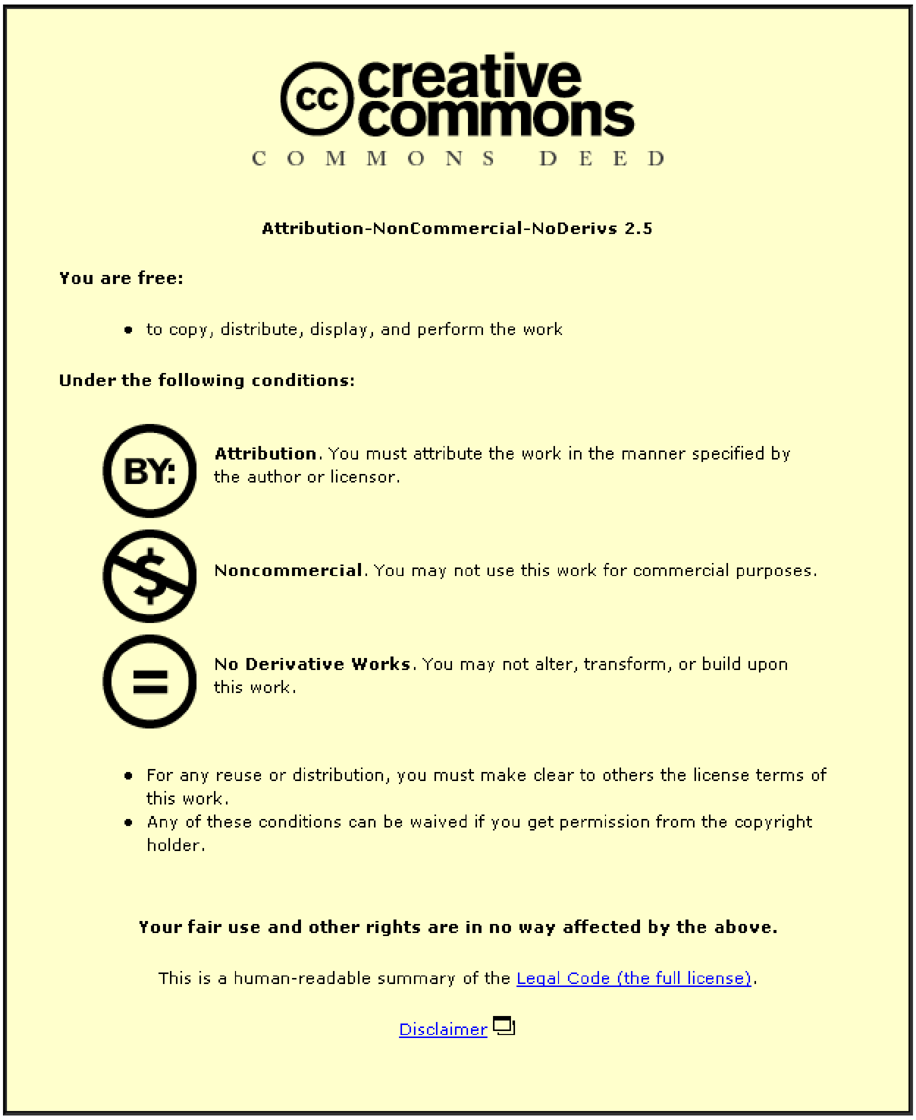

2009 International Conference on Advanced Information Networking and Applications Workshops

Security of Group Key Exchange Protocols with Different Passwords

Raphael C.-W. Phan[†]
Electronic & Electrical Engineering
Loughborough University, UK
Email: r.phan@lboro.ac.uk

Abstract

Password-based authenticated group key exchange protocols allow group users to jointly share a session key based on a human-memorizable password. In this paper, we present an undetectable online dictionary attack on N-EKE-D, a recent provably secure protocol designed to explicitly resist this type of attack. Thus, our result contradicts the design goal. We also give a simple attack on the key indistinguishability of N-EKE-D and two N-EKE-M variants that exploits the definition of partnering in their security model.

