社団法人　情報処理学会　研究報告
IPSJ SIG Technical Reports

2005－ＣＳＥＣ－30 (37)
2005／7／22

Efficient N-Party Password-based Authenticated Key Exchange Protocol

辛　星漢†　　古原　和邦†　　今井　秀樹†

† 〒153-8505 東京都目黒区駒場 4-6-1 東京大学生産技術研究所

E-mail: †shinsh@imailab.iis.u-tokyo.ac.jp, ††{kobara,imai}@iis.u-tokyo.ac.jp

あらまし　In this paper, we propose an efficient N-party password-authenticated key exchange (so-called N-PAKE) protocol after showing the intermediate step. For that, we propose another 3-party PAKE protocol that is a basis of the N-PAKE protocol. The N-PAKE protocol is remarkably efficient rather than the previous works and a per-client computational cost is independent on the group size. Specifically, each client involved in the protocol is required only four exponentiations and some negligible operations.

キーワード　鍵交換プロトコル、パスワード、オンライン攻撃、オフライン攻撃

Efficient N-Party Password-based Authenticated Key Exchange Protocol

SeongHan SHIN†, Kazukuni KOBARA†, and Hideki IMAI†

† Institute of Industrial Science, The University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan

E-mail: †shinsh@imailab.iis.u-tokyo.ac.jp, ††{kobara,imai}@iis.u-tokyo.ac.jp

Abstract　In this paper, we propose an efficient N-party password-authenticated key exchange (so-called N-PAKE) protocol after showing the intermediate step. For that, we propose another 3-party PAKE protocol that is a basis of the N-PAKE protocol. The N-PAKE protocol is remarkably efficient rather than the previous works and a per-client computational cost is independent on the group size. Specifically, each client involved in the protocol is required only four exponentiations and some negligible operations.

Key words　authenticated key exchange, passwords, on-line and off-line dictionary attacks

1. Introduction

One of the important cryptographic primitives is an authenticated key exchange (AKE) protocol by which a client and a server (in the 2-party case) can authenticate each other and share the same session key to be used for subsequent algorithms. In particular, many researchers have paid much attention to the password-based authenticated key exchange (PAKE) protocols due to its practical aspects (see the Jablon's research link [12] for extensive surveys). However, the difficulty of designing a secure password-based protocol resides in the fact that password is chosen from a small set of values. The low entropy of passwords are always making password-based protocols susceptible to on-line or off-line dictionary attacks. While on-line attacks are possible by simply guessing the password and impersonating the client, we cannot completely rule out these attacks. The only way we can do is to limit an adversary's capability to the on-line case only and to prevent from giving sufficient information enough to verify off-line whether the guessed password is correct or not.

While most researches are focusing on the 2-party case, only a few (e.g., [1], [7], [11], [14]~[18]) took into account the problem in the 3-party setting where the clients who are sharing passwords with a common server want to establish a session key via the server that is considered to be a trusted third party (TTP). However, some of them turned out to be insecure, re-fixed and re-broken and so on. At PKC2005, Abdalla et al., [1] have proposed a generic construction of a 3-party AKE from any secure 2-party one. There are three phases in their generic construction. In the first phase, a high-entropy session key is generated between the server and each of the two clients using an instance of the 2-party PAKE protocol for each client. In the second phase, a message authentication code (MAC) key is distributed by the server to each client using a 3-party key distribution protocol. In the final phase, both clients execute an authenticated version of the Diffie-Hellman key exchange protocol [10] using the MAC keys obtained in the previous phase. They also showed another security notion in addition to AKE security: key privacy with respect to server.(注 1) Later, the same author [2] claimed that their construction for 3AKE at PKC2005 is not efficient so that they have proposed its efficient construction, by using only a secure 2PAKE protocol, whose security is based on stronger variants of the decisional Diffie-Hellman assumption. Very recently, Byun et al., [9] have considered the PAKE protocols in N-party case, which is an extended 3-party case, where $N-1$ clients who are sharing passwords with a common server want to establish a session key via the server. The core idea is to combine encrypted key exchange [8] with password-based group key exchange [3], [5] protocol not to expose some information about password to an adversary. Though the situation is more realistic and reasonable, their protocols don't enjoy either efficiency or security notion (i.e., key privacy).

