one of the most superior of all 3PEKE protocols, it seems valuable to remedy the security weakness and enhance their efficiency. Hence, we shall propose an efficient 3PEKE (L-3PEKE) scheme. Unlike other 3PEKE protocols, our proposed L-3PEKE not only uses no public key cryptosystem but also no symmetric cryptosystem. Compare with other protocols, L-3PEKE is more secure and efficient.

1.2. Organization of This Paper

The rest of this paper is organized as follows. We first review the CCLC-3PEKE in Section 2 and show its security weakness. In Section 3, we show an efficient 3PEKE to enhance the security and efficiency of CCLC-3PEKE. The requirements analyses of L-3PEKE are discussed in Section 4. Finally, some conclusions are drawn in Section 5.

2. The Weakness of CCLC-3PEKE

Firstly, we review CCLC-3PEKE in subsection 2.1. The security flaw of CCLC-3PEKE is shown in subsection 2.2.

2.1. A Review of CCLC-3PEKE

Chen et al. [5] had proposed a three-party encrypted key exchange protocol against undetectable on-line password guessing attacks which is called CCLC-3PEKE. In this section, we shall show that their protocol suffers from replaying attacks. Firstly, we review CCLC-3PEKE. The notations used throughout this paper are listed in Table 1. The details of CCLC-3PEKE are given as follows:

Step 1 A generates two random values r_A and R_A, and computes N_A=g^{R_A} mod$p$, K_{AS}=$N_A^{r_A} \bmod p$, $E3_{P_A}(N_A \bigoplus r_A)$, $F_S(r_A)$, and $f_{K_{AS}}(N_A)$. Then A sends (ID_A, ID_B, ID_S, $E3_{P_A}(N_A \bigoplus r_A)$, $F_S(r_A)$, $f_{K_{AS}}(N_A)$) to B as request.

Step 2 After receiving A's request, B generates two random values r_B and R_B, and computes N_B=g^{R_B}mod$p$, K_{BS}=$N_B^{r_B} \bmod p$, $E3_{P_B}(N_B \bigoplus r_B)$, $F_S(r_B)$, and $f_{K_{BS}}(N_B)$. Then B sends the received messages with ($E3_{P_B}(N_B \bigoplus r_B)$, $F_S(r_B)$, $f_{K_{BS}}(N_B)$) to S as request.

Step 3 Upon receiving B's messages, S firstly uses a trapdoor to obtain r_A/r_B from $F_S(r_A)$/$F_S(r_B)$. Next, S uses P_A/P_B and r_A/r_B to derive N_A/N_B from $E3_{P_A}(N_A \bigoplus r_A)$/$E3_{P_B}(N_B \bigoplus r_B)$. Then, S can compute K_{AS} and K_{BS} using r_A/r_B and N_A/N_B. Finally, S can authenticate A/B by verifying $f_{K_{AS}}(N_A)$/$f_{K_{BS}}(N_B)$. If it is correct, S believes that he/she is communicating with a legitimate A/B; otherwise, S regards A/B illegal and terminates the protocol.

S generates a random value R_S and computes $N_B^{R_S} \bmod p$/$N_A^{R_S} \bmod p$, and the corresponding hashed credential $f_{K_{AS}}(A, B, K_{AS}, N_B^{R_S})$/$f_{K_{BS}}(A, B, K_{BS}, N_A^{R_S})$. After that, S sends $N_B^{R_S}$/$N_A^{R_S}$ and $f_{K_{AS}}(A, B, K_{AS}, N_B^{R_S})$/$f_{K_{BS}}(A, B, K_{BS}, N_A^{R_S})$ to B.

Table 1 The Notations

Notations	Description
A/B	communication parties
S	the trusted server
$ID_A/ID_B/ID_S$	the identity of $A/B/S$
P_A/P_B	the password securely shared by A/B with S
$E3_P()$	a symmetric encryption scheme with a password P
$F_S()$	a super-poly-to-one trapdoor function (TDF) constructed from one-way hash function where only S knows the trapdoor
p	a large prime
g	an element of order q with modulus p
G	a finite cyclic group generated by g in Z_p
$R_A/R_B/R_S$	the random exponents chosen by $A/B/S$
r_A/r_B	the random exponents chosen by A/B
N_A/N_B	N_A=g^{R_A} mod$p$/N_B=g^{R_B} mod$p$
$f_K(\cdot)$	a pseudo-random function (PRF) indexed by K
K_{AS}/K_{BS}	a one-time strong key shared by A/B and S
K_{AB}	a common session key shared by A and B
T_A/T_B	a time-stamp generated by A/B
$\|\|$	a concatenation