Keywords: *Password-authenticated key exchange, group, model, proof, key indistinguishability, undetectable online dictionary attack, cryptanalysis.*

1. Introduction

With password authenticated key exchange (PAKE) protocols [4], [10], [11], [17], [24], parties can jointly establish a secret key for securing the communication between them. Such protocols take into consideration the fact that systems often involve humans, who cannot be expected to remember highly random-looking secrets.

The first known PAKE secure against dictionary attacks is the Encrypted Key Exchange (EKE) by Bellovin and Merritt [4], for 2 parties. Extensions for groups appear, e.g. in [1], [24], [6], [12].

While formally treated group key exchange protocols have more conventionally been based on a common password shared among all group parties [6], more recently due formal treatment has been given to the idea of using different unique passwords between each party and a central server [1]. In this paper, we consider the latter setting.

The first provably secure group PAKE protocol for n parties is due to Byun et al. [12]. However, attacks were found on this protocol [25], and as a consequence, this protocol was revised in [13], [14], but further attacks were shown in [20], [18]. A variant was given by Nam to resist his attack in [18], while another variant was given at SPC 2006 [15] to cater to mobile ad-hoc networks. All these protocols are also constant-round in the sense that the number of communication rounds is constant irrespective of group size; they make use of broadcast channels.

Our Results. We consider the security of N-EKE-D and N-EKE-M protocol variants of [13], [14], [18], [15]. Although current protocols require a trusted server S, the advantage of this setting is that it partitions the trust of the group secret among the group members, thus in the event of compromise e.g. the shared password is leaked by the compromised member, the remaining non-compromised members can safely establish future group session keys without needing any change to the members' individual passwords.

The existence of undetectable online dictionary attacks is an issue [16], [25], [20], [21], [26] because while it is accepted that online dictionary attacks are inevitable for password-based protocols, the typical mitigation (which is outside the scope of protocol design) is to limit the number of failed login attempts. However, this measure will only work if failed attempts can be detected in the first place. Yet for undetectable online dictionary attacks, incorrect password guesses and thus failed attempts go unnoticed by the legitimate protocol participant being attacked by the adversary.

Hence, the adversary's active attacks via Send queries (in the context of typical security models; see Section II) cannot be differentiated from honest executions of a legitimate and honest protocol participant, so security against undetectable online dictionary attacks cannot be bound in terms of Send queries in the same way as detectable online dictionary attacks.

N-EKE-D [15] comes with an explicit proof of security against undetectable online dictionary attacks

978-0-7695-3639-2/09 $25.00 © 2009 IEEE
DOI 10.1109/WAINA.2009.38

by a malicious insider. We contradict this proof by giving such an attack. The oversight in the proof is due to the pre-supposition of how the adversary would behave, when it is exactly this that the definition and use of security models aim to avoid. See [25], [19], [20], [21], [18], [22], [23] for other examples of attacks found on protocols at times even ones with provable security goals.

Furthermore, we show that an inconsistency exists between the definition of partnering in the security model and how the group session key is generated, and this leads to a simple but subtle attack on key indistinguishability that fails to be captured by proofs of the protocols. This in fact is related to showing that the protocols do not meet the correctness requirement [2], [5], [3] of AKE protocols, which defines that if two instances of protocol participants are partnered and have accepted, then both should hold the same session key.

To the best of our knowledge, our results are the first known analysis of the protocols in [15], [18].

2. Preliminaries

2.1. Model for Group Key Exchange

For completeness and a much clearer understanding of our attack descriptions in later sections, we describe here the group key exchange (GKE) security model [7], [8], [9].

PROTOCOL PARTICIPANTS AND EXECUTION. Let $\mathcal{U}$ be a non-empty set of protocol players or parties. The adversary, A, controls the communications between all protocol players by interacting with the set of oracles, $\Pi^s_{U_i}$, where Π^s_U is defined to be the s^{th} instantiation of a player, $U_i \in \mathcal{U}$, in a specific protocol run. The adversary A controls the communication channels via the queries to the targeted oracles. A description of the oracle query types is presented as follows.

