Enhanced Password-Based Simple Three-Party Key Exchange Protocol

[1]P.Karthikeyan,
Student,
M.Tech, Dept.of CSE,
Bharaht University,
Chennai.
E-Mail: karthikprakasam@yahoo.com

[2]G.Michael,
Assistant Professor,
Department of CSE,
Bharaht University,
Chennai.
E-Mail: micmgeo@yahoo.co.in

Abstract

Going alongside the fast development of internet technologies, folks will create a good amount of service requests to service supplier's victimization mobile devices anytime and anyplace. However, the service requester and also the service suppliers might not trust one another and that they could find at completely different domain. They require a communal trusty third party to assist them establish a shared session key for secure communications. It's questionable triangular key exchange. Recently, several password-based triangular key exchange protocols were planned against varied well-known security threats. In those protocols, to prevent the arcanum idea attack, a wide used method is to use public-key and/or symmetric-key cryptosystems to shield the changed messages. As we tend to legendary, the encrypted and decrypted operations in a very public-key cryptosystem square measure long. During this paper, we tend to propose a password-based triangular key exchange protocol with the computation-efficiency while not victimization public-key systems. Finally, we tend to prove the security of the planned protocol within the random oracle model.

Keywords: cryptography; separate index problem; on-line undetectable arcanum estimate attack; three-party key exchange.

Introduction

Today, folks have several opportunities to obtain services or resources from application servers by exploitation their mobile devices through the Internet. However, each of the shoppers and also the Servers could also be distributed over totally different network domains and don't win the trust one another. A secure mechanism has got to confirm that the identity of the shoppers and also the server is authenticated one another and also the communications are secure against associate degree unauthorized user from eavesdropping the delivery contents [1-2, 5]. The client and also the application server need a communal trustworthy third party [3-4, 17]. Password is wide used to construct a secure key exchange protocol since password-based protocols area unit simply to be developed and to be maintained. However, users have to worry concerning whether or not their passwords (have low entropies) are guessed or not. The arcanum idea attack is divided into three kinds [11-12]

1. On-line detectable estimate attack.

Attacker will enumerate all the cause passwords and develop one from the list. Then the aggressor sends the chosen password to attach the server and verifies the server's response in on-line. Most password-based protocols will stop this attack by the server limits the fail times.

2. On-line undetectable estimate attack.

Attacker will enumerate all the drive passwords and obtain one from the list. Then the offender sends the chosen countersign to connect the server and verifies the server's response in on-line. Since the server cannot discriminate whether or not the request is malicious or honest, thus the server continually replies a honest response. The offender will catch this chance to guess the countersign till the password is properly obtained [23].

3. Off-line approximation attack.

Since the communicated channel is open, any eavesdropper can collect all the Communications. Then the wrongdoer can enumerate all the campaigning passwords to launch the attack off-line till a success is obtained without the help of the server. Many password-based three-party key exchange protocols were planned and addressed to overcome the above approximation attacks by victimization the construct of public-key and symmetric-key techniques [10-11, 19-20, 26]. For enhancing the efficiency dramatically, in 2007, Lu and Cao proposed a straightforward three-party key exchange protocol [21] without victimization the server's public key. Unfortunately, Lu-Cao's key exchange protocol suffered from the unknown key sharing1, the on-line undetectable approximation, and the impersonation attacks [12, 15, 18, 23].

For guaranteeing the quality of communication services, low communication and computation price is needed in a three-party key exchange protocol. In 2009, Huang [16] planned AN

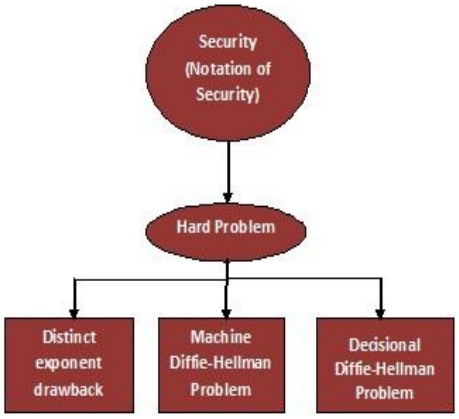
efficiency-enhanced password-based three-party key exchange protocol. Huang claimed that the planned protocol is also more economical than Lu-Cao's protocol and may be applied in follow. However, Huang's protocol is still not secure against the on-line undetectable approximation attack [25].

