ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2012, Vol. 41, No. 1

SECURITY ENHANCEMENT ON SIMPLE THREE PARTY PAKE PROTOCOL

Shirisha Tallapally

Malla Reddy Engineering College, Hyderabad, Andra Pradesh, India
e-mail: shirisha27@yahoo.co.in

crossref http://dx.doi.org/10.5755/j01.itc.41.1.842

Abstract. In the field of cryptography, the three-party authenticated key exchange protocol is an important tool, especially in the secure communication areas. In this protocol, two clients share a human-memorable password with a trusted server whereby the two clients receive a secure session key. Most recently, Huang proposed a simple and efficient three party password-based key exchange protocol. She claimed that the proposed protocol is secure against various attacks. However, Yoon and Yoo proved an undetectable online password guessing attack on Huang's protocol. In the present paper, an unknown key share attack on Huang's three party PAKE protocol using undetectable online password guessing attack is demonstrated. Additionally, an alternative protocol that eliminates this attack is proposed. Moreover, the proposed protocol requires only four message transmission rounds.

Keywords: Huang's three party PAKE protocol; undetectable online password guessing attack; unknown key share attack; password.

1. Introduction

Three party authenticated key exchange protocol (PAKE) is invariably an important cryptographic tool in the area of secure communication whereby a pair of clients communicate over a public unreliable channel while generating a secure session key. This, rather simple and efficient protocol requires the users to memorize low-entropy password. In a three-party PAKE protocol, each client first shares a human-memorable password with a trusted server, and then when two clients want to agree a session key, they resort to the trusted server for authenticating each other. However, these types of password based key exchange protocols are susceptible to password guessing attacks since users generally prefer easy to remember passwords. During a password guessing attack, the attacker's main aim is to retrieve the legitimate communication party's password. In general, the password guessing attacks can be divided into three classes and they are listed below [1]:

- **Detectable online password guessing attacks**: An attacker attempts to use a guessed password in an online transaction. He/she verifies the correctness of his/her guess using the response from server. A failed guess can be detected and logged by the server.
- **Undetectable online password guessing attacks**: Similar to Detectable online password guessing attacks, an attacker tries to verify a password guess in an online transaction. However, a failed guess cannot be detected and logged by server, as server is not able to distinguish an honest request from a malicious one.
- **Off-line password guessing attacks**: An attacker guesses a password and verifies his/her guess off-line. No participation of server is required, so the server does not notice the attack.

The first ever practical key exchange protocol was proposed by Diffie-Hellman [2]. This landmark finding was followed up by other two-party PAKE protocols [3, 4, 5, 6, 7]. Bellovin and Merritt [8] proposed the first PAKE protocol, known as Encrypted key Exchange (EKE) protocol. However, the two party PAKE protocols are only suitable for the client-server architecture. Many researchers have recently begun to study the three-party PAKE protocols [9, 10, 11, 12, 13, 14, 15, 16, 17].

Recently, Lu and Cao [16] proposed a simple three-party key exchange (STPKE) protocol based on the chosen-basis computational Diffie-Hellman (CCDH) assumption. There protocol was claimed to be able to resist various attacks and is superior to similar protocols and comparatively more efficient. Nevertheless, Kim and Choi [18] proved that the STPKE protocol is vulnerable to undetectable online password guessing attacks by using formal description. They proposed an alternative protocol (STPKE' protocol).

In 2009, Huang proposed a novel, simple three party password – based key exchange (3 PAKE) protocol [19]. Yoon and Yoo proved an undetectable online password guessing attack on 3 PAKE protocol [20].

In the current study, an unknown key share attack (A client 'B' not supposedly involved in a protocol run can end up sharing a session key with client 'A', but 'A' thinking it is sharing key with client 'C') using undetectable online password guessing attack is demonstrated on the Huang's protocol.

The paper is organized as follows: section 2 reviews Huang's protocol and undetectable online password guessing attack on Huang's protocol. Section 3 describes unknown key share attack on Huang's protocol. Section 4 describes the proposed protocol. Section 5 reports the security and efficiency analyses and the concluding remarks are made in section 6.

2. Review of Huang's 3 party PAKE protocol

The notations used in the paper are defined as follows:

p : a large prime

g : generator with order q ($q \geq 2^{256}$)

S: trusted server

pw_A : password shared between A and server

pw_B : password shared between B and server

A, B : two identity numbers of clients (users)

h ( ): a public one-way hash function

x , r_A : random numbers chosen by client A

y , r_B : random numbers chosen by client B

F_S ( ): trapdoor function

2.1. Huang's 3 party PAKE protocol

This section briefly reviews Huang's 3 party PAKE protocol.