- $\mathsf{Send}(\Pi^s_{U_i}, m)$ query. This query models adversary A sending messages to instances of players. A gets back from his query the response which oracle $\Pi^s_{U_i}$ would have generated in processing message m. If oracle $\Pi^s_{U_i}$ has not yet terminated and the execution of protocol leads to accepting, variables SIDS are updated. A query of the form $\mathsf{Send}(\Pi^s_{U_i}, \text{“}start\text{”})$ initiates an execution of this protocol.
- $\mathsf{Reveal}(\Pi^s_{U_i})$ query. Any oracle upon receiving such a query and if it has accepted and thus holds some session key K, sends this back to A.
- $\mathsf{Corrupt}(U)$ query. This query allows the adversary to learn the long-lived key of user U. Under the strong corruption model, internal data of any instances of U executing the protocol are also given to A. Under the weak corruption model, only the long-lived key is given to the adversary.
- $\mathsf{Test}(\Pi^s_{U_i})$ query. This query is the only oracle query that does not correspond to any of A's abilities or any real-world event, and is only available if $\Pi^s_{U_i}$ is “fresh”. This query allows to define the indistinguishability-based notion of security for the key, defined by the following game, denoted $\mathbf{Game}^{\mathsf{GKE}}(A, P)$, between adversary A and oracles $\Pi^s_{U_i}$ involved in executions of protocol P. During the game, A can ask any of the above queries, and may only ask the Test query once. Depending on a randomly chosen bit, $b \in \{0, 1\}$, A is given the actual session key if $b = 1$ or a session key drawn randomly from the session key distribution if $b = 0$. Finally, A outputs a guess b'. Informally, A succeeds if it can guess the bit b with non-negligible advantage $\mathbf{Adv}^{\mathsf{GKE}}_{P,A}$ over randomly guessing, where the advantage is defined as
$$\mathbf{Adv}^{\mathsf{GKE}}_{P,A} = 2\Pr[b' = b] - 1.$$

Note that the first three queries: $\mathsf{Send}, \mathsf{Reveal}, \mathsf{Corrupt}$ are common for any kind model for authenticated key exchange protocols, to model the adversary's ability to attack the protocol. The final query Test is used to define the security of the protocol for which the adversary aims to break, in this case, that of the indistinguishability of the session key.

2.2. N-Party EKE Protocols

Let $U_1, \ldots, U_n$ be the identities in lexical order of n users. Denote by $\mathbb{G}$ a finite cyclic group of order $q \in \mathbb{Z}^*_p$, where p and q are two primes such that $p = 2q + 1$, and p a safe prime such that the Decisional Diffie Hellman (DDH) is hard in $\mathbb{G}$. Let g denote a generator of $\mathbb{G}$ of order q, and $||$ denotes concatenation. All arithmetic operations in this paper are performed under the group $\mathbb{G}$.

The n-party EKE-D protocol due to Byun et al. in [15] involves n group members and 1 server, and is specially designed to suit particularly multicast networks. Recall that *multicast* networks allow for communication between a single sender and multiple receivers. For multicast networks, all messages from individual single senders can be sent in parallel during a single round to all receivers, thus more round-efficient group-based protocols can be designed in such networks. Denote $\mathcal{H}_i$, for $i = \{1, 2, 3\}$ as an ideal hash function, H a collision resistant hash function, and $\mathcal{E}_{pw}(\cdot)$ the encryption under secret password pw.

EKE-D consists of two rounds. Round 1 is a simultaneous run of a 2-party PAKE between each group member U_i with the server S to set up a secure channel (in the confidentiality sense) between them. In Round 2, the server distributes a common keying message to all clients via the secure channel. This will be used to form the common secret session key sk among all group members. See Fig. 1.

In fact, N-EKE-D is one of the latest variants that builds on the N-EKE-M protocol originally proposed in [12]. Due to attacks in [25], the designers revised the protocol in [13], [14], which we denote as N-EKE-M$^+$. This is shown in Fig. 2. Subsequently an attack appeared in [18] where another variant was further proposed, for which we denote as N-EKE-M^{++}.

3. Cryptanalysis of N-EKE Protocols

In this section we will consider the security of the latest three variants of the N-EKE protocol initiated by the work of Byun et al. [12]. These variants are namely the N-EKE-D [15], N-EKE-M$^+$ [14] and N-EKE-M^{++} [18].

