An Improvement on a Three-party Password-based Key Exchange Protocol Using Weil Pairing

Zeng Yong[1], Ma Jianfeng[1], and Sangjae Moon[2]
(Corresponding author: Zeng Yong)
Key Lab of Computer Networks and Information Security, Ministry of Education, Xidian University
No.2, South Taibai Road, Xi'an 710071, P.R. China[1]
Mobile Network Security Technology Research Center, Kyungpook National University, Korea[2]
(Email: {yongzeng, ejfma}@hotmail.com, sjmoon@ee.knu.ac.kr)
(Received May 11, 2006; revised Sep. 19, 2006; and accepted May. 2, 2007)

Abstract

The three-party password-based key exchange protocols using Weil pairing proposed by Wen is vulnerable to impersonation attack. By introducing hard artificial intelligence problem, we show an improved protocol, which can resist against not only the impersonation attack but also all the other well-known attacks. Analysis also shows that improved protocol reduces about one third computational cost and two thirds throughput. The protocol is suitable for lightweight or mobile equipments.

Keywords: Hard artificial intelligence problem, key exchange protocol, password-based authentication, weil pairing

1 Introduction

Password-based authenticated key schemes have a wide range of applications [7, 9, 15, 17], especially the consumers who have no device capable of securely storing high entropy secret keys. In general, such schemes require that there is a shared human memorable password between the users and the server machine. The password is always weak (low-entropy). So it requires carefully bootstrapping from a weak shared password to a strong one.

To overcome above problem, a password-based scheme was firstly proposed in [11]. Many password-based systems are vulnerable to dictionary attack [2]. Since EKE (the encrypted key exchange) protocol against this attack was proposed in [2], a lot of 2-PAKE (two-party password-based authenticated key exchange schemes) have been proposed. 2-PAKE protocols are suitable for client-server architectures. However, it is very inconvenient in key management since it require every pair of participants to share a password. As a result, the first 3-PAKE (three-party password-based authenticated key exchange protocols) have been proposed in [8] to overcome the inconvenience, which allows any two participants to authenticate mutually through a trusted server. Since then the 3-PAKE protocols have received much attention. Recently, the first Weil paring based 3-PAKE was proposed in [16], which is shown vulnerable to impersonation attacks in [12]. However, no improvement is proposed yet.

On the other hand, CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart), a hard AI (Artificial Intelligence) problem proposed by [1], has been widely used in many internet company to prevent free accounts registration by machine alone. CAPTCHA is also used to design a password-only authentication scheme in [10, 13]. However it is shown they suffer off-line dictionary attacks in [14].

As to the best our knowledge, the human being's special abilities are not carefully considered to improve the efficiency and security of most known 3-PAKE protocols. We will first introduce CAPTCHA to improve the Weil paring based 3-PAKE. The improved protocol is shown that it can resist against all the well-known attacks. Furthermore, participation of human beings in the protocol reduces the computing complexity of consumers' equipments largely. The analysis shows that improved protocol need only about two thirds computing amounts and one third throughput (receiving and transmitting data amounts) of the old one in the users' equipments, respectively. So the improved protocol is well suited for lightweight or mobile consumer equipments.

2 Review

This section reviews some basic assumptions firstly, then the Weil paring based 3-PAKE protocol [16], finally the impersonation attack [12].

2.1 Preliminaries

Bilinear Paring. Let G_1 be an additive group of prime order q and G_2 a cyclic multiplicative group of the same order q. The discrete logarithm problems (DLP) in both G_1 and G_2 are assumed to be hard. Let P be a generator of G_1. $H : \{0,1\}^* \rightarrow Z_q$ be one cryptographic hash functions, and $G : \{0,1\}^* \rightarrow G_1$ be the cryptographic one-way hash function that maps a string to a point of G_1 [12]. $e : G_1 \times G_1 \rightarrow G_2$ be a bilinear mapping satisfying the following conditions.