We propose a provably secure password-based three-party key exchange protocol to withstand numerous well-known security threats by victimization the random oracle model [3, 11, 22]. Compared with the connected protocols [10-11, 20], our proposed protocol is computation-efficient. In the next section, we have a tendency to first give a notation of security. In Section 3, we have a tendency to propose a completely unique three-party key exchange protocol. In Section four, we analyze the protection of the planned protocol. In Section five, we have a tendency to analyze the efficiency among our proposed protocol and the connected protocols. Finally, we have a tendency to conclude this paper in Section half dozen.

An unknown key-sharing attack on a key exchange protocol that provides the key confirmation property is Associate in Nursing attack whereby an entity A believes that she shares a session key with the communicated entity B. Unfortunately, it's undeniable fact that if the entity *B* mistakenly believes that the session key's instead shared with another entity *E*, where $E \neq A$. A secure key exchange protocol ought to be against this threat [6, 8].

2. Notations of Security

We initial outline some onerous mathematical problems and security of a password-based three-party protocol.



2.1 onerous issues

1) Distinct exponent drawback (DLP).
2) Machine Diffie-Hellman Problem (CDHP).
3) Decisional Diffie-Hellman Problem (DDHP).

2.2 Security Definitions

The concrete security of a 3 party-based protocol is made up each the property of the session key sameness and also the protection of the password [7, 22]. During a password-based protocol, AN on-line detectable estimation attack [14] is inherent and is inevitable. However, this attack can be prevented by lockup the account once some affordable unsuccessful tries in most password-based protocols. A additional dangerous attack is that the off-line estimation attack once an adversary copies a transcript of executions during a password-based protocol. The mission of a password-based protocol is to rule out the off-line guessing attack and to limit the somebody solely to the on-line detectable estimation attack. For thwarting the web detectable estimation attack, the service requesters' requests area unit needed to be authenticated for the operations of the sure server from distinctive malicious tries from real requests. Also, for deterring the on-line undetectable and also the off-line estimation attacks, the proposed protocol needs to live up to the requirement of attackers that they will develop the correct parole however cannot verify their estimation from the eavesdropped messages.

We denote the projected protocol, a service requester C_A and a service supplier $C_B \in \hat{C}$ $\{C_1,\ldots..,$ $C_{NC}\}$ and a sure server S. every service requester C_A and a service supplier $C_B \in \hat{C}$ hold memorial passwords $p\omega_A$ and $p\omega_B$, and also the server S maintains a parole table . We also assume that an somebody AD United Nations agency controls all the communications that occur by C_A^i, C_B^j and S could be a probabilistic machine, wherever we tend to denote that C_A^i is that the *i*th instance of the service requester C_A and C_B^j is that the *j*th instance of the service supplier C_B. AD will move with all the participants (C_A, C_B, S) through the subsequent oracle queries.

a) Execute(C_A^i, C_B^j), Execute(C_A^i, S), Execute(C_B^j, S): we have a tendency to use this question to model passive attacks wherever *AN* aggressor will listen all the communications between the instances $(C_A^i, C_B^j)$and between the instances (C_A^i, S),and$(C_A^i,)$respectively.

b) Send Client (C_A^i, m): we have a tendency to use this question to model a lively attack against that the aggressor sends a message *m* to a participant C_A at the *i*th instance. Then question outputs the results of C_A from receiving the message m to come up with.

c) Send Server (*m*): we have a tendency to use this question to model a lively attack against that the aggressor sends a message

m to the server S. Then question outputs the results of S from receiving the message m to generate.

d) Reveal (C_A^i): we have a tendency to use this question to model a lively attack against the known-key attack at the *i*th instance C. The question says that if the instance doesn't settle for the session key, the output is ⊥; otherwise, the output is that the real session key.

e) Corrupt (C_A): we have a tendency to use this question to permit that an aggressor AD will corrupt the whole internal state of an entity C_A.

f) Test (C_A^i): If AN aggressor AD queries this oracle and no session key for $C_A^i \in \hat{C}$ is accepted, this oracle outputs ⊥; otherwise, the oracle flips a coin b. If b = 1, returns the important session key; if b = 0; returns a random key that has identical key with the important session key.