3.1. Attack on N-EKE-D

We describe an undetectable online dictionary attack by a malicious insider which applies to N-EKE-D [15], a variant of the N-EKE-M designed for multi-layer ad-hoc networks. This attack directly contradicts the security of N-EKE-D since N-EKE-D was explicitly proven to be secure against this type of attack.

1) The malicious insider adversary U_i initiates a protocol session where only S is activated, while other group members U_j $(j \neq i)$ can be absent.
2) In Round 1, U_i computes $y'_j = \mathcal{E}_{pw'_j}(g^{x_j})$ for $j = 1 \dots n, j \neq i$ where pw'_j is its guess of the password shared between U_j and S and x_j is any value. It also computes its own contribution $y'_i = \mathcal{E}_{pw_i}(g^{x_i})$ as normal. It sends y'_j (for $j = 1 \dots n$) to S.
3) When S broadcasts $t_j = \mathcal{E}_{pw_j}(g^{s_j})$ (for $j = 1 \dots n$), these are eavesdropped by U_i.
4) In Round 2, S broadcasts $\overline{sk}_j \oplus N$ (for $j = 1 \dots n$) which are eavesdropped by U_i. Since U_i can compute $N = (\overline{sk}_i \oplus N) \oplus \overline{sk}_i$, it can thus compute $\overline{sk}_j = (\overline{sk}_j \oplus N) \oplus N$ (for $j = 1 \dots n$). This means it can compute $\mathcal{H}_2(\overline{sk}_j || U_j)$ (for $j = 1 \dots n$) to be sent to S for correct verification.
5) U_i then computes $sk'_j = \mathcal{H}_1(\mathsf{sid}||(\mathcal{E}^{-1}_{pw'_j}(t_j))^{x_j})$ and checks if $H(sk'_j) = \overline{sk}_j$. If not, it repeats from step (1.) with a different password guess pw'_j.

Note that this attack allows the adversary U_i to try passwords pw_j of all U_j (for $j = 1 \dots n, j \neq i$) in parallel, thus U_i can verify $n - 1$ password guesses in each session rather than just one.

It is intriguing to note that the N-EKE-D has an explicit theorem proving that it is secure against undetectable online dictionary attacks by a malicious insider. The flaw in the proof is that the designers pre-supposed on the behaviour of the adversary, and in doing so, overlooked the fact that an insider adversary can obtain N and thus any $\overline{sk}_j$ $(j = 1 \dots n)$ which can be used to verify the adversary's password guess as in step 5 of the above attack. Similarly, the proof also overlooked the fact that since the adversary can obtain N and thus any $\overline{sk}_j$ $(j = 1 \dots n)$, the authenticator $\mathcal{H}_2(\overline{sk}_j || U_j)$ can be forged easily even if $\mathcal{H}_2$ is an ideal hash function.

Proving the security of a protocol within a security model by assuming only on what resources an adversary can access to, has the advantage that it captures all types of attacks that an adversary can mount given those resources. However, if the proof pre-supposes that the adversary attacks in a particular way, then there is a risk, as shown above, that the proof will fail to capture other attacks where the adversary behaves differently. Thus in this case the advantage of proving security within a generic model is lost.

3.2. Key (In)Distinguishability of N-EKE-D, N-EKE-M$^+$, and N-EKE-M^{++}

Here we describe a simple attack on the N-EKE-D protocol [15] and all N-EKE-M protocols [14], [18] except the original [12]. It exploits the definition of partnering in the N-EKE-M and N-EKE-D security model in [12], and breaks the key indistinguishability [15] of the protocols.

We restate the definition here for completeness.

Partnering [12]. Let the session identifier sid of a participant instance be the concatenation of *all* the messages between S and all group members U_j $(j = 1 \dots n)$. Any pair of instances U_i and U_k of S and U_j $(j = 1 \dots n)$ are said to be partnered if and