1) Bilinear: Let $a, b \in Z$ and $P, Q \in G_1, e(aP, bQ) = e(P, Q)^{ab}$.
2) Non-degenerate: There exists $P \in G_1$ such that $e(P, P) \neq 1 \in G_2$.
3) Polynomial-time computable: The mapping function $e(P, Q)$ is computable in polynomial time.

Bilinear Diffie-Hellman Problem (BDHP). For a bilinear paring $e : G_1 \times G_1 \rightarrow G_2$ is defined as follows: Given $cP, bP, aP, P \in G_1$, compute $e(P, P)^{abc}$, where a, b, c are random numbers from Z_q^*. BDHP is a variant of the Computational Diffie-Hellman (CDH) problem. It is also called the Weil Diffie-Hellman (WDH) problem. It is commonly believed that the BDHP problem is hard. How to correctly choose G_1, G_2 and $e : G_1 \times G_1 \rightarrow G_2$ in the real world applications can be seen in [3].

2.2 The Protocol

Setup. Let $(G_1, G_2, H, G, e(), E()/D())$ be the public system parameters, where $E()$ denotes an ideal symmetric encryption function and $D()$ denotes the corresponding decryption function. $ID_S/ID_A/ID_B$ respectively denotes the identity of the authentication server S/user A/user B. The server S owns its secret key s and publicizes its public key $P_s = sP$. The users A and B share passwords PW_A and PW_B with the server S, respectively.

Execution. To share an authenticated session key, the server S, the users A and B perform the following steps. "$A \rightarrow B : M''$ denotes that A sends the message M to B. The following is the protocol.

1) $A \rightarrow B$: (ID_A, aP, c_a). User A selects a random number a, computes aP and $k_a = H(aP, P_S, Q, e(P_S, aQ))$, where $Q = G(ID_S)$. Then A computes $c_a = E_{ka}(PW_A)$ and sends (ID_A, aP, c_a) to user$B$.
2) $B \rightarrow S(ID_A, aP, c_a, ID_B, bP, c_b, \mu_b)$. User B selects a random number b, computes bP, $k_b = H(bP, P_S, Q, e(P_S, bQ))$ and $K = e(aP, bU)$, where $U = G(ID_A, ID_B)$. Then B computes $c_b = E_{kb}(PW_B)$, $\mu_b = H(ID_B, K)$ and sends $(ID_A, aP, c_a, ID_B, bP, c_b, \mu_b)$ to server S.
3) $S \rightarrow A$: $(ID_B, bP, \mu_b, \sigma_b, \sigma_a)$. S computes $k_a = H(aP, P_S, Q, e(aP, sQ))$, $k_b = H(bP, P_S, Q, e(bP_S, sQ))$, and verifies the equality $PW_A = D_{ka}(c_a)$ and $PW_B = D_{kb}(c_b)$, respectively. If any one of the verifications fails, S rejects the session; otherwise, S computes $\sigma_a = H(kb, aP)$ and $\sigma_b = H(k_a, bP)$, and sends $(ID_B, bP, \mu_b, \sigma_b, \sigma_a)$ to user A.
4) $A \rightarrow B$: (μ_a, σ_a). A computes $K = e(bP, aU)$ and checks the equality of $\sigma_b = H(k_a, bP)$ and $\mu_b = H(ID_B, K)$, respectively. If any one of the verifications fails, A rejects the session. Otherwise, A computes $\mu_a = H(ID_A, K)$ and sends (μ_a, σ_a) to B.
5) Upon receiving the data in Step 4, B verifies the equality $\sigma_a = H(k_b, aP)$ and $\mu_a = H(ID_A, K)$, respectively. If any verification fails, B rejects the session; otherwise, B accepts and completes the session.

The final session key shared between A and B is $SK = H(aP, bP, U, K)$.

2.3 Impersonation Attack

Say user W who owns his identity ID_W and shares his password PW_W with the server S. Let "$B(W)$" denote that W impersonate B to send message. Let "$A \rightarrow W! \rightarrow B''$ denote that A sends messages to B are intercepted by W.