The security definition of the planned protocol depends on the partnership and freshness of oracles, wherever the partnership of the oracles is outlined victimization the session identifiers cot death and therefore the partnership is outlined to limit the adversary's Reveal and Corrupt queries. If the partnership isn't accepted by the oracles, the antagonist is attempting to guess the session key.

- ✓ Partnership: are saying that two oracles C_A^i and C_B^j are partners, if and providing each of the oracles have accepted an equivalent session key with an equivalent session symbol and that they have in agreement on an equivalent set of exchanging messages. Besides C_A^i and C_B^j, no different oracles have accepted with an equivalent session symbol.

- ✓ Freshness: are saying that oracles C_A^i and C_B^j are recent if and providing the oracle C_A^i has accepted another partner oracle C_B^j, the oracle C_B^j has accepted another partner oracle C_A^i, and every one the oracles C_A^i and C_B^j haven't been sent a Reveal question a Corrupt question.

- ✓ Session key security: we have a tendency to use the quality linguistics security notation to model this property [22]. the protection of session secret's outlined that the opposer United Nations agency desires to discriminate a true key from a random one within the game *G* is indistinguishable, wherever the sport compete between the opposer *AD* and a collections of U_x^i oracles. The players $U_x \in \hat{C}$ and *S* and instances $i \in \{1, ..., N_i\}AD$ runs the sport *G* with the subsequent stages

Stage 1: *AD* is allowed to send the queries (Execute, Send Client, Send Server, Reveal and Corrupt) within the game. Throughout the sport *G*, at some purpose, *AD* will opt for a contemporary session and finish a check question to at least one of the contemporary oracles C_A^i and C_B^j for the testing. Reckoning on the unbiased coin *b*, *AD* is given ether the particular session key *K* or a random one from the session key distribution.
Stage 2: *AD* will still send the queries to the oracles Execute, SendClient, SndServer, Reveal and Corrupt for its selection. However, *AD* is restricted to send the Reveal and Corrupt queries to the oracles for its check session. Eventually, *AD* finally ends up the sport simulation and decides to output its guess bit *b'*.

The success of *AD* from breaking the protocol within the game depends on passwords that square measure drawn from a lexicon D and is measured in terms of the advantage of *AD* from identifying whether or not the received price is that the real key or a random one.

Let $Adv_{P,D}^{G,AD}(k, q_{fake\text{-}C})$ be the advantage of *AD* and therefore the advantage operate be be outlined as follows.

$$\boldsymbol{Adv_{P,D}^{G,AD}(k, q_{fake\text{-}C}) = |Pr\ [b'\text{-}b]\text{-}q_{fake\text{-}C}/N - 1/2*(N\text{-}q_{fake\text{-}C})|} \quad \textbf{-(1)}$$

Where *k* may be a security parameter, *N* denotes the scale of the lexicon *D* and $q_{fake\text{-}C}$ denotes the quantity of tries of the someone from faking the shopper. once $q_{fake\text{-}C}$ times of faking the shopper, the intuition of the formulation is that the advantage of the someone from finding the proper positive identification and from faking the session key with success ought to have the likelihood at the most $q_{fake\text{-}C}/N$. the remainder of non-successful faking cases could have the winning likelihood.

2.3. Password protection:

An oppose may try to guess the countersign of a legitimate consumer and verify its guess through the interaction with the server or the consumer or from the intercepted messages. we need that the protocol has got to offer the specific authentication of a client's request for thwarting the online detectable idea attack in which the server can do some actions specified the limitation of invalid request makes an attempt cannot exceed the pre-defined threshold. Security against the opposer from launching the off-line idea and the on-line

undetectable idea attacks, the protocol mustn't offer any advantageous info to outsiders or to a curious partner to verify its guess.