Step 4Upon receiving the messages, B verifies $f_{K_{BS}}(A, B, K_{BS}, N_A^{R_S})$ to authenticate S. If it is correct, B believes that the received $N_A^{R_S}$ is valid and then computes the session key $K_{AB} = (N_A^{R_S})^{R_B} \bmod p$. Finally, B computes $f_{K_{AB}}(B, K_{AB})$. B sends $N_B^{R_S}$, $f_{K_{AS}}(A, B, K_{AS}, N_B^{R_S})$, and $f_{K_{AB}}(B, K_{AB})$ to A.

Step 5Upon receiving the messages, A verifies $f_{K_{AS}}(A, B, K_{AS}, N_B^{R_S})$ to authenticate S. If it is correct, A believes that the received $N_B^{R_S}$ is valid and then computes the session key $K_{AB} = (N_B^{R_S})^{R_A} \bmod p$. Next, A verifies $f_{K_{AB}}(B, K_{AB})$ to authenticate B. If it holds, A believes that B is a legitimate user and sends $f_{K_{AB}}(A, K_{AB})$ to B. Later, B can authenticate A by checking the validation of $f_{K_{AB}}(A, K_{AB})$.

Finally, A and B can share the common session key K_{AB} to encrypt and decrypt their communicated messages. In the meantime, mutual authentication between A and B is done.

2.2. Security Weakness

In this subsection, we shall show that CCLC-3PEKE is not robust enough against replaying atttacks from an evil E. An evil E can intercept transmitted messages from public channel and then forge other parties to communicate with S by replaying attacks. How CCLC-3PEKE suffers from replaying attacks is given as follows. In Step 2 of CCLC-3PEKE, E can intercept (ID_A, ID_B, ID_S, $E3_{P_A}(N_A \bigoplus r_A)$, $F_S(r_A)$, $f_{K_{AS}}(N_A)$, $E3_{P_B}(N_B \bigoplus r_B)$, $F_S(r_B)$, $f_{K_{BS}}(N_B)$). After a moment, E can replay the intercepted messages to S. In Step 3 of CCLC-3PEKE, S can verify these messages and believe that he/she is communicating with a legitimate A/B. Therefore, E can forge A and B to communicate with S successfully. After that, S will perform their following procedure and send some messages back to E. Although E cannot get the password of A/B, he/she can enable S to believe that he/she is communicating with A/B.

To solve this problem, we can easily add time-stamp [20] to their protocol. In next section, an efficient protocol is proposed to remedy this problem. As CCLC-3PEKE is currently one of the most superior of all 3PEKE approaches; it seems worthwhile and valuable to remedy this problem. In addition, our protocol has less computation cost and is superior to CCLC-3PEKE.

3. An Efficient Protocol L-3PEKE

In our protocol (called L-3PEKE), we do not only remedy the above security weakness, but improve the efficiency of CCLC-3PEKE. The requirements of our protocol are first listed in subsection 3.1. The details are presented in subsection 3.2.

3.1. Requirements

In this subsection, we set up five goals that L-3PEKE is aimed to achieve. The goals of this paper are roughly listed as follows and will be discussed in detail later in Section 4.

(1)Mutual authentication: Among A, B, and S, the legality of the three communication parties is ensured.
(2)Resistance to three classes of password guessing attacks: The proposed 3PEKE should be protected against undetectable on-line password guessing attacks, detectable on-line password guessing attacks, and off-line password guessing attacks.
(3)Resistance to replaying attacks: The proposed 3PEKE should be protected against replaying attacks.
(4)Round and computation efficiencies: Round and computation efficiencies are taken into consideration. The proposed 3PEKE preserves the advantages of the schemes in LSH, SCH, LSSH, CC, C, and CCLC in terms of round efficiency as well as computation efficiency.
(5)Practicality: The proposed 3PEKE employs super-poly-to-one trapdoor functions instead of public key cryptosystem. Therefore, no certificate is needed.

3.2. L-3PEKE

In this subsection, the details of L-3PEKE are given as follows:

Step 1 A generates two random values r_A and R_A, and computes N_A=g^{R_A} mod$p$, K_{AS}=$N_A^{r_A} \bmod p$, $(P_A \bigoplus N_A)$, $F_S(r_A)$, and $f_{K_{AS}}(N_A||T_A)$. Then A sends $(ID_A, ID_B, ID_S, (P_A \bigoplus N_A), F_S(r_A), f_{K_{AS}}(N_A||T_A), T_A)$ to B as request.