1) $A \rightarrow W! \rightarrow B(ID_A, aP, c_a)$. User A sends (ID_A, aP, c_a) to user B, but it is intercepted by W.
2) $W \rightarrow S(ID_A, aP, c_a, ID_W, eP, cw, \mu_w)$. Adversary W selects a random number w, computes wP, $k_w = H(wP, P_S, Q, e(P_S, wQ))$ and $K = e(aP, wU)$, where $U = G(ID_A, ID_B)$. Then W computes $c_w = E_{kw}(PW_W)$, $\mu_w = H(ID_B, K)$ and sends $(ID_A, aP, c_a, ID_W, wP, c_w, \mu_w)$ to server S.
3) $S \rightarrow W! \rightarrow A$: $(ID_W, wP, \mu_w, \sigma_b, \sigma_a)$. S computes $k_a = H(aP, P_S, Q, e(aP, sQ))$, $k_w = H(wP, P_S, Q, e(wP_S, sQ))$, and verifies the equality $PW_A = D_{ka}(c_a)$ and $PW_W = D_{kw}(c_w)$, respectively. If any one of the verifications fails, S rejects the session; otherwise, S computes $\sigma_a = H(k_w, aP)$ and $\sigma_b = H(k_a, wP)$, and sends $(ID_W, wP, \mu_w, \sigma_b, \sigma_a)$ to user A, but this message is intercepted by W.
4) $S(E) \rightarrow A$: $(ID_B, wP, \mu_w, \sigma_b, \sigma_a)$. After intercepting the message in Step 3, W replaces the identity ID_W with ID_B, and impersonates S to send the message $(ID_B, wP, \mu_w, \sigma_b, \sigma_a)$.
5) $A \rightarrow W! \rightarrow B$: (μ_a, σ_a). A computes $K = e(wP, aU)$ and checks the equality of $\sigma_b = H(k_a, wP)$ and $\mu_w = H(ID_B, K)$, respectively. Since both the verifications succeed, A wrongly believes she is communicating with B, then A computes $\mu_a = H(ID_A, K)$

and sends (μ_a, σ_a) to B. This message is also intercepted by W.

Finally, A wrongly believes she is communicating with B, but A un-intentionally shares the session key $SK = H(aP, wP, U, K)$ with W.

3 Improved Protocol

Denote $\varphi(r, t)$ as a distorted picture function (our scheme uses CAPTCHA), where $r \in \psi_n$, ψ is the set of all the 52 upper-case and lower-case letters and 10 digits, ψ_n is the set of all n length strings of symbols in ψ, and t is a random number to generate a distorted picture of r such that humans have the ability φ^{-1} to recognize r from the distorted picture but machines have not. And different t can generate different distorted pictures of r to machine while humans the same. To make the offline dictionary attack computationally infeasible, the size of ψ_n has to be $62^{14} > 2^{80}$, so that the length of the string should be larger than 14, $n \geq 14$. The improved protocol is as follows:

Setup. This phase is the same as the old scheme. And the trusted server S shares $\varphi(r, t)$ with all the users.

Execution. To share an authenticated session key, the server S, the users A and B perform the following steps. "$A \rightarrow B : M$" denotes that A sends the message M to B.

1) $A \rightarrow B$: (ID_A, c_a). User A selects a random number a, computes aP and $c_a = E_{pwa}(aP)$. Then A sends (ID_A, c_a) to user B.

2) $B \rightarrow S(ID_A, ID_B, c_a, c_b)$. User B selects a random number b, computes bP and $c_b = Epwb(bP)$. Then B sends (ID_A, ID_B, c_a, c_b) to server S.