	S	U_1	$\cdots$	U_n
Round 1:	$s_i \in_\$ \mathbb{Z}_q^*$ Broadcast: $\mathcal{E}_{pw_i}(g^{s_i})$	$x_1 \in_\$ \mathbb{Z}_q^*$ Broadcast: $\mathcal{E}_{pw_1}(g^{x_1})$	$\cdots$ $\cdots$	$x_n \in_\$ \mathbb{Z}_q^*$ Broadcast: $\mathcal{E}_{pw_{n-1}}(g^{x_n})$
Round 2:	$sk_i = \mathcal{H}_1(\mathsf{sid}\|\|g^{x_i s_i})$	$sk_1 \quad = \mathcal{H}_1(\mathsf{sid}\|\|g^{x_1 s_1})$	$\cdots$	$sk_n = \mathcal{H}_1(\mathsf{sid}\|\|g^{x_n s_n})$
	$\overline{sk}_i = H(sk_i)$ $\mathsf{List}_{sc} = \{\overline{sk}_1, \overline{sk}_2, \ldots, \overline{sk}_n\}$ $N \in_\$ \mathbb{Z}_q^*$ Broadcast: $N \oplus \overline{sk}_i$, $\mathcal{H}_2(\overline{sk}_i\|\|S)$,	$\overline{sk}_1 = H(sk_1)$ Broadcast: $\mathcal{H}_2(\overline{sk}_1\|\|U_1)$	$\cdots$ $\cdots$	$\overline{sk}_n = H(sk_n)$ Broadcast: $\mathcal{H}_2(\overline{sk}_n\|\|U_n)$
Key Computation:	Check: $\mathcal{H}_2(\overline{sk}_i\|\|U_i)$	Check: $\mathcal{H}_2(\overline{sk}_1\|\|S)$	 $\cdots$	Check: $\mathcal{H}_2(\overline{sk}_n\|\|S)$
	Everyone computes: $sk = \mathcal{H}_3(\mathsf{sids}\|\|N)$ where $\mathsf{sids} = \mathsf{sid}'\|\|sk_1 \oplus N\| \ldots \|sk_n \oplus N$ and $\mathsf{sid}' = \mathcal{E}_{pw_1}(g^{x_1})\|\| \ldots \|\|\mathcal{E}_{pw_n}(g^{x_n})$			

Figure 1. N-EKE-D [15]

only if sid_i=sid_k and they compute the same session key.

With this definition, sid then includes all broadcast messages sent and received by S and U_j ($j = 1 \ldots n$), including the authenticator messages $\mathcal{H}_2(\cdot)$.

We describe the attack as applied to N-EKE-M$^+$ in [14]. It is straightforward to see that it also works on other variants in [15], [18].

1) Adversary A initiates a protocol session.
2) In Round 2, A issues a Send query to cause U_i to receive some chosen x instead of $\mathcal{H}_2(sk_j||S)$. Note that U_i does not use this message in any of its computations, since this broadcast authenticator is actually for S to verify but U_i receives it anyway due to the broadcast channel; so the rest of the protocol proceeds normally.
3) A issues a Reveal query to U_i and obtains the session key sk.
4) A issues a Test query to U_j ($j \neq i$) and obtains the test session key sk^*.
5) A checks if $sk^* = sk$ and outputs $b = 1$ if this is true; otherwise it outputs $b = 0$.

Issuing a Reveal query to U_i does not affect the freshness of the instance of U_j ($j \neq i$) since by the definition of partnership restated above, U_i will have an sid_i that differs from U_j's sid_j thus they will not be partnered. This makes it valid to issue a Test query to U_j. Furthermore, since the computation of the session key sk does not involve the changed authenticator message $\mathcal{H}_2(sk_j||S)$, then clearly U_i and U_j will compute the same sk.

This result breaks the key indistinguishability security of the N-EKE-D, N-EKE-M$^+$ and N-EKE-M^{++} protocols.

The oversight in the design that led to this attack is because in extending the original N-EKE-M protocol of [12] to newer variants in [14], [15], [18], the computation of the session key sk was not redefined to include the authenticator messages. To be precise, if the session key computation is a function of all broadcast messages to be consistent with the partnering definition, this attack can be captured in the key indistinguishability proof in [15].

This fact can also be seen from the view of correctness, which is defined such that if instances are partnered and have accepted (with a session key), then they should hold the same session key. Hence, if they are not partnered, they should therefore not hold the same key. But this latter case was shown in the context of the above attack.