Step 2 After receiving A's request, B generates two random values r_B and R_B, and computes N_B=g^{R_B} mod$p$, K_{BS}=$N_B^{r_B} \bmod p$, $(P_B \bigoplus N_B)$, $F_S(r_B)$, and $f_{K_{BS}}(N_B||T_B)$. Then B sends the received messages with $((P_B \bigoplus N_B), F_S(r_B), f_{K_{BS}}(N_B||T_B), T_B)$ to S as request.

Step 3 Upon receiving B's messages, S firstly uses a trapdoor to obtain r_A/r_B from $F_S(r_A)/F_S(r_B)$. Next, S uses P_A/P_B to derive N_A/N_B from $(P_A \bigoplus N_A)/(P_B \bigoplus N_B)$. Then, S can compute K_{AS}=$N_A^{r_A} \bmod p$ and K_{BS}=$N_B^{r_B} \bmod p$ using r_A/r_B and N_A/N_B. After that, S checks if the time-stamp T_A/T_B is valid. If it is valid, S can authenticate A/B by verifying $f_{K_{AS}}(N_A||T_A)/f_{K_{BS}}(N_B||T_B)$. If it is correct, S believes that he/she is communicating with a legitimate A/B; otherwise, S regards A/B illegal and terminates the protocol.

S generates a random value R_S and computes $N_B^{R_S} \bmod p$/$N_A^{R_S} \bmod p$, and the corresponding hashed credential $f_{K_{AS}}(A, B, K_{AS}, N_B^{R_S})/f_{K_{BS}}(A, B, K_{BS}, N_A^{R_S})$. After that, S sends $N_B^{R_S}/N_A^{R_S}$ and $f_{K_{AS}}(A, B, K_{AS}, N_B^{R_S})/f_{K_{BS}}(A, B, K_{BS}, N_A^{R_S})$ to B.

Step 4 This Step is the same as CCLC-3PEKE.

Step 5 This Step is the same as CCLC-3PEKE.

Finally, A and B can share the common session key K_{AB} to encrypt and decrypt their communicated messages. In the meantime, mutual authentication between A and B is also done.

4. Discussions

In this section, we shall discuss whether L-3PEKE is able to satisfy all the requirements, mutual authentication, resistance to three classes of password guessing attacks, round and computation efficiencies, resistance to replaying attacks, and practicality, mentioned in Section 3.1 as follows.

4.1. Requirement 1: mutual authentication

The analysis of mutual authentication among A, B, and S is decomposed into three parts. How S can authenticate A/B is the first part. The second part is how A/B can authenticate S. The last part is how mutual authentication between A and B is done. Next, the three parts are examined in the following.

Firstly, A and B use the trapdoor function $F_S(\cdot)$ to hide the secret number r_A and r_B. Since only S knows the trapdoor, she/he can derive r_A and r_B from $F_S(r_A)$ and $F_S(r_B)$. Furthermore, S can derive N_A and N_B using the pre-shared password P_A and P_B to compute one-time key K_{AS} and K_{BS}. Then, S can verify the validation of $f_{K_{AS}}(N_A||T_A)$ and $f_{K_{BS}}(N_B||T_B)$. If it holds, S assures that A/B possesses the correct password, and $(N_A, T_A)/(N_B, T_B)$ is generated by A/B. Thus, S can authenticate A/B if they possess the correct password.

Secondly, upon receiving the message $f_{K_{AS}}(A, B, K_{AS}, N_B^{R_S})/f_{K_{BS}}(A, B, K_{BS}, N_A^{R_S})$, A/B can authenticate S individually through verifying S's knowledge of the corresponding one-time key K_{AS} and K_{BS}. If S is a legitimate server, he/she must know the trapdoor and possess the valid password to obtain a valid one-time key K_{AS} and K_{BS}. This clearly indicates that A/B can authenticate S.