Step 1. User A chooses a random number x and computes $R_A = (g^x \bmod p) \oplus h(pw_A, A, B)$, then sends $(A,\ R_A)$ to user B, where $\oplus$ signifies an exclusive-or operator.

Step 2. User B chooses a random number y and computes $R_B = (g^y \bmod p) \oplus h(pw_B, A, B)$, then sends (A, R_A, B, R_B) to server.

Step 3: Server finds $g^x = R_A \oplus h(pw_A, A, B)$ and $g^y = R_B \oplus h(pw_B, A, B)$. Server selects a random number z and computes $a = g^{xz}, b = g^{yz}$. Server calculates $Z_A = b \oplus h(pw_A, g^x)$ and $Z_B = a \oplus h(pw_B, g^y)$. Server sends Z_A, Z_B to user B .

Step 4: User B finds $a = Z_B \oplus h(pw_B, g^y)$ and the session key $K = a^y = g^{xyz}$. He computes $S_B = h(K, B)$ and forwards Z_A, S_B to User A.

Step 5: User A finds $b = Z_A \oplus h(pw_A, g^x)$ and the session key $K = b^x = g^{xyz}$. Then A checks whether $S_B = h(K, B)$ holds or not. If it does not hold, A terminates the protocol. Otherwise, A is convinced that $K = g^{xyz}$ is a valid session key. Then he computes $S_A = h(K, A)$ and sends it to user B.

Step 6: Upon receiving S_A, user B computes $S_A = h(K, A)$ and checks whether it is correct or not. If it is not correct, B terminates the protocol. Otherwise, K is a valid session key. Both the users A and B can use this session key K for secure communication. Here, K is only used for one session.

2.2. Undetectable online password guessing attack

This section demonstrates undetectable online password guessing attack on Huang's three party PAKE protocol, demonstrated by Yoon & Yoo, which allows one party's password to be revealed to another party by a guessing attack. i.e. B can guess A's password pw_A [20].

Step1: User A chooses a random number x and computes $R_A = (g^x \bmod p) \oplus h(pw_A, A, B)$, then sends $(A,\ R_A)$ to user B.

Step2: User B guesses a password, pw_A^* and finds $h(pw_A^*, A, B)$. B calculates $R_A \oplus h(pw_A^*, A, B)$, which is $(g^{x^*} \bmod p)$. Let $g^{x^*} \bmod p = g^y \bmod p$.

Step 3: $R_B = (g^y \bmod p) \oplus h(pw_B, A, B)$ is computed. B forwards (A, R_A, B, R_B) to server.

Step 4: Upon receiving (A, R_A, B, R_B), the server first uses pw_A and pw_B to compute $g^x = R_A \oplus h(pw_A, A, B)$ and $g^y = R_B \oplus h(pw_B, A, B)$ respectively. Then S chooses another random number z and computes $a = g^{xz} \bmod p, b = g^{yz} \bmod p$. Finally, S sends (Z_A, Z_B) to user B, where $Z_A = b \oplus h(pw_A, g^x)$ and $Z_B = a \oplus h(pw_B, g^y)$.

Step 5: Now B finds a and b where $a = Z_B \oplus h(pw_B, g^y)$ and $b = Z_A \oplus h(pw_A^*, g^{x^*})$. If $a = b$ then the password guessed is correct.

Therefore, B can get the real password pw_A of A using an undetectable online password guessing attack, wherein B runs the server S without detected by A (No participation of A is required).

Figure 1 illustrates Undetectable online password guessing attack.

3. Unknown key share attack on Huang's 3 party PAKE protocol

This section demonstrates unknown key share attack on Huang's three party PAKE protocol.

Let us assume A and C want to establish a session key. Here B can perform an unknown key share attack, using the password of A by first mounting undetectable online password guessing attack as shown in Fig. 1. Now, B knows A's password. The client B who is not supposedly involved in a protocol run may end up sharing a session key with client A while A thinks he/she is sharing key with client C. Fig. 2 illustrates Unknown key share attack.

Step 1: User A chooses a random number x and computes $R_A = (g^x \bmod p) \oplus h(pw_A, A, C)$, then sends (A, R_A) to user C.