Square measure saying that a password-based tripartite key exchange protocol is secure in our model once the subsequent necessities are satisfied:

1) Validity: Among three oracles (C_A^i, C_B^j, *S*), the oracles (C_A^i, C_B^j) settle for a similar session key within the absence of a lively opposer.

2) Session key indistinguishability: For all probabilistic, the advantage of the opposer *AD* is negligible at intervals a polynomial time.

3) Specific authentication: because the higher than mentioned, the protocol ought to make certain that the specific authentication of communicated parties is completed for being against the net detectable estimate attacks.

4) Parole protection: because the higher than mentioned, the protocol shouldn't offer any advantageous data to outsiders or to a curious partner to verify its guess for being against the off-line estimate and also the undetectable on-line estimate attacks.

3. Projected Protocol

In our protocol, we have a tendency to outline h_1 *()* and h_2 *()* are secure crypto logical unidirectional hash functions and that we can model the unctions as random oracles within the security proof. The opposite parameters ar introduced as follows:

A. The system selects an outsized prime *p*, wherever *(p - 1)* incorporates a divisor letter.

B. Let *g* be a generator with order letter in $GF_{(p)}$.

C. *TS* denotes the trusty third party.

D. *A* and B denote two communicated parties.

E. pw_A and pw_B denote the passwords that *A* shared with *TS* and *B* shared with *TS*, severally.

F. denotes Associate in Nursing exclusive *OR* operation.

G. For simplicity, all the mathematical operation operations are below the standard *p* like g^x *mod p* $\rightarrow g^x$.

- ✓ Request that instigator A selects a random number x, calculates $R_A = g^x \square h_1(pw_A, A, B, sid)$, and sends *(A, sid, R_A)* to the communicator *B*, wherever the sid denotes the session identity.
- ✓ Upon receiving the request, *B* conjointly selects a random range *y*, calculates metallic element $R_B = g^y \square h1\ (pw_B, A, B, sid)$, and sends (*B*, R_B) with *A's* request to the trusty server *TS*.
- ✓ Upon receiving (*A, B, sid,* R_A, R_B), TS employs the passwords pw_A and pw_B to extract the changed data g^x and g^y, severally. Then *T* selects 3 random numbers (z_1, z_2, z_3) and calculates (*a, b, c, d*), where $a = g^{xz1}$, $b = g^{yz1}$, $c = g^{z2}$, and $d = g^{z3}$.
- ✓ TS sends (*A, sid,* Z_{A1}, Z_{A2}) and (*B, sid,* Z_{B1}, Z_{B2}) to A and B in parallel, wherever $Z_{A1} = b \square h_1(pw_A +1, A, B, sid)$, $Z_{A2} = c \square h_1(pw_A +2, A, B, sid)$, $Z_{B1} = a \square h_1(pw_B +1, A, B, sid)$, *and* $Z_{B2} = d \square h1(pw_B +2, A, B, sid)$.

4. Waste parallel

(a) Upon receiving *(B, sid,* Z_{B1}, Z_{B2}*), B* employs $h_1(pw_B +1, A, B, sid)$ and $h_1(pw_B +2, A, B, sid)$ to recover a and d. *B* then calculates the session key $K = h_2(A, B, sid, ay)$, $S_{B1} = h1(A, B, sid, K)$ *and* $S_{B2} = h_1(A, B, sid, d^y, a)$. B sends S_{B1} to *A* and S_{B2} to *TS* for characteristic the validation of its identity and therefore the session key.

(b) Upon receiving *(A, sid,* Z_{A1}, Z_{A2}*), A* employs $h_1(pw_A +1, A, B, sid)$ and $h_1(pw_B +1, A, B, sid)$ to recover *b* and *c*. A then calculates the session key $K = h_2(A, B, sid, b^x)$, $S_{A1} = h_1(A, B, sid, K+1)$ *and* $S_{A2} = h_1(A, B, sid, c^x, b)$.

A sends S_{B1} to *B* and S_{A2} to TS for characteristic the validation of its identity and therefore the session key. each of A and B will attest one another by checking the validation of S_{B1} and S_{A1} and believe that the closely-held session secret's contemporary. Upon receiving *A* and *B*'s responses, *TS* will check the validation of S_{B2} and S_{A2}. If any of the conditions doesn't hold, *TS* can come "connection failure" message to the corresponding parties and increase the fail times by one.