Thirdly, assume that S is a trusted server. In 3PEKE protocols, the participants A and B can authenticate each other by the trusted server S's help. In all 3PEKE protocols, S must be the trusted server; otherwise, S can impersonate A or B since he/she knows P_A or P_B. If A/B is authenticated by S, S generated two evidences $f_{K_{AS}}(A, B, K_{AS}, N_B^{R_S})$ and $f_{K_{BS}}(A, B, K_{BS}, N_A^{R_S})$ as a prove of the $N_B^{R_S}$ and $N_A^{R_S}$, respectively. After that, A and B can verify if $N_B^{R_S}$ and $N_A^{R_S}$ are generated by S. As a result, A and B can compute the same session key $K_{AB} \equiv (N_A^{R_S})^{R_B} \bmod p \equiv g^{R_A R_S R_B} \bmod p \equiv (N_B^{R_S})^{R_A} \bmod p$. Then, mutual authentication between A and B is done by verifying $f_{K_{AB}}(A, K_{AB})/f_{K_{AB}}(B, K_{AB})$. Note that, the session key K_{AB} is a Diffie-Hellman public key [7], it is considered computational infeasible for an attacker to obtain the session key without knowing R_A or R_B.

4.2. Requirement 2: resistance to three classes of password guessing attacks

In this subsection, we shall show that L-3PEKE is secure against detectable on-line password guessing attacks, undetectable on-line password guessing attacks, and off-line password guessing attacks.

Firstly, an attacker may want to guess the password with detectable on-line password guessing attacks. He/she may impersonate A/B to mount on-line password guessing attacks. If an attacker impersonate A or B, S will detect it in Step 3 by verifying $f_{K_{AS}}(N_A||T_A)/f_{K_{BS}}(N_B||T_B)$. The attacker may perform the procedure many times. However, it does not pass S's verification under a different $f_{K_{AS}}/f_{K_{BS}}$. Once the attacker perform a small amount of failed guesses, S will be able to react it appropriately and terminate the protocol. Hence, the detectable on-line password guessing attacks cannot work in L-3PEKE.

Secondly, an attacker may want to guess the password with undetectable on-line password guessing attacks. A malicious outsider cannot guess the password with undetectable on-line password guessing attacks because S can authenticate the participants mentioned in Requirement 1. If a insider, said B, wants to guess A's password P_A, B can run the following procedure to mount undetectable on-line password guessing attacks. First, B obtains N'_A by XOR operation with a guessed P'_A. To verify the guess, B must have the key K_{AS} to verify $f_{K_{AS}}(N_A||T_A)$. Since K_{AS} is computed by r_A, there is no way to obtain it without the trapdoor. Moreover, B can receive the messages $N_A^{R_S}$ from S. However, in order to verify it, it is the difficulty of solving the discrete logarithm problems. Hence, undetectable on-line password guessing attacks cannot work in L-3PEKE. In addition, mutual authentication can protect against this attack.

Thirdly, an attacker may want to mount off-line password guessing attacks to guess the password P_A/P_B. He/she can intercept the messages $(P_A \bigoplus N_A)/(P_B \bigoplus N_B)$ from the public channel. Then the attacker tries to gain N_A/N_B and r_A/r_B to compute K_{AS}/K_{BS} used to verify $f_{K_{AS}}(N_A||T_A)/f_{K_{BS}}(N_B||T_B)$. However, the attacker cannot guess the password to verify his/her guess because there is no feasible way of knowing r_A/r_B from the trapdoor. Therefore, off-line password guessing attacks cannot work in L-3PEKE.

4.3. Requirement 3: resistance to replaying attacks

L-3PEKE is secure against replaying attacks. It uses the time-stamp to avoid replaying attacks [9]. It embedded the time-stamp T_A/T_B in $f_{K_{AS}}(N_A||T_A)/f_{K_{BS}}(N_B||T_B)$. S can check if the time-stamp T_A/T_B is valid and generated by A/B. Without the knowledge of K_{AS}/K_{BS}, no one can compute $f_{K_{AS}}(N_A||T_A)/f_{K_{BS}}(N_B||T_B)$. That is to say, the pattern is used only once. These messages cannot be intercepted for reuse because they are different values for each authentication. On the other hand, the server can test if $T' - T < \Delta T$ to prevent replaying attacks. Hence, L-3PEKE can protect against replaying attacks.

4.4. Requirement 4: round and computation efficiencies

In [3,5], Chang (C) and Chen et al. (CCLC), respectively, had demonstrated that their protocol preserves the advantages of the schemes in LSH [13], SCH [22], LSSH [14], LHL [12], LC [16], and CC [4] in terms of round efficiency as well as computation efficiency. As a result, CCLC-3PEKE and C-3PEKE are currently two of the most superior of all 3PEKE approaches in recent year. It had shown and explained in [3,5]. Therefore, we just compare our proposed L-3PEKE with C-3PEKE and CCLC-3PEKE. Table 2 shows the performance comparison of C-3PEKE, CCLC-3PEKE, and L-3PEKE.