ORGANIZATION. In Section 2., we propose another 3-party PAKE protocol that is a basis of our main contribution in Section 3., less efficient rather than [2], but enjoys its security proof with the standard Diffi-Hellman assumption. Section 3. presents an efficient N-party PAKE protocol, which is a combination of the construction in Section 2. and Burmester and Desmedt's protocol, along with comparison with the previous works.

2. A Three-Party PAKE Protocol

In this section, we show another Three-party PAKE (called T-PAKE) protocol, different from [2], in that the former explicitly provides mutual authentication between client and server while the latter does not. In fact, the T-PAKE protocol is an intermediate one for the N-party case in Section 3.. Here we consider the following situation where the clients who are sharing passwords with a common server want to establish a session key each other.

2.1 Preliminaries

First we explain some notations to be used throughout this paper. The security of the present and the next protocols are based on the computa-

(注 1) : That means a shared session key between the clients should be hidden from the point of view of server where the server is assumed to be honest but *curious* about the session key.

tional Diffie-Hellman (CDH) assumption, which we describe now. Let G be a cyclic group of prime order q and let g be a fixed generator of G. Informally, the CDH problem is to compute g^z from g^x and g^y where $z = xy$; G is said to satisfy the CDH assumption if it is computationally infeasible in an appropriate security parameter. One standard way to generate a group is to choose primes p, q such that $p = aq + 1$ and let G be the subgroup of order q in $Z_p^\star$ where (g, p, q) is the represented group. However, other choices of G are also possible.

Let k and l denote the security parameters, where k ($k > l$) can be thought of as the general security parameter for the CDH assumption (say, 1024 bits) and l can be thought of as the security parameter for hash functions (say, 160 bits). Let D be a dictionary size (cardinality) of passwords. Let $\{0,1\}^\star$ denote the set of finite binary strings and $\{0,1\}^l$ the set of binary strings of length l. If A is a set, then $a \stackrel{R}{\leftarrow} A$ indicates the process of selecting a at random and uniformly over A. Let "$||$" denote the concatenation of bit strings in $\{0,1\}^\star$. Let us define secure one-way hash functions. While $\mathcal{G} : \{0,1\}^\star \rightarrow Z_p^\star$ denotes a full-domain hash (FDH) function, the other hash functions are denoted $\mathcal{H}_j : \{0,1\}^\star \rightarrow \{0,1\}^l$ for $j = 1, 2, 3$ and 4. Here $\mathcal{G}$ and $\mathcal{H}_j$ are distinct random functions one another. Let A, B and S denote the identities of client A, B and server S, respectively.

2.2 Protocol Description

The T-PAKE protocol is mainly constructed on the password-based key exchange protocol of [4] in the 2-party setting where each party exchanges a password-masked Diffie-Hellman public value and the Diffie-Hellman key is used to generate each authenticator and the same session key. The whole description of the T-PAKE protocol is given in Fig. 1.

The protocol consists of three rounds of message. First, each client computes a Diffie-Hellman public value with a random element in $Z_q^\star$ and raising g to the that power, masks it using the output of $\mathcal{G}$ with each password as the input, and sends the result to the server. Upon receiving a message from each client, the server un-masks these messages to recover each client's Diffie-Hellman public value, chooses two random numbers in $Z_q^\star$ and computes each Diffie-Hellman public value. Each exponentiation of the recovered Diffie-Hellman public value to the z-th power results in a keying material KM to be used to generate an authenticator. Additionally, the server raises KM_2 (resp., KM_1) to the power z_1 (resp., z_2), masks it using the output of $\mathcal{G}$ with pw_A (resp., pw_B) as input and sends to client A (resp., B) the masked result of the randomized keying material of their partner Y_2 (resp., X_2) along with the Diffie-Hellman public value and the authenticator Z_1, H_1 (resp., Z_2, H_2).(注 2) Upon receiving a message from the server, each client computes the keying material for checking its authenticator, recovers the randomized keying material of his partner and computes the Diffie-Hellman key K as well as the session key SK via a hash function $\mathcal{H}$ using as input K and the transcript of the conversation among the clients and the server. The session identification is defined by the transcript $T = X_1||X_2||Y_1||Y_2||Z_1||Z_2$ of the conversation among the server and the clients along with their identity strings. In the third round of messages, each client sends his authenticator to the server. For key confirmation between client A and B, they can exchange hashed values of the session key after the T-PAKE protocol.