3) $S \rightarrow B$: $(ID_A, M_1, M_2, M_3, M_4)$. S obtains aP and bP by decrypting $Epwa(aP)$ and $Epwb(bP)$. S randomly chooses number s_1, s_2 and computes $k_a = e(aP, s_1Q)$ and $k_b = e(bP, s_2Q)$, $Q = G(ID_S)$. Then S selects a string r ($r \in \psi_n$, the length of r is larger than 14), two random numbers t_a and t_b. S computes $M_1 = E_{ka||pwa}(\varphi(r, t_a))$, $M_2 = E_{pwa}(s_1P)$, $M_3 = E_{kb||pwb}(\varphi(r, t_b))$, $M_4 = E_{pwb}(s_2P)$. Then S sends $(ID_A, M_1, M_2, M_3, M_4)$ to user B.

4) $B \rightarrow A$: (M_1, M_2, M_5). B obtains s_2P by decrypting $M_4 = E_{pwb}(s_2P)$ and computes $k_b = e(s_2P, bQ)$. Then B gets $\varphi(r, t_b)$ by decrypting $M_3 = E_{k_b||pwb}((r, t_b))$. B has the ability φ^{-1} to recover r. If r is not recognizable, the protocol terminates. Otherwise B computes $M_5 = H(1||r||ID_B||ID_A)$. B sends (M_1, M_2, M_5) to user A.

5) $A \rightarrow B$: (M_6). A obtains s_1P by decrypting $M_2 = E_{pwa}(s_1P)$ and computes $k_a = e(s_1P, aQ)$. Then A gets $\varphi(r, t_a)$ by decrypting $M_1 = E_{k_a||pwa}(\varphi(r, t_a))$. A also has the ability φ^{-1} to recognize r. And A can verify M_5 by using r. If the verification fails, the protocol terminates. Otherwise A computes $M6 = H(1||r||ID_A||ID_B)$. The session key $sk = H(2||r||ID_A||ID_B)$. A sends (M_6) to user B.

6) When B receives (M_6) from A, B firstly verifies (M_6) by using r. If it is true, B also computes the session key $sk = H(2||r||ID_A||ID_B)$. Otherwise, A's request is rejected.

The final session key shared between A and B is $sk = H(2||r||ID_A||ID_B)$.

4 Security Analysis

In this section, some attacks and models are introduced firstly. Then the analysis shows that the improved protocol can resist against all the well-known attacks.

4.1 Attacks in the 3-party Scenario

Our 3-PAKE protocol uses a shared password between a client and a server (Shared password authentication, SPA, for short). SPA is the most used one. We discuss the security in SPA model. The following is the notions of security and attacks under SPA listed in [4].

- Perfect Forward Secrecy. A protocol is said to have perfect forward secrecy if compromise of a shared password does not compromise past session keys.
- Denning-Sacco Attack [6]. Compromise of a common session key allows an attacker to mount a dictionary attack on the long-term shared password or to impersonate one of the parties. There are two types, insider attack and outsider attack. Insider adversary is a legal user of the system while outsider adversary is not.
- On-line Password Guessing Attack. An attacker guesses a password on-line. By using the response from the honest client or the server, he verifies the correctness of his guess.
- Off-line Password Guessing Attack. An attacker uses the eavesdropped information to guess a password and verifies his guess off-line. The honest client and the server will not participate.
- Impersonation Attack. An adversary impersonates a client A to communicate with client B. Section 2 gives an illustration.

  Let $\varepsilon(k)$ be a negligible function. An attacker is a probabilistic polynomial time machine

Attacker($1^k, m$) where 1^k is security parameter and m is the useful information to attacker. Let $Adv^*(k)$ be the advantage of attacker in some kind of attack, where "*" denotes the name of the attack. If the advantage is negligible, or the attack is computationally infeasible, we say $Adv^*(k) < \varepsilon(k)$. The attacker is infeasible to solve the BDHP and CDH.

4.2 Security Analysis

- On-line Password Guessing Attack: We have two cases.