Table 2 Performance comparison of C-3PEKE, CCLC-3PEKE, and L-3PEKE

Schemes	C-3PEKE			CCLC-3PEKE			L-3PEKE		
	A	B	S	A	B	S	A	B	S
Modular exponential	3	3	4	3	3	4	3	3	4
Public key en/decryption	0	0	0	0	0	0	0	0	0
Symmetric en/decryption	1	1	2	1	1	2	0	0	0
PRF operation	4	4	4	4	4	4	4	4	4
Hash/TDF operation	1	2	3	1	1	2	1	1	2
Random number	1	1	2	2	2	1	2	2	1
XOR operation	0	0	0	1	1	2	1	1	2
Round		5			5			5	
Security weakness		OPGA			RA			N	

OPGA: Off-line Password Guessing Attacks
RA: Replaying Attacks
N: No

It is seen that, round efficiency is the same. As shown in [3,5], our 3PEKE protocols have one or two less communication rounds than the LHL and LSSH. It requires less communication loading.

Taking computation efficiency into account, L-3PEKE is superior to the C-3PEKE and CCLC-3PEKE. The C-3PEKE and CCLC-3PEKE use some symmetric encryption/decryption. Our L-3PEKE has no symmetric encryption/decryption. It can reduce the heavy burden. Unlike other 3PEKE protocols, our proposed L-3PEKE not only uses no public key cryptosystem but also no symmetric cryptosystem. Hence, L-3PEKE is most efficient than other 3PEKE protocols. In addition, L-3PEKE not only preserves the superior merits of the C-3PEKE and CCLC-3PEKE but also fixes the security weaknesses.

4.5. Requirement 5: practicality

L-3PEKE is also practicality same as CCLC-3PEKE. It only employs super-poly-to-one trapdoor functions instead of public keys cryptosystem. Therefore, no certificate infrastructure is needed to be established. As mentioned above, it provides both round and computation efficiencies. Thus, L-3PEKE is also practical.

5. Conclusions

This paper had shown that CCLC-3PEKE suffers from replaying attacks and proposed a computation-efficient 3PEKE (L-3PEKE) which preserves the advantages of the schemes in LSH, SCH, LSSH, LHL, LC, CC, CCLC, C. According

to the analyses in Section 4, L-3PEKE is secure, efficient, and practical. Compare with two of the most superior of 3PEKE (C and CCLC) protocols, the proposed L-3PEKE is secure and has less computation cost.

Acknowledgement

This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC 100-2221-E-030-015.