5. Security Analysis

In this section, we have a tendency to analyze that the projected protocol is secure against some well-known attacks. Before our analysis, we have a tendency to 1st assume that the subsequent mathematical issues are arduous to be solved [9, 13].

(a) Though $a = g^{xz1}$ Associate in Nursingd $b = g^{yz1}$ area unit legendary by an individual, supported the problem of the *CDHP*, the individual cannot derive the session key $K = g^{xyz1}$ except the parties *A* and *B*.

(b) Supported the properties of unidirectional hash perform and therefore the exclusive-OR operator, the individual is useless to derive (g^x, b, g^y, a) while not the data of *A* and *B*'s passwords. the explanation is that the extracted values can't be verified. The individual desires to discriminate (g^x, b, g^y, a) from $(R_A, R_B, Z_{A1}, Z_{B1})$, the chance of getting the session key K is resembling solve the *CDHP* on $(Z_{A1}, S_{A1}, Z_{B1}, S_{B1})$.

5.1. Replay Attack.

An soul World Health Organization desires to imitate the requester A will resend the used messages $(R_A = g^x \square h_1(pw_A +1, A, B, sid))$ to B or to TS and expect to get some helpful data from *TS* like $(Z_{A1} = g^{yz1} \square h_1(pw_A +1, A, B, sid), Z_{A2} = g^{z_2} \square h_1(pw_A +1, A, B, sid))$. Supported the *CDHP* assumption, the soul not solely cannot derive new session key $K = g^{xyz1}$ while not the data of the temporary keys x, however additionally cannot win the trust of TS while not the data of the arcanums pw_A since g^{z_2} is encrypted victimisation the password pw_A.

5.2. Impersonation Attack.

In spherical three of our planned protocol, once somebody sends the changed messages to TS, TS continually returns the messages $(Z_{A1}, Z_{A2}, Z_{B1}, Z_{B2})$ back. The soul will catch this opportunity to launch the attack. Note that TS waits the responses in spherical four. Since all the changed messages should be encrypted victimisation the arcanum severally, the soul cannot grasp whether or not the guessed arcanum is correct or not and additionally cannot decide whether or not the received message S_{B1} and therefore the computed results (S_{A1}, S_{A2}) area unit correct or not. supported the troublesome of the *CDHP*, this manner is blocked.

5.3. Arcanum guesswork Attack.

On-line detectable guesswork attack. In current systems, there's a regular mechanism to defeat this attack. the answer is that the remote server logs and counts the quantity of trial failures. If the quantity is larger than the pre-defined threshold values, the server stops the affiliation. this idea is applied to our protocol since TS verifies whether or not A and B's responses (S_{A2}, S_{B2}) area unit correct or not in spherical four and records the failure times. On-line undetectable guesswork attack. To launch the attack with success, the wrongdoer should get some helpful data prior to for manipulating the info and collateral their guess on TS's response (or B's response). The attack cannot work on our protocol since all the requests need to be sent to TS and TS can wait the feedbacks from both of A and B. It implies that any trial method are going to be detected by TS. The attack fails. Off-line guesswork attack. The entire changed messages area unit encrypted victimisation the passwords severally. The goal of the soul is to guess the arcanum and to verify the correctness on the intercepted messages. Supported the difficult of the CDHP, the soul cannot use the guessed arcanum and derive messages to get any results on the messages $(S_{A1}, S_{A2}, S_{B1}, S_{B2})$ in spherical four.

5.4. Forward/Backward Secrecy.

In every session, *A, B* and *TS* choose their temporary keys (x, y, z_1, z_2) to construct $(R_A = g^x \square h_1(pw_A +1, A, B, sid)$, metallic element $R_A = g_y \square h_1(pw_B +1, A, B, sid)$, $z_{A1} = b \square h_1(pw_A +1, A, B, sid)$, $z_{B1} = a \square h_1(pw_B +1, A, B, sid))$. Supported the troublesome of the CDHP, the soul cannot calculate the session key $K = h_2(A, B, sid, g^{xyz1})$ altogether the sessions even if the passwords area unit guessed properly. The property of the forward secrecy is provided. notwithstanding one in every of the used session key $K = h_2(A, B, sid, g^{xyz1})$ is compromised by the soul, the soul cannot acquire any helpful data on the corresponding messages. for example, the soul might guess the arcanum to get $g^{x'}$ and $g^{yz1'}$. Supported the troublesome of the CDHP, the soul cannot verify the guessed arcanum. Because the higher than mentioned, while not the data of the arcanum, the soul cannot launch any attacks. Hence, the backward secrecy is additionally unbroken in our protocol.