  1) Outsider Adversary. The password PW_A of user A is used only in $c_a = E_{pwa}(aP)$, $M_2 = E_{pwa}(s_1P)$ and $M_1 = E_{ka||pwa}(\varphi(r, t_a))$, where $k_a = e(aP, s_1Q)$. An outsider attacker has to guess candidate passwords to decrypt c_a and M_2 to find k_a, which is the only helpful number to verify his guess. However it is obviously computationally infeasible. Furthermore, get $k_a = e(aP, s_1Q)$ from aP and s_1P is a BDHP. A similar analysis exists for the password PW_A of user B. The random number r is selected over ψ_n by server, which has a enough entropy to resist password guessing attack. So $Adv^{on}(k)$ is negligible.

  2) Insider Adversary. Suppose user B is an insider attacker. He can impersonate as user A. B guesses $PW_A^{'}$, selects a random number a and compute $c_a^{'} = E_{pwa'}(a^{'}P)$, and then sends $(ID_A, ID_B, c_a^{'}, c_b)$ to server S. When he receives message, he decrypts M_1 and M_3, then gets $\varphi^{'}(r, t_a)$ and r respectively. Thus he can compare the recognized $r^{'}$ from $\varphi^{'}(r, t_a)$ with r to verify his guess. However, this is a hard AI problem which cannot be solved by machine only. The random number r has a enough entropy to resist password guessing attack. Once human participates to recognize them, he has to take months to find correct password. So the advantage is still negligible.

  By Cases 1, 2 we concludes $Adv^{on}(k) < \varepsilon(k)$.

- Off-line Password Guessing Attack. For the same reason an off-line password guessing attack need to solve BDHP and hard AI problem. So the protocol is immune to off-line password guessing attack.

- Perfect Forward Secrecy. Let $Adv^{PFS}(k)$ be the advantage of attacker in attacking perfect forward secrecy. Then we show $Adv^{PFS}(k) < \varepsilon(k)$. Assume that an attacker knows PW_A and PW_B. Then he can obtain aP and bP by decrypting c_a and c_b. But Attacker(1^k, PW_A, PW_B, aP,bP) still cannot determine a and b ($a, b \in Z$) because of CDH. Hence, he cannot decrypt M_1 and M_3 because he has to enumerate a and b to obtain k_a and k_b. So we have $Adv^{PFS}(k) \leq 1/|Z| < \varepsilon(k)$.

- Denning-Sacco Attack. Let $Adv^{DSA}(k)$ be the advantage of attacker in Denning-Sacco attack. We have two cases to be analyzed.

  1) Outsider Adversary Attacker (1^k, sk, ID_A, ID_B, c_a, c_b, M_1, M_2, M_3, M_4, M_5, M_6). Attacker can obtain sk, ID_A, ID_B, c_a, c_b, M_1, M_2, M_3, M_4, M_5 and M_6). But these values cannot help him to compromise PW_A and PW_B, because attacker has to get aP and bP (or s_1P and s_2P) to verify his guessed password. According to the analysis of on-line and off-line password guessing attack, aP and bP (or s_1P and s_2P) are bounded by the probability of finding a and b (or s_1 and s_2) in the Z. Thus $Adv^{PFS}(k) \leq 1/|Z| < \varepsilon(k)$.

  2) Insider Adversary Attacker(1^k,PW_A, sk, ID_A, ID_B, c_a, c_b, r, s_1P, M_3, M_4). Assume user A is a malicious attacker. So he knows PW_A, sk, ID_A, ID_B, c_a, c_b, r, s_1P, M_3,M_4). We will show that attacker cannot mount a dictionary attack on PW_B. First user A cannot get bP and s_2P by decrypting $c_b = E_{pwb}(bP)$ and $M_4 = E_{pwb}(s_2P)$ without PW_B. $M_3 = E_{k_b||pwb}((r, t_b))$, so similar to on-line password guessing attack, the hard AI problem and the random number tb can prevent A from mounting a dictionary attack against PW_B. So $Adv^{DSA}(k)$ is negligible.

  By Cases 1, 2 we concludes $Adv^{DSA}(k) < \varepsilon(k)$.