Step 2: User B intercepts (A, R_A) this message. As he/she has already got the pw_A (as shown in section 2.2), he/she computes $h(pw_A, A, C)$ and $g^x = R_A \oplus h(pw_A, A, C)$. Now, R_A will be modified as: $R_A' = g^x \oplus h(pw_A, A, B)$.

Step 3: User B chooses a random number y and computes $R_B = (g^y \bmod p) \oplus h(pw_B, A, B)$, then sends (A, R_A', B, R_B) to server.

Step 4: As S believes that A and B want to establish a session key, it finds $g^x = R_A' \oplus h(pw_A, A, B)$ and $g^y = R_B \oplus h(pw_B, A, B)$. S chooses a random number z and computes $a = g^{xz}, b = g^{yz}$ and $Z_A = b \oplus h(pw_A, g^x)$, $Z_B = a \oplus h(pw_B, g^y)$. Then S sends Z_A, Z_B to B.

Step 5: B finds $a = Z_B \oplus h(pw_B, g^y)$ and $K = a^y = g^{xyz}$.

Step 6: B calculates $S_C = h(K, C)$ to make A believe that the message is from C. It sends Z_A, S_C to A.

Step 7: A finds $b = Z_A \oplus h(pw_A, g^x)$, $K = b^x = g^{xyz}$ and verifies S_C, believing that the message is from C (But it is from B).

A ends up thinking it is sharing a key with C. But B has obtained the key.

4. The proposed protocol

Trap door function concept was introduced into Encrypted key exchange protocol by Chang and Chang [21]. As a result of this, most of the attacks i.e. unknown key share attack, impersonation of the initiator and impersonation of the responder attacks are avoided. Many to one trapdoor function [22] is utilized in the proposed protocol to prevent password guessing attacks since trapdoor can be opened only by the server.

This section presents the proposed protocol. In the proposed protocol, parallel message transmission mechanism is utilized to reduce one message transmission round in comparison with the Huang's 3 party PAKE protocol. Fig. 3 illustrates the proposed protocol.

Step1: User A generates two random number x, r_A and computes $F_S(r_A)$ and $R_A = (g^x \bmod p) \oplus h(r_A, pw_A, A, B)$, then sends $(A, R_A, F_S(r_A))$ to server. User B generates two random number y, r_B and computes $F_S(r_B)$ and

User A	User B	Server
x, pw_A	y	z
1) $R_A = g^x \oplus h(pw_A, A, B)$ $\xrightarrow{A,R_A}$		
	2) guess password pw_A^*	
	find $h(pw_A^*, A, B)$.	
	[h( ) is a public one-way hash function]	
	find $R_A \oplus h(pw_A^*, A, B), = g^{x*}$	
	let $g^{x^*} = g^y$	
	$R_B = g^y \oplus h(pw_B, A, B)$	
	$\xrightarrow{A,R_A,B,R_B}$	
		3) $g^x = R_A \oplus h(pw_A, A, B)$
		$g^y = R_B \oplus h(pw_B, A, B)$
		$a = g^{xz}, b = g^{yz}$
		$Z_A = b \oplus h(pw_A, g^x)$
		$Z_B = a \oplus h(pw_B, g^y)$.
	$\xleftarrow{Z_A,Z_B}$	
	4) $a = Z_B \oplus h(pw_B, g^y)$	
	$b = Z_A \oplus h(pw_A^*, g^{x^*})$	
	check $a = b$	
	If $a = b$ password guessed is correct	

Figure 1. Undetectable on-line password guessing attack on Huang's 3 party PAKE protocol

$R_B = (g^y \bmod p) \oplus h(r_B, pw_B, A, B)$, then sends $(B, R_B, F_S(r_B))$ to server.

Step 2: Server extracts r_A, r_B from $F_S(r_A)$ and $F_S(r_B)$ and finds $g^x = R_A \oplus h(r_A, pw_A, A, B)$ and $g^y = R_B \oplus h(r_B, pw_B, A, B)$. S selects a random number z and computes $a = g^{xz}, b = g^{yz}$. Server calculates $Z_A = b \oplus h(r_A, pw_A, g^x)$ and $Z_B = a \oplus h(r_B, pw_B, g^y)$, sends Z_A to user A and Z_B to user B.