4. Conclusion

We have treated group key exchange protocols in the setting where users each shares a different pass-

	S	U_1	$\cdots$	U_n
Round 1:	$s_i \in_{\$} \mathbb{Z}_q^*$ Broadcast: $\mathcal{E}_{pw_i}(g^{s_i})$	$x_1 \in_{\$} \mathbb{Z}_q^*$ Broadcast: $\mathcal{E}_{pw_1}(g^{x_1})$	$\cdots$ $\cdots$	$x_n \in_{\$} \mathbb{Z}_q^*$ Broadcast: $\mathcal{E}_{pw_{n-1}}(g^{x_n})$
Round 2:	$sk_i = \mathcal{H}_1(\mathsf{sid}\|\|g^{x_i s_i})$ $N \in_{\$} \mathbb{Z}_q^*$ Broadcast: $N \oplus sk_i$, $\mathcal{H}_2(sk_i\|\|S)$,	sk_1 $\mathcal{H}_1(\mathsf{sid}\|\|g^{x_1 s_1})$ Broadcast: $\mathcal{H}_2(sk_1\|\|U_1)$	$= \quad \cdots$ $\cdots$	$sk_n = \mathcal{H}_1(\mathsf{sid}\|\|g^{x_n s_n})$ Broadcast: $\mathcal{H}_2(sk_n\|\|U_n)$
Key Computation:	Check: $\mathcal{H}_2(sk_i\|\|U_i)$	Check: $\mathcal{H}_2(sk_1\|\|S)$	 $\cdots$	Check: $\mathcal{H}_2(sk_n\|\|S)$
	Everyone computes: $sk = \mathcal{H}_3(\mathsf{sids}\|\|N)$ where $\mathsf{sids} = \mathsf{sid}'\|\|sk_1 \oplus N\|\| \ldots \|\|sk_n \oplus N$ and $\mathsf{sid}' = \mathcal{E}_{pw_1}(g^{x_1})\|\| \ldots \|\|\mathcal{E}_{pw_n}(g^{x_n})$			

Figure 2. N-EKE-M$^+$ [14]

word with the server. Notably, we have shown an undetectable online dictionary attack against N-EKE-D which directly contradicts its security proof in its Theorem 3.4 [15].

We also showed that due to correctness issues caused by inconsistencies exist between the definition of partnering in security models of protocols in [13], [14], [18], [15] and the group session key computation or verification computations.

Acknowledgements

We thank anonymous referees for comments. We thank God for granting us the ability to remember passwords [Phillippians 1:3].

References

[1] M. Abdalla, P.-A. Fouque, and D. Pointcheval. Password-Based Authenticated Key Exchange in the Three-Party Setting. In *Proc. PKC '05*, LNCS 3386, pp. 65-84, Springer-Verlag, 2005.

[2] M. Abdalla, and D. Pointcheval. A Scalable Password-Based Group Key Exchange in the Standard Model. In *Advances in Cryptology - ASIACRYPT '06*, LNCS 4284, pp. 332-347, Springer-Verlag, 2006.

[3] S.S. Al-Riyami, and K.G. Patterson. Tripartite Authenticated Key Agreement Protocols from Pairings. In *Proc. IMA Cryptography and Coding '03*, LNCS 2898, pp. 332-359, Springer-Verlag, 2003.

[4] S.M. Bellovin, and M. Merritt. Encrypted Key Exchange: Password-based Protocols Secure against Dictionary Attacks. In *Proc. IEEE S&P '92*, pp. 72-84, IEEE Press, 1992.

[5] S. Blake-Wilson, D. Johnson, and A. Menezes. Key Agreement Protocols and Their Security Analysis. In *Proc. IMA Cryptography and Coding '97*, LNCS 1355, pp. 30-45, Springer-Verlag, 1997.

[6] E. Bresson, O. Chevassut, and D. Pointcheval. Group Diffie Hellman Key Exchange Secure against Dictionary Attacks. In *Advances in Cryptology - ASIACRYPT '02*, LNCS 2501, pp. 497-514, Springer-Verlag, 2002.

[7] E. Bresson, O. Chevassut, D. Pointcheval and J.-J. Quisquater. Provably Authenticated Group Diffie-Hellman Key Exchange. In *Proc. ACM CCS '01*, pp. 255-264, 2001.