- Impersonation Attack. From above analysis we know that an attack cannot simply replace some information to obtain users' passwords or the session key. He has to solve the CDH problem and hard AI problem. Then this kind of attack is also computationally infeasible.

To summarize, the improved protocol can resist against all the well-known attacks.

Table 1: Results of computational cost comparison

	Wen's protocol	Improved protocol
Symmetric operations	2	6
Hash operations	14	6
Point multiplications	6	6
Weil pairing	4	2

5 Efficiency Comparison

The computational cost of the two protocols is compared in this section. The results of computational cost comparison are listed in Table 1. The symmetric operations mean the operations of symmetric cryptographic function $E()/D()$ (an ideal symmetric encryption/decryption function) operations. The hash operations mean the operations of cryptographic hash functions $H : \{0,1\}^* \rightarrow Z_q$ and $G : \{0,1\}^* \rightarrow G_1$. The point multiplications mean the operations point multiplication such as aP, $a \in Z$ and $P \in G_1$. The Weil pairing means the operations of a mapping $e : G_1 \times G_1 \rightarrow G_2$.

Generally speaking [5], the computational cost of a symmetric operation is the same to that of a hash operation. The computational cost of a point multiplication is about 10 times than that of a hash operation. The computational cost of a Weil pairing operation is about three times than that of a point multiplication.

According to Table 1, we can see that improved protocols has the same point multiplication operations with that of Wen's one, and reduces four symmetric operations and two Weil pairing operations, respectively. If all the operations are shown as symmetric ones, then Wen's protocol has about 196 symmetric operations and improved one 132. As a result, the improved protocol has about $132/196(\approx 2/3)$ computational cost of Wen's protocol. To summarize, improved protocol reduces one third computational cost.

The throughput of the two protocols is compared. For the Wen's protocol, user A need transmit $(ID_A, aP, c_a, \mu_a, \sigma_a)$ and receive $(ID_B, bP, \mu_b, \sigma_b, \sigma_a)$, and user B need transmit $(ID_A, aP, c_a, ID_B, bP, c_b, \mu_b)$ and receive $(ID_A, aP, c_a, \mu_a, \sigma_a)$. That means users A and B have the throughput of five IDs, five elements over G_1, eight hash values, and four symmetric encrypted values. For simplicity, we denote them as:

$$(5 - ID, 5 - P, 8 - H, 4 - E).$$

For the improved protocol, user A need transmit (ID_A, c_a, M_6) and receive $(ID_B, M_1, M_2, M_5, \sigma_b, \sigma_a)$, and user B need transmit $(ID_A, c_a, c_b, ID_B, M_1, M_2, M_5)$ and receive $(M_1, M_2, M_3, M_4, ID_A, M_6)$. That means users A and B have the throughput of four ID_s, two hash values, and four symmetric encrypted values. For simplicity, we denote them as:

$$(5 - ID, 2 - H, 4 - E).$$

The analysis shows that five elements over G_1 and six hash values transitions are reduced. Because the lengths of P, H and E equal, whereas that of ID is about fifth of theirs, the throughput of improved protocol is 7/18 ($\approx 1/3$) of Wen's. To summarize, the improved protocol reduces about two thirds throughput.

By the above analysis, we can conclude that improved protocol reduces about one third computational cost and two thirds throughput than Wen's.

Acknowledgments

The authors thank for anonymous reviewers and Editor-in-Chief's help.The work is supported by National Natural Science Foundations of China (90204012, 60633020, 60503012), Hi-Tech Research and Development Program of China (2007AA01Z429, 2007AA01Z405).

References

[1] L. V. Ahn, M. Blum, N. J. Hopper, and J. Langford, "CAPTCHA: using hard AI problems for security," *Advances in Cryptology - Eurocrypt' 03*, LNCS 2656, pp. 294-311, Springer-Verlag, 2003.

[2] S. M. Bellovin, and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," *Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 72-84, 1992.