Step 3: User A finds $b = Z_A \oplus h(r_A, pw_A, g^x)$ and the session key $K = b^x = g^{xyz}$. Then he/she computes $S_A = h(K, A)$ and sends to User B. Meanwhile, User B finds $a = Z_B \oplus h(r_B, pw_B, g^y)$ and the session key $K = a^y = g^{xyz}$.

Step 4: User B computes $S_B = h(K, B)$ and sends to User A. A and B verifies S_B and S_A, respectively.

User A | User B | Server

x, pw_A | y, pw_B

1) $R_A = g^x \oplus h(pw_A, A, C)$

$\xrightarrow{A, R_A}$

'B' intercepts this message i.e. it gets R_A.
(from Undetectable on-line password guessing attack it has already obtained A's password)

compute $h(pw_A, A, C)$

compute $g^x = R_A \oplus h(pw_A, A, C)$

Now R_A will be modified as follows:

$R_A^{'} = g^x \oplus h(pw_A, A, B)$

2') $R_B = g^y \oplus h(pw_B, A, B)$

$\xrightarrow{A, R_A', B, R_B}$

('S' believes that 'A' & 'B' want to establish a session key)

3) $g^x = R_A' \oplus h(pw_A, A, B)$

$g^y = R_B \oplus h(pw_B, A, B).$

$a = g^{xz}, b = g^{yz}$

$Z_A = b \oplus h(pw_A, g^x)$

$Z_B = a \oplus h(pw_B, g^y)$

$\xleftarrow{Z_A, Z_B}$

4) $a = Z_B \oplus h(pw_B, g^y)$

$K = a^y = g^{xyz}.$

To make 'A' believe that the message is from 'C', B will calculate: $S_C = h(K, C)$

$\xleftarrow{Z_A, S_C}$

5) $b = Z_A \oplus h(pw_A, g^x)$

$K = b^x = g^{xyz}$

Verify: $S_C = h(K, C)$

(Confirms that the message is from 'C' but it is from 'B')

$S_A = h(K, A)$

$\xrightarrow{S_A}$

Check: $S_A = h(K, A)$

Figure 2. Unknown key share attack on Huang's 3 party PAKE protocol

User A	User B	Server
x, r_A, pw_A	y, r_B, pw_B	z
$F_S(r_A)$	$F_S(r_B)$	
$R_A = g^x \oplus h(r_A, pw_A, A, B)$	$R_B = g^y \oplus h(r_B, pw_B, A, B)$	

$$\xrightarrow{A, R_A, F_S(r_A)}$$

$$\xrightarrow{B, R_B, F_S(r_B)}$$

Extract r_A, r_B from $F_S(r_A)$ and $F_S(r_B)$

$$g^x = R_A \oplus h(r_A, pw_A, A, B)$$
$$g^y = R_B \oplus h(r_B, pw_B, A, B)$$
$$a = g^{xz}, b = g^{yz}$$
$$Z_A = b \oplus h(r_A, pw_A, g^x)$$
$$Z_B = a \oplus h(r_B, pw_B, g^y)$$

$$\xleftarrow{Z_A} \qquad \xleftarrow{Z_B}$$

User A	User B
$b = Z_A \oplus h(r_A, pw_A, g^x)$	$a = Z_B \oplus h(r_B, pw_B, g^y)$
$K = b^x = g^{xyz}$	$K = a^y = g^{xyz}$.
$S_A = h(K, A)$	$S_B = h(K, B)$

$$\xrightarrow{S_A}$$
$$\xleftarrow{S_B}$$

Verify : $S_B = h(K, B)$ Verify : $S_A = h(K, A)$

Figure 3. The proposed protocol

5. Security and efficiency analyses

The following are the security requirements to be met by a password key exchange protocol [21]:

- resistance to the password guessing attacks;
- transmission round.

The proposed protocol is satisfying the above requirements. In this section, we present a brief report on the security analyses of our protocol with respect to requirements.

Resistance to the password guessing attacks: First, a malicious attacker may want to guess the password with undetectable online password guessing attack. Then he/she computes $R_A = (g^x \bmod p) \oplus h(r_A, pw_A^{*}, A, B)$ and sends $A, R_A, F_S(r_A)$ to server. Server extracts r_A and computes $g^x = R_A \oplus h(r_A, pw_A^{*}, A, B)$ (this will not be equal to the obtained g^x which is equal to $R_A \oplus h(r_A, pw_A, A, B)$). Since g^x itself is wrong, the computed 'a', 'Z_A' will not be correct.Then the retrieved key is not a valid one. Hence, undetectable online password guessing attacks fail to be mounted on the proposed protocol, by a malicious attacker from outside.