2.3 Discussions

Two security notions (i.e., AKE security and key privacy(注 3)) of the T-PAKE protocol follow directly from the security of the underlying 2-party PAKE protocol [4]. Though we do not prove its security here, the T-PAKE protocol is provably secure in the random oracle model under the CDH assumption (or its variants). This construction also indicates that small modifications of secure 2-party PAKE protocols can lead to secure 3-party AKE ones, however, more complicated proof is needed.

The T-PAKE protocol is, in fact, less efficient rather than [2] since each pair of client and server runs a separate 2-party PAKE protocol (for mutual authentication) and the randomized keying material should be sent to each client in order to compute

(注 2) : The messages that the server exchanged with that partner are omitted in Fig. 1. for clarity.

(注 3) : Any Diffie-Hellman assumption implies key privacy if the server contributes its secret value to the Diffie-Hellman key.

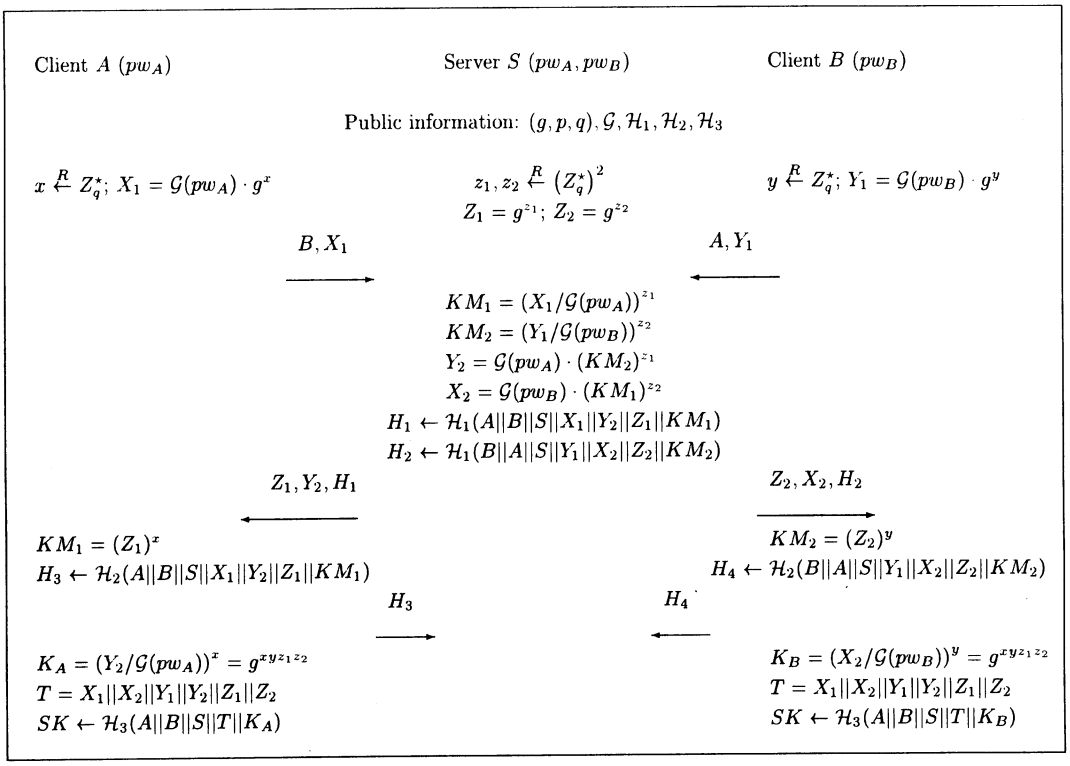

図 1 A Three-party PAKE (for short, T-PAKE) protocol that is secure against identity misbinding attack and provides mutual authentication between client A/B and server S. The T-PAKE protocol is AKE-secure and has key privacy with respect to server S as [2]. The first round of the T-PAKE protocol is a password-masked Diffie-Hellman public value, followed by a set of Diffie-Hellman public values and authenticator that is just the hashed values easily computable by a pair of parties. Client A and B check the received authenticator prior to accepting the session key SK

the Diffie-Hellman key K. However, this can be useful when considering asynchronous (or delayed) networks since the server can run the protocol first with client A and later transmit the masked value of the randomized keying material of client B. Another difference of the T-PAKE protocol from [2] lies in the fact that the clients can assure that they are sharing the same session key at completing the former protocol while in the latter the clients should exchange authenticators to make sure that.

More important is that the core techniques of the T-PAKE protocol are exploited in order to construct an efficient N-party PAKE protocol without interactions between the clients in Section 3..

3. An Efficient N-Party PAKE Protocol

In this section, we describe an efficient N-party PAKE (N-PAKE) protocol based on the group key exchange protocol of Burmester and Desmedt [6]. However, a simple combination of Section 2. and [6] doesn't necessarily provide security against off-line dictionary attacks.

3.1 Burmester and Desmedt's Protocol

For an easier understanding of the N-PAKE protocol, we explain the Burmester and Desmedt protocol briefly. When N parties A_i $(1 \leq i \leq N)$ wish to generate a session key, they proceed as follows (the indices are taken modulo N so that party A_0 is A_N and party A_{N+1} is A_1):

Round 1 Each party A_i chooses a random number x_i in $Z_q^\star$ and broadcasts $X_i = g^{x_i}$.