6. Potency Analysis

In this section, we have a tendency to analyze the computation price of a service requester as a result of the requester may use personal mobile devices to get the fascinating services. Also, as introduced in [24], we will learn a relationship as follows: the time of one modular operation is quicker 5/3 times than the time of 1 public-key en/decryption operation, the time of 1 standard multiplication computation is quicker 240 times than the time of 1 standard operation operation, and the time of 1 unidirectional hash operate operation is quicker 600 times than the time of 1 standard operation.

In A calculates $R_A = g^x \square h_1 (pw_A +1, A, B, sid)$. The value is one standard operation and one hash operates operation.

$b = Z_{A1} \square h_1 (pw_A +1, A, B, sid)$ and

$c = Z_{A2} \square h_1 (pw_A +1, A, B, sid)$.

The value is hash function operations. Then A calculates the session key

$K = h_2 (A, B, sid, b^x)$

$S_{A1} = h_1 (A, B, sid, K+1)$ and

$S_{A2} = h_1 (A, B, sid, c^x, a)$.

The value is standard operation and four hashes operate operations. By the on top of, the computation price of A is three standard exponentiations and hash operate operations.

Within the communication price, we have a tendency to denote that:

Message Step denotes that one entity has sent knowledge to the communicated party. Communication spherical means if the sent knowledge are freelance between every message steps, one or additional message steps will be integrated into an equivalent communication spherical because of the sent knowledge will be performed in parallel. The burden of the communication price will be reduced.

We have a tendency to summarize the ends up in Table one and that we will see that our protocol is additional economical than the connected protocols [10-11, 16, 20-21].

Table: Comparisons of the Computation Cost At Requester Side and the Communication Cost

Operation	Our	Lu-Cao	Huang	Chien-Wu	Chen-et-al	Lo-Yeh
T_{EXP}	3	4	2	2	3	3
T_{MUL}	0	2	0	0	0	0
T_H	7	3	4	4	4	4
T_{PKC}	0	0	0	1	1	1
T_{SYM}	0	0	0	0	1	1
Total (T_{MUL})	722.8	963.2	481.6	881.6	1121.6 +1 T_{SYM}	1121.6 +1 T_{SYM}
Rounds	4/8	5/5	5/5	4/4	5/5	4/6

- ✓ **TEXP** → modular exponentiation operations.
- ✓ **TMUL** → modular multiplication computation.
- ✓ **TH** → hash function operation.
- ✓ **TPKC** →public key endecryption operation.
- ✓ **TSYM** → symmetric key endecryption operation

7. Conclusions

In this paper, we've got planned a demonstrably secure password-based multilateral key exchange protocol to beat some standard security threats. Compared with the connected protocols, the computation potency continues to be unbroken in our planned protocol.