If 'B' tries to guess A's password then he intercepts the message $R_A = (g^x \bmod p) \oplus h(r_A, pw_A, A, B)$ and guesses a password pw_A^*. But $h(r_A, pw_A^*, A, B)$ cannot be computed since r_A is protected by Trapdoor function and it is impossible to open the trapdoor function until trapdoor is known. Hence 'B' cannot mount an undetectable on-line password guessing attack on the proposed protocol.

Second, an adversary may try to mount off-line password guessing attack to guess the password. Upon intercepting $R_A, F_S(r_A)$ or $R_B, F_S(r_B)$, he cannot open the trapdoor and get r_A or r_B. Hence, off-line password guessing attacks cannot be mounted on the proposed protocol.

Unknown Key share attack: If any attacker C tries to mount an unknown key share attack, then he/she frames his own message C, R_C, $F_S(r_C)$ (where $R_C = g^w \oplus h(r_C, pw_C, A, C)$) and sends it to server. Server believes the communication is between A & C, he extracts r_A, r_C from $F_S(r_A)$ and $F_S(r_C)$ and finds $g^x = R_A \oplus h(r_A, pw_A, A, C)$ (this will not be equal to actual g^x as the actual $g^x = R_A \oplus h(r_A, pw_A, A, B)$) and $g^w = R_C \oplus h(r_C, pw_C, A, C)$. As g^x calculated is wrong the key obtained will not be a valid session key. Hence it is impossible to mount unknown key share attack until password of A or B is known.

Perfect forward secrecy: The enhanced protocol has the perfect forward secrecy. The session key is computed as follows: $K = b^x = a^y$. If the attacker gets Z_A or Z_B , then in order to obtain the session key, he should know r_A or r_B and x or y. Since this is not possible he cannot get the key.

The session keys generated in different sessions are independent since r_A, x, r_B and y are randomly chosen by A & B, respectively. This indicates that the attacker cannot obtain previous session keys even if he obtains the session key used in this run.

Known-Key Security: In the enhanced protocol, r_A, x, r_B and y are randomly chosen by A and B, and are independent among protocol executions. This leads to the in-vulnerability of Known-Key security.

Trivial attack: An attacker may directly try to compute the session key from Z_A or Z_B. However, due to the intractability of DLP and the one-wayness of hash function, the trivial attack is not possible in the proposed protocol.

Replay attack: Since the user does not know the password, the random number is protected using one way trapdoor hash function, as a result of this, he cannot retrieve the key $K = g^{xyz}$. The proposed protocol is in-vulnerable of this attack.

Transmission round: The efficiency of Three Party password-based key exchange protocol is measured in terms of the number of transmission rounds (steps) and the computation complexity . Table 1 shows the comparison analyses of the proposed protocol and Huang's three party PAKE protocol. From the view point of the transmission round, our protocol adopts the parallel message transmission mechanism (i.e A→S and B→S) to achieve fewer transmission rounds than the protocol proposed by Huang (i.e. A→B→S).

Table 1. Comparison between the proposed protocol and Huang's three party PAKE protocol

	Proposed protocol			Huang's 3 PAKE protocol		
Communication party	A	B	S	A	B	S
Modular exponentiation	2	2	2	2	2	2
Hash function	3	3	2	3	3	2
TDF (Trap door function)	1	1	2	0	0	0
Random number	2	2	1	1	1	1
Total round numbers		4			5	

6. Conclusion

Three-party authenticated key exchange protocol is an important cryptographic tool in the secure communication areas. By this protocol two clients will share human-memorable passwords with a trusted server. In turn they obtain a secure session key. Most recently, Huang proposed a simple and efficient three party password-based key exchange protocol. She claimed that the proposed protocol is secure against various attacks.

The present study demonstrated an unknown key share attack on Huang's 3 PAKE protocol. Additionally, an alternative protocol is proposed to eliminate these attacks. Moreover, the proposed protocol is efficient and requires only four message transmission rounds and it is secure.

References

[1] **Ding Y., Horster P.** Undetectable on-line password guessing attacks. *ACM Operat Syst Rev* 1995, 29(4), 77– 86.

[2] **W. Diffie, M. Hellman**. New Directions in cryptography. *IEEE Transactions on Information theory*, 1976, 22(6) ,644-654.