Round 2 Each party A_i broadcasts $W_i = (X_{i+1}/X_{i-1})^{x_i}$.
Key Computation Each party A_i computes their session key as:

$$K_i = (X_{i-1})^{N \cdot x_i} \cdot W_i^{N-1} \cdot W_{i+1}^{N-2} \ldots W_{i+N-2} \cdot (1)$$

It can be easily verified that all parties compute the same key $g^{x_1x_2+x_2x_3+\ldots+x_Nx_1}$.

3.2 Protocol Description

Unlike [6], a client in the N-PAKE protocol sends every message to the remaining clients of the group via server. For simplicity, we assume that a group G and a generator g have been fixed in advance and are known to all clients (including server) in the network; however, this assumption can be avoided at the expense of an additional round in which the server simply generates and broadcasts these values in a secure manner. The whole description of the N-PAKE protocol is given in Fig. 2.

The protocol consists of four rounds of message. First, client A_i computes a Diffie-Hellman public value X_i with a random element x_i in $Z_q^\star$ and raising g to the that power, masks it using the output of $\mathcal{G}$ with the password pw_{A_i} as the input, and sends the result to the server. Upon receiving the message X_i^* from client A_i, the server un-masks these messages to recover client A_i's Diffie-Hellman public value, chooses two random numbers y, z_i in $Z_q^\star$ and computes a Diffie-Hellman public value Z_i with z_i. The exponentiation of the recovered Diffie-Hellman public value to the z_i-th power results in a keying material KM_i to be used to generate an authenticator. Additionally, the server raises (X_{i+1}/X_{i-1}) to the power y, masks it using the output of $\mathcal{G}$ with pw_{A_i} as input and sends to client A_i the masked result of the randomized Diffie-Hellman public values of clients A_{i+1} and A_{i-1} along with the Diffie-Hellman public value Z_i and the authenticator H_{i1}. Upon receiving the message from the server, client A_i computes the keying material KM_i and checks its authenticator H_{i1}. If it is valid, client A_i recovers the randomized Diffie-Hellman public value, raises it to the power x_i and sends the result W_i with his authenticator H_{i2}. If the authenticator is valid, the server computes X'_{i-1}, by raising X_{i-1} to the power $y \cdot N$, as well as W'_i as the partial part of the Diffie-Hellman key K_i, and sends X'_{i-1}, W'_i with its authenticator.(注 4) If authenticator H_{i3} is valid, client A_i computes the key K_i from two values X'_{i-1}, W'_i and the session key SK via a hash function $\mathcal{H}$ using as input K_i and the transcript of the conversation among the clients and the server. The session identification is defined by the transcript $T = \{X_i^*||Y_i^*||Z_i||W_i||X'_i||W'_i\}_{1 \leqq i \leqq N}$ of the conversation among the server and the clients along with their identity strings. For key confirmation among clients A_i $(1 \leqq i \leqq N)$, they can exchange hashed values of the session key after the N-PAKE protocol.