References

1. M. Abdalla, E. Bresson, O. Chevassut, B. MÖoller, D. Pointcheval, Strong Password-Based Authentication in TLS using the Three-Party Group Diffie-Hellman Protocol, International Journal of Security and Networks. 2007,
2. (3/4):284-296. 2. M. Abdalla, D. Catalano, C. Chevalier, D. Pointcheval, Efficient Two-Party Password-Based Key Exchange Protocols in the UC Framework, in: Topics in Cryptology - CT-RSA 2008, LNCS 4964, 2008, 335-351.
3. M. Abdalla, P.-A. Fouque, D. Pointcheval, Password-Based Authenticated Key Exchange in the Three-Party Setting, in: Public Key Cryptography - PKC 2005, LNCS 3386, 2005, 65-84.
4. M. Abdalla, P.-A. Fouque, D. Pointcheval, Password-Based Authenticated Key Exchange in the Three-Party Setting, IEE Proceedings, 2006, 153 (1):27-39.
5. M. Abdalla, D. Pointcheval, A Scalable Password-based Group Key Exchange Protocol in the Standard Model, in: Advances in Cryptology – ASIACRYPT 2006, LNCS 4284, 2006, 332-347.
6. J. Baek, K. Kim, Remarks on the unknown key-share attacks, IEICE Trans. on Fundamentals, 2000, E83-A(12):2766-2769.
7. M. Bellare, P. Rogaway, Provably secure session key distribution the three party case, in: Proc. of the 27th ACM Annual Symposium on the Theory of Computing, 1995, 57-66.
8. S. Blake-Wilson, A. Menezes, Unknown key-share attacks on the stationto-station (STS) protocol, in: Public Key Cryptography (PKC '99) Proceedings, LNCS 1560, 1999, 154-170.
9. S. Boneh, B. Lynn, H. Shacham, Short Signatures from theWeil Pairing, in: Advances in Cryptology – ASIACRYPT 2001, LNCS 2284, 2001, 514-532.
10. H.-B. Chen, T.-H. Chen, W.-B. Lee, C.-C. Chang, Security enhancement for a three-party encrypted key exchange protocol against undetectable on-line password guessing attacks, Computer Standards & Interfaces, 2008, 30(1-2):95-99.
11. H.-Y. Chien, T.-C. Wu, Provably secure password-based three-party key exchange with optimal message steps, The Computer Journal, 2009, 52(6):646-655.
12. H.-R. Chung, W.-C. Ku, Three weaknesses in a simple three-party key exchange protocol, Information Sciences, 2008, 178(1):220-229.
13. W. Diffiee, M. Hellman, New directions in cryptology, IEEE Transations on Information Theory, 1976, IT-22(6):644-654.
14. Y. Ding, P. Horster, Undetected on-line password guessing attacks, ACM Operating Systems Review, 1995, 29(4):77-86.
15. Guo, Z. Li, Y. Mu, X. Zhang, Cryptanalysis of simple three-party key exchange protocol, Computers & Security, 2008, 27(1-2):16-21.
16. H.-F. Huang, A simple three-party password-based key exchange protocol, International Journal of Communication Systems, 2009, 22(7):857-862.
17. W.-S. Juang, Efficient three-party key exchange using smart cards, IEEE Trans. On Consumer Electronics, 2004, 50(2):619-624.

18. H.-S. Kim, J.-Y. Choi, Enhanced password-based simple three-party key exchange protocol, Computers & Electrical Engineering, 2009, 35(1):107{114.
19. T.-F. Lee, J.-L. Liu, M.-J. Sung, S.-B. Yang, C.-M. Chen, Communication-efficient three-party protocols for authentication and key agreement, Computers & Mathematics with Applications, 2009, 58(4):641-648.
20. N.-W. Lo, K.-H. Yeh, Cryptanalysis of two three-party encrypted key exchange protocols, Computer Standards & Interfaces, 2009. 31(6):1167-1174.
21. R. Lu, Z. Cao, Simple three-party key exchange protocol, Computers & Security, 2007, 26(1):94-97.
22. D. P. M. Bellare, P. Rogaway, Authenticated and key exchange secure against dictionary attacks, Advances in Cryptology - EUROCRYPT 2000, LNCS 1807, 2000, 139-155.
23. R. C.-W. Phan, W.-C. Yau, B.-M. Goi, Cryptanalysis of simple threeparty key exchange protocol (S-3PAKE), Information Sciences, 2008, 178(13):2849-2856.
24. B. Schneier, Applied cryptography, 2nd edition, John Wiley & Sons Inc., 1996.
25. S. Wu, Weakness of a three-party password-based authenticated key exchange protocol, Report 2009/535, CryptEAr (Nov. 2009). URLhttp://eprint.iacr.org/2009/535.pdf
26. E.-J. Yoon, K.-Y. Yoo, Improving the novel three-party encrypted key exchange protocol, Computer Standards & Interfaces, 2008, 30(5):309-314.