[3] D. Boneh, and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586-615, 2003.

[4] J. W. Byun, I.R. Jeong, D. H. Lee, and C. S. Park, "Password-authenticated key exchange between clients with different passwords," *The Fourth International Conference on Information and Communication Security (ICICS 2002)*, LNCS 2513, pp. 134-146, Springer-Verlag, 2002.

[5] W. Dai, *Speed Benchmarks for Some of the most Commonly Used Cryptographic Algorithms.* (http://www.eskimo.com/weidai/benchmarks.html)

[6] D. Denning, G. Sacco, "Timestamps in key distribution protocols," *Communications of the ACM*, vol. 24, no. 8, pp. 533-536, 1981.

[7] R. Dutta and R. Barua, "Password-based encrypted group key agreement," *International Journal of Network Security*, vol. 3, no. 1, pp. 23-34, 2006.

[8] L. Gong, M. Lomas, R. Needham, and J. Saltzer, "Protecting poorly chosen secrets from guessing attacks," *Journal on Selected Areas in Communications*, vol. 11, no. 5, pp. 648-656, 1993.

[9] M. S. Hwang and C. Y. Liu, "Authenticated encryption schemes: Current status and key issues," *International Journal of Network Security*, vol. 1, no. 2, pp. 61-73, 2005.

[10] C. S. Laih, L. Ding, and Y. M. Huang, "Password-only authenticated key establishment protocol without public key cryptography," *Electronics Letters*, vol. 41, no. 4, pp. 185-186, 2005.

[11] T. Lomas, L. Gong, J. Saltzer, and R. Needham. "Reducing risks from poorly chosen keys," *ACM SIGOPS Operating System Review*, vol. 23, no. 5, pp. 14-18, 1989.

[12] J. Nam, S. Kim, and D. Won, "Security weakness in a three-party password-based key exchange protocol using weil pairing," *Cryptology ePrint Archive*, 2005. (http://eprint.iacr.org/2005/269.pdf)

[13] B. Pinkas, and T. Sander, "Securing passwords against dictionary attacks," *Proceedings ACM Computer and Security Conference (CCS)*, pp. 161-170, 2002.

[14] Q. Tang, and C.J. Mitchell, "Enhanced password-based key establishment protocol," *Cryptology ePrint Archive*, 2005. (http://eprint.iacr.org/2005/141.pdf)

[15] C. S. Tsai, C. C. Lee, and M. S. Hwang, "Password authentication schemes: Current status and key issues," *International Journal of Network Security*, vol. 3, no. 2, pp. 101-115, 2006.

[16] H. A. Wen, T. F. Lee, and T. Hwang, "Provably secure three-party password based authenticated key exchange protocol using Weil pairing," *IEE Proc-Commun*, vol. 152, no. 2, pp. 138-143, 2005.

[17] S, Wu and Y, Zhu, "Proof of forward security for password-based authenticated key exchange," *International Journal of Network Security*, vol. 7, no. 3, pp. 335-341, 2008.

Zeng Yong received his B.S,M.S.,and Ph.D degrees from Xidian University(Xi'an), in 2000, 2003,and 2008 respectively. Since 1995 he has been with Xidian University as a lecturer. His research fields were the nature-inspired techniques applied to combinatorial optimization problems.

MA Jianfeng received his B. S. degree from Shaaxi Normal University (Xian) in 1985, and obtained his M. E. and Ph. D. degrees from Xidian University (Xi an) in 1988 and 1995 respectively. Since 1995 he has been with Xidian University as a lecturer, associate professor and professor. His research interests include information security, coding theory and cryptography.

Sangjae Moon received the B.E. (1972) and M.E. (1974) degrees from Seoul National University, Korea, and the PhD (1984) degree from University of California, Los Angeles. He is currently a professor in the School of Electronic, Electrical and Computer Science, Kyungpook National University, Korea. His research interests currently are in the areas of cryptography, network security, and security applications.