CORRECTNESS. In an honest execution of the N-PAKE protocol, all clients have the same key K_i:

$$\begin{aligned} K_i &= (X'_{i-1})^{x_i} \cdot W'_i \\ &= (X_{i-1})^{y \cdot N \cdot x_i} \cdot W_i^{N-1} \cdot W_{i+1}^{N-2} \ldots W_{i+N-2} \\ &= (X_{i-1})^{x_i \cdot y \cdot N} \cdot \left(\frac{X_{i+1}}{X_{i-1}}\right)^{x_i \cdot y \cdot (N-1)} \\ &\quad \cdot \left(\frac{X_{i+2}}{X_i}\right)^{x_{i+1} \cdot y \cdot (N-2)} \ldots \left(\frac{X_{i+N-1}}{X_{i+N-3}}\right)^{x_{i+N-2} \cdot y} \\ &= (X_{i-1})^{x_i \cdot y} \cdot (X_{i+1})^{x_i \cdot y} \\ &\quad \cdot (X_{i+2})^{x_{i+1} \cdot y} \ldots (X_{i+N-1})^{x_{i+N-2} \cdot y} \\ &= g^{y(x_{i-1}x_i + x_ix_{i+1} + x_{i+1}x_{i+2} \ldots x_{i+N-2}x_{i+N-1})} \quad (2) \end{aligned}$$

3.3 Discussions

As in Section 2., we can assure that the following proposition holds, however, the proof is more complicated and intricate to deal with.

[Proposition 1] The N-PAKE protocol of Fig. 2. is AKE-secure and provide key privacy with respect to server in the random oracle model under the CDH assumption.

The number of modular exponentiations is a major factor to evaluate efficiency of a cryptographic protocol because that is the most power-consuming operation. In Table 1, we compare efficiency as well as key privacy with respect to server of 3-party and N-party PAKE protocols. We can say that the N-PAKE protocol is remarkably efficient compared to EKE-U [9] in that each client and server

(注 4) : The messages that the server exchanged with the remaining parties are omitted in Fig. 2. for clarity.