[8] E. Bresson, O. Chevassut and D. Pointcheval. Provably Authenticated Group Diffie-Hellman Key Exchange - the Dynamic Case. In *Advances in Cryptology - ASIACRYPT '01*, LNCS 2248, pp. 290-309, 2001.

[9] E. Bresson, O. Chevassut and D. Pointcheval. Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions. In *Advances in Cryptology - EUROCRYPT '02*, LNCS 2332, pp. 321-336, 2002.

[10] M. Burmester, and Y. Desmedt. A Secure and Efficient Conference Key Distribution System (Extended Abstract). In *Advances in Cryptology - EUROCRYPT '94*, LNCS 950, pp. 275-286, Springer-Verlag, 1994.

[11] M. Burmester, and Y. Desmedt. A Secure and Scalable Group Key Exchange System. In *Information Processing Letters*, Vol. 94, No. 3, pp. 137-143, 2005.

[12] J.W. Byun, and D.H. Lee. N-Party Encrypted Diffie-Hellman Key Exchange Using Different Passwords. In *Proc. ACNS '05*, LNCS 3531, pp. 75-90, Springer-Verlag, 2005.

[13] J.W. Byun, and D.H. Lee. Comments on Weaknesses in Two Group Diffie-Hellman Key Exchange Protocols. IACR ePrint Archive, 2005/209, 2005.

[14] J.W. Byun, D.H. Lee, and J. Lim. Password-based Group Key Exchange Secure against Insider Guessing Attacks. In *Proc. CIS '05*, LNAI 3802, pp. 143-148, Springer-Verlag, 2005.

[15] J.W. Byun, S.-M. Lee, D.H. Lee, and D. Hong. Constant-Round Password-based Group Key Generation for Multi-Layer Ad-Hoc Networks. In *Proc. SPC '06*, LNCS 3934, pp. 3-17, Springer-Verlag, 2006.

[16] Y. Ding, and P. Horster. Undetectable On-line Password Guessing Attacks. In *ACM Operating Systems Review*, Vol. 29, No. 4, pp.77-86, 1995.

[17] D. Jablon. Strong Password-only Authenticated Key Exchange. In *ACM Computer Communications Review*, Vol. 20, No. 5, pp. 5-26, 1996.

[18] J. Nam. Enhancing Security of a Group Key Exchange Protocol for Users with Individual Passwords. IACR ePrint Archive, 2007/166, 2007.

[19] R.C.-W. Phan, and B.-M. Goi. Cryptanalysis of an Improved Client-to-Client Password-Authenticated Key Exchange (C2C-PAKE) Scheme. In *Proc. ACNS '05*, LNCS 3531, pp. 33-39, Springer-Verlag, 2005.

[20] R.C.-W. Phan, and B.-M. Goi. Cryptanalysis of the N-Party Encrypted Diffie-Hellman Key Exchange using Different Passwords. In *Proc. ACNS '06*, LNCS 3989, pp. 226-238, Springer-Verlag, 2006.

[21] R.C.-W. Phan, and B.-M. Goi. Cryptanalysis of Two Provably Secure Cross-Realm C2C-PAKE Protocols. In *Progress in Cryptology - Indocrypt '06*, LNCS 4329, pp. 104-117, Springer-Verlag, 2006.

[22] K. Ouafi, and R.C.-W. Phan. Privacy of Recent RFID Authentication Protocols. In *Proc. ISPEC '08*, LNCS 4991, pp. 263-277, Springer-Verlag, 2008.

[23] K. Ouafi, and R.C.-W. Phan. Traceable Privacy of Recent Provably-Secure RFID Protocols. In *Proc. ACNS '08*, LNCS 5037, pp. 479-489, Springer-Verlag, 2008.

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

[25] Q. Tang, and L. Chen. Weaknesses in Two Group Diffie-Hellman Key Exchange Protocols. IACR ePrint Archive, 2005/197, 2005.

[26] W. Wang, and L. Hu. Efficient and Provably Secure Generic Construction of Three-Party Password-based Authenticated Key Exchange Protocols. In *Progress in Cryptology - INDOCRYPT '06*, LNCS 4329, pp. 118-132, Springer-Verlag, 2006.