References

[1] Bellare, M., S. Halevi, A. Sahai, and S. Vadhan, Many-to-one trapdoor functions and their relations to public-key cryptosystems, Proc. of Advanced in Cryptology-Crypto'98, Santa Barbara, CA, (1998).
[2] Bellovin, S.M. and M. Merrit, Encrypted key exchange: password-based protocols secure against dictionary attacks, Proc. of 1992 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, California, (1992), 72-84.
[3] Chang, Y.F., A practical three-party encrypted key exchange protocol with round efficiency, International Journal of Innovative Computing, Information and Control, 4(4), (2008), 953-960.
[4] Chang, Y.F. and C.C. Chang, A novel three-party encrypted key exchange protocol, Computer Standards & Interfaces, 26(5), (2004), 471-476.
[5] Chen, H.B., T.H. Chen, W.B. Lee, and C.C. Chang, Security enhancement for a three-party encrypted key exchange protocol against undetectable on-line password guessing attacks, Computer Standards & Interfaces, 30, (2008), 95-99.
[6] Chung, H.R., and W.C. Ku, Three weaknesses in a simple three-party key exchange protocol, Information Sciences, 178(1), (2008), 220-229.
[7] Diffie, W., and M. E. Hellman, New directions in cryptography, IEEE Trans. Inform. Theory, IT-22, (1976), 644-654.
[8] Ding, Y. and P. Horster, Undetectable on-line password guessing attacks, ACM Operating Systems Review, 29(4), (1995), 77-86.
[9] Hwang, M.S., Eric J.L. Lu, and I.C. Lin, Adding timestamps to the secure electronic auction protocol, Data & Knowledge Engineering, 40(2), (2002), 155-162.
[10] Lee, C.C., Chang, R.X., and Ko, H.J., Improving two novel three-party encrypted key exchange protocols with perfect forward secrecy, International Journal of Foundations of Computer Science, 21(6), (2010), 979-991.
[11] Lee, C.C., and Y.F. Chang, On security of a practical three-party key exchange protocol with round efficiency, Information Technology and Control, 37(4), (2008), 333-335.
[12] Lee, T.F., T. Hwang and C.L. Lin, Enhanced three-party encrypted key exchange without server public keys, Computers & Security, 23(7), 2004, 571-577.
[13] Lin, C.L., H.M. Sun and T. Hwang, Three-party encrypted key exchange: attacks and a solution, ACM Operating Systems Review, 34(4), (2000), 12-20.
[14] Lin, C.L., H.M. Sun, M. Steiner and T. Hwang, Three-party encrypted key exchange without server public-keys, IEEE Communications Letters, 5(12), (2001), 497-499.
[15] Lu, R. and Z. Cao, Off-line password guessing attack on an efficient key agreement protocol for secure authentication, International Journal of Network Security, 3(1), (2006), 35-38.
[16] Lu, R. and Z. Cao, Simple three-party key exchange protocol, Computers & Security, 26(1), (2007), 94-97.
[17] Morris, R. and K. Thompson, Password security: a case history, Communications of the ACM, (1979), 594-597.
[18] Nam J., 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), (2007), 1364-1375.
[19] National Institute of Standards and Technology, Specification for the Advanced Encryption Standard (AES), Federal Information Processing Standards Publication (FIPS) 197, Available at: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.
[20] Shen, J.J., C.W. Lin, and M.S. Hwang, Security enhancement for the timestamp-based password authentication scheme using smart cards, Computers & Security, 22(7), (2003), 591-595.
[21] Steiner, M., G. Tsudik and M. Waidner, Refinement and extension of encrypted key exchange, ACM Operating Systems Review, 29(3), (1995), 22-30.
[22] Sun, H.M., B.C. Chen, and T. Hwang, Secure key agreement protocols for three-party against guessing attacks, The Journal of Systems and Software, 75, (2005), 63-68.
[23] Tsai, C.S., C.C. Lee, and M. S. Hwang, Password Authentication Schemes: Current Status and Key Issues, International Journal of Network Security, 3(2), (2006), 101-115.
[24] Wen, H.A., T.F. Lee, and T. Hwang, Provably secure three-party password-based authenticated key exchange protocol using Weil pairing, IEE Proceedings-Communications, 152(2), (2005), 138-143.

Cheng-Chi Lee received the B.S. and M.S. in Information Management from Chaoyang University of Technology (CYUT), Taichung, Taiwan, in 1999 and in 2001. He researched in Computer and Information Science from National Chiao Tung University (NCTU), Taiwan, Republic of China, from 2001 to 2003. He received the Ph.D. in Computer Science from National Chung Hsing University (NCHU), Taiwan, in 2007. He was a Lecturer of Computer and Communication, Asia University, from 2004 to 2007. From 2007, he was an assistant professor of Photonics and Communication Engineering, Asia University. From 2009, he is an Editorial Board member of International Journal of Network Security and International Journal of Secure Digital Information Age. From 2010, he is now an assistant professor of Library and Information Science, Fu Jen Catholic University. His current research interests include

information security, cryptography, and mobile communications. Dr. Lee had published over 80+ articles on the above research fields in international journals.

Shun-Der Chen was born in Keelung, Taiwan,R.O.C., in 1965. He received his B.S in Mathematics from Fu Jen Catholic University, M.S. and Ph.D. in Computer Science from National Tsing Hua University, Taiwan in 1988, 1990 and 1996, respectively. Now, he is an Associate Professor in the Department of Library and Information Science at Fu Jen Catholic University in Tainan. His current research interests include Information retrieval, Data mining, Natural language processing and Digital archives.

Chin-Ling Chen was born in Taiwan in 1961. He received the B.S. degree in Computer Science and Engineering from the Feng Chia University in 1991; the M.S. degree and Ph.D. in Applied Mathematics at National Chung Hsing University, Taichung, Taiwan, in 1999 and 2005 respectively. He is a member of the Chinese Association for Information Security. From 1979 to 2005, he was a senior engineer at the Chunghwa Telecom Co., Ltd. He is currently an associate professor of the Department of Computer Science and Information Engineering at Chaoyang University of Technology, Taiwan. His research interests include cryptography, network security and electronic commerce.