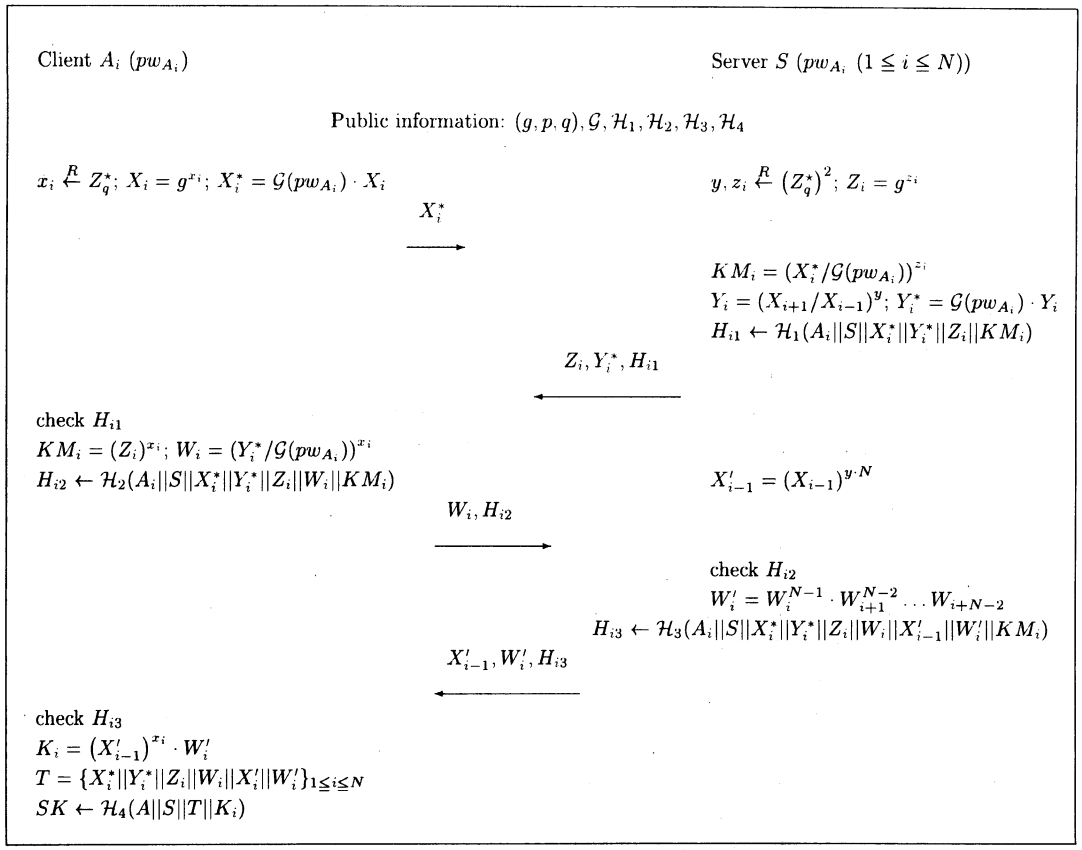

図 2 An efficient N-party PAKE (for short, N-PAKE) protocol

are required to compute four (注 5) and $3N$ modular exponentiations, respectively, while in the latter both are around $O(N^2)$. In particular, the remaining number of modular exponentiations after precomputation becomes $2N$ in the N-PAKE protocol. On the other hand, the EKE-M protocol doesn't satisfy key privacy so that the server can always eavesdrop and know the content of communications between clients.

文　献

[1] M. Abdalla, P. A. Fouque, and D. Pointcheval. Password-based Authenticated Key Exchange in the Three-Party Setting. In *Proc. of PKC 2005*, LNCS 3386, pages 65-84. Springer-Verlag, 2005. The full version is available at `http://www.di.ens.fr/~pointche/pub.php?reference=AbPo05`.

[2] M. Abdalla and D. Pointcheval. Interactive Diffie-Hellman Assumptions with Applications to Password-based Authentication. In *Proc. of FC 2005*, Springer-Verlag, 2005. The full version is available at `http://www.di.ens.fr/~pointche/pub.php?reference=AbPo05`.

[3] E. Bresson, O. Chevassut, and D. Pointcheval. Group Diffie-Hellman Key Exchange Secure against Dictioinary Attacks. In Proc. of ASIACRYPT 2002, LNCS 2501, pages 497-514. Springer-Verlag, 2002.

[4] E. Bresson, O. Chevassut, and D. Pointcheval. New Security Results on Encrypted Key Exchange. In *Proc. of PKC 2004*, LNCS 2947, pages 145-158. Springer-Verlag, 2004. The full version is available at `http://www.di.ens.fr/~pointche/pub.php?reference=BrChPo04`.

[5] E. Bresson, O. Chevassut, D. Pointcheval, and J. J. Quisquater. Provably Authenticated Group Diffie-Hellman Key Exchange. In Proc. of 8th ACM Conference on Computer and Communications Security, pages 255-264, 2001.