[3] **K. Kobara, H. Imai**. Pretty-simple password-authenticated key exchange under standard assumptions. *IEICE Transactions, E85-A*, oct.2002, (10), 2229-2237. Also available at http://eprint.iacr.org/2003/038/.

[4] **Bellare M., Pointcheval D., Rogaway P.** Authenticated key exchange secure against dictionary attacks. *Proceedings of the 2000 Advances in Cryptology (EUROCRYPT'2000). Berlin, Germany: Springer-Verlag*, 2000, 139-155.

[5] **E. Bresson, O. Chevassut, D. Pointcheval.** New security results on encrypted key exchange. *Proc. PKC 2004, LNCS 2947*, *Springer-Verlag,* Mar. 2004, 145-158. Mar.

[6] **M. Abdalla, D. Pointcheval**. Simple Password-Based Encrypted Key Exchange Protocols. *Proc. of Topics in Cryptology - CT-RSA 2005, LNCS* 3376, *Springer-Verlag*. 191-208,

[7] **M. Abdalla, O. Chevassut, D. Pointcheval.** One-time verifier-based encrypted key exchange. *Proc. of PKC '05, LNCS. Springer-Verlag*, 2005, 3386, 47-64.

[8] **S. M. Bellovin, M. Merritt**. Encrypted key exchange: Password-based protocols secure against dictionary attacks. *Proc. 1992 IEEE Symposium on Security and Privacy, 7 IEEE Computer Society Press,* May1992, 72-84.

[9] **C.L. Lin, H.M. Sun, M. Steiner, T. Hwang**. Three-party encrypted key exchange without server's public keys. *IEEE Communications Letters*, 2001,5(12), 497-499.

[10] **Bresson E., Chevassut O., Pointcheval D.**. New security results on encrypted key exchange. *Proceedings of the 7th International Workshop on Theory and Practice in Public Key Cryptography (PKC'2004), Singapore*, *Berlin, Germany: Springer-Verlag,* 2004: 145-158.

[11] **S. W. Lee, H. S. Kim, K. Y. Yoo** . Efficient verifier-based key agreement protocol for three parties without server's public key. *Applied Mathematics and Computation*, 2005,167(2), 996-1003.

[12] **T.F. Lee , T. Hwang, C.L. Lin** . Enhanced three-party encrypted key exchange without server public keys. *Computers and Security*, 2004,23(7),571-577.

[13] **C. L. Lin, H.M. Sun, T. Hwang**. Three-party encrypted key exchange: attacks and a solution. *ACM Operating Systems Review*, 2000, 34(4), 12-20.

[14] **Abdalla M., Fouque P.-A., Pointcheval D.** Password-based authenticated key exchange in the three-party setting. *In: IEE proceedings in Information Security*, March 2006 153(1), 27–39.

[15] **Abdalla M., Pointcheval D.**. Interactive Diffie-Hellman Assumptions with Applications to Password-based Authentication. *Proceedings of the 9th International Conference on Financial Cryptography (FC'2005), Roseau, Dominica, 2005. Berlin, Germany: Springer-Verlag,* 2005, 341-356.

[16] **R. Lu, Z. Cao**. Simple three-party key exchange protocol. *Computers and Security*, 2007, 26, 94-97.

[17] **R. Padmavathy**. Improved three party EKE protocol. Information Technology and control, 2010, 39, 220-226.

[18] **H. S. Kim, J. Y. Choi**. Enhanced Password-based simple three-party Key exchange protocol. *Computers and Electrical Engineering* 2009. 35(1), 107-114.

[19] **H. F. Huang**. A simple three-party password-based key exchange protocol. *International Journal of communication systems*. 2009,22, 857-862.

[20] **E. J. Yoon, K. Y. Yoo**. Cryptanalysis of a simple three-party password-based key exchange protocol. *International Journal of Communication systems,* April 2011, 24(4),532-542.

[21] **C.C. Chang. Y.F. Chang**. A novel three party encrypted key exchange protocol, *Computer Standards and Interfaces*,2004, 26(5), 471-476.

[22] **M. Bellare, S. Halevi, A. Sahai, S. Vadhan**. Many-to-one trapdoor functions and their relations to public-key cryptosystems, *Proceedings of Advanced in Cryptology-Crypto'98, Lecture Notes in Computer Science*

Received July 2011.