[6] M. Burmester and Y. Desmedt. A Secure and Efficient Conference Key Distribution System. In *Proc. of EUROCRYPT'94*, LNCS 950, pages 275-286. Springer-Verlag, 1994.

[7] J. Byun, I. Jeong, D. Lee, and C. Park. Password-

(注 5) : Note that each party computes only four full-length exponentiations in G since $N \ll q$ in practice (typically, $q \approx 2^{160}$ while $N \ll 2^{32}$).

表 1 Comparison of 3-Party and N-Party PAKE protocols where N is the number of clients

Protocols		The number of modular exponentiations				Key privacy
		Each client		Server		
		Total	Pre-comp.*1	Total	Pre-comp.*1	
3-Party	GPAKE [1]	4	1	4	2	guaranteed
	3PAKE [2]	2	1	2		guaranteed
	T-PAKE	3	1	6	2	guaranteed
N-Party	EKE-U [9]	$\frac{1}{2}\left(N^2+3N\right)$ *2	2 *3	$\frac{1}{2}\left(N^2+3N-1\right)$		guaranteed
	EKE-M [9]	2	1	$2N$	N	not guaranteed
	N-PAKE	4	1	$3N$	N	guaranteed

*1: The number of modular exponentiations that are pre-computable

*2: On average

*3: Only for the first client

Authenticated Key Exchange between Clients with Different Passwords. In *Proc. of ICICS 2002*, LNCS 2513, pages 134-146. Springer-Verlag, 2002.

[8] S. M. Bellovin and M. Merritt. Encrypted Key Exchange: Password-based Protocols Secure against Dictioinary Attacks. In *Proc. of IEEE Symposium on Security and Privacy*, pages 72-84. IEEE Computer Society, 1992.

[9] J. Byun and D. Lee. N-Party Encrypted Diffie-Hellman Key Exchange using Different Passwords. In *Proc. of ACNS 2005*, LNCS 3531, pages 75-90. Springer-Verlag, 2005.

[10] W. Diffie and M. Hellman. New Directions in Cryptography. In *IEEE Transactions on Information Theory*, Vol. IT-22(6), pages 644-654, 1976.

[11] L. Gong. Optimal Authentication Protocols Resistant to Password Guessing Attacks. In *Proc. of the 8th IEEE Computer Security Foundation Workshop*, pages 24-29, 1995.

[12] D. Jablon. Research Papers on Password-based Cryptography. `http://www.jablon.org/passwordlinks.html`.

[13] H. Krawczyk. SIGMA: The 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and Its Use in the IKE Protocols. In *Proc. of CRYPTO 2003*, LNCS 2729, pages 400-425. Springer-Verlag, 2003.

[14] C. L. Lin, H. M. Sun, and T. Hwang. Three-Party Encrypted Key Exchange: Attacks and a Solution. *ACM SIGOPS Operating Systems Review*, Vol. 34, No. 4, pages 12-20, 2000.

[15] R. C. W. Phan and B. Goi. Cryptanalysis of an Improved Client-to-Client Password-Authenticated Key Exchange (C2C-PAKE) Scheme. In *Proc. of ACNS 2005*, LNCS 3531, pages 33-39. Springer-Verlag, 2005.

[16] M. Steiner, G. Tsudik, and M. Waidner. Refinement and Extension of Encrpted Key Exchange. *ACM SIGOPS Operating Systems Review*, Vol. 29, No. 3. pages 22-30, 1995.

[17] S. Wang, J. Wang, and M. Xu. Weaknesses of a Password-Authenticated Key Exchange Protocol between Clients with Different Passwords. In *Proc. of ACNS 2004*, LNCS 3089, pages 414-425. Springer-Verlag, 2004.

[18] H. T. Yeh, H. M. Sun, and T. Hwang. Efficient Three-Party Authentication and Key Agreement Protocols Resistant to Password Guessing Attacks. *Journal of Information Science and Engineering*, Vol. 19, No. 6, pages 1059-1070, 2003.