SCIS 2008 The 2008 Symposium on Cryptography and Information Security
Miyazaki, Japan, Jan. 22-25, 2008
The Institute of Electronics, Information and Communication Engineers

Improving Choi *et al.*'s ID-based Authenticated Group Key Agreement Scheme at PKC2004

Hyewon Park * Kyusuk Han * Chan Yeob Yeun * Kwangjo Kim *

Abstract— In modern network computing, authenticated group key agreement (AGKA) is important for conferencing. After Shamir [2] proposed the ID-based cryptosystem in 1984, the various applications on the ID-based cryptosystem have been actively studied, due to the simple key management complexity. For the AGKA, Choi *et al.* [16] proposed an ID-based authenticated group key agreement with bilinear maps, which was extended from Burmester and Desmedt conference key agreement protocol [5]. After that, Zhang and Chen [15] showed that the impersonation attack on Choi *et al.* protocol is feasible when two malicious users have the previous authentication transcripts of the entity. Shim [19] showed that the insider colluding attacks can be done without the previous transcripts. In this paper, we propose an improved ID-based AGKA. In our scheme, Key Generation Center (KGC) keeps the list of randomized user index instead of only generating private key. The random user indexing means KGC shuffles the order of users' indices by randomizing to hide it so that the malicious users cannot know the order. KGC also verify all users than only verifies 3 users in Choi *et al.*'s protocol. Our protocol can prevent replay attack of Zhang and Chen and insider colluding attack of Shim.

Keywords: Authenticated Group Key Agreement, ID-based Cryptosystem, Random User Indexing

1 Introduction

Recently, many conference systems exist like IP telephony, video conferencing, collaborative workspace and chatting for supporting reliable group communication, and they need that their private conference or working are secure. Key agreement protocol is that two or more entities establish a shared secret key. Diffie-Hellman [1] first introduced the key agreement protocol, which allows two entities can share a key without exchanging key material before the session starts. However, this protocol suffers from man-in-the-middle attack. Some works [6, 7, 8] to solve this attack were proposed. The key agreement protocol can be extended to group security, which is called group key agreement (GKA). Many collaborative and distributed systems can use GKA for their security. GKA allows users to share a common secret key which is committed by each member. In addition to this protocol, an authenticated group key agreement (AGKA) provides mutual key authentication for users during key sharing process. This AGKA protocol is required to be mandatory for the real applications.

Among various authentication schemes, ID-based cryptosystem has been rapidly used to authenticate because of its simplicity. In 1984, Shamir [2] firstly introduced the concept of ID-based cryptosystem. In this system, each user already knows the public identity of the users and uses it as the public key. It doesn't need any public key infrastructure (PKI), so the cryptosystem can be simplified. After that, ID-based scheme is improved and applied to key agreement protocol [3, 4, 9, 17].

ID-based cryptosystem has been applied for AGKA for reducing the managing complexity of public keys. Several papers have tried to establish ID-based group key agreement schemes. Reddy and Nalla [12] proposed bilinear pairing and one-way function tree (OFT) based group key agreement scheme, and analyzed informally that their protocol satisfies implicit key authentication. However, it suffers from man-in-the-middle attack, and requires much time. The scheme based on ternary tree was proposed by Barua *et al.*[13]. Their protocol is extended version of Joux's [10] tripartite key agreement protocol. It is similar structure compared with Reddy and Nalla scheme, but it uses bilinear map. This protocol is secure against passive attack, but it requires $\log_3 n$ rounds. Du [14] *et al.*'s scheme resists against the impersonation attack. Their scheme has constant 2 communication rounds, but group members must keep synchronization because of time constant. Shi *et al.* scheme [18] has only one communication round, and it uses bilinear pairing. They formally verify their protocol about implicit key authentication, known session key security, forward secrecy and no key compromise impersonation. However, it requires n^2 of computation time. Choi *et al.* [16] (denoted by CHL for short) proposed an ID-based authenticated group key agreement with bilinear maps, which was extended from Burmester and Desmedt conference key agreement protocol [5]. It also uses bilinear pairing, and has 2 con-

* Information and Communications University, Munji-dong, Yuseong-gu, Daejeon, 305-732 Korea, {inde1776, hankyusuk, cyeun, kkj}@icu.ac.kr

stant rounds. After that, Zhang and Chen [15] showed that the impersonation attack on CHL protocol is feasible when two malicious users have the previous authentication transcripts of the entity, and Shim [19] showed that the insider colluding attacks is possible without previous transcripts.

In this paper, we review the CHL protocol and propose an improved ID-based AGKA scheme. Our design can prevent the insider colluding attack on CHL scheme in the real application using random user indexing. Also we compare and analyze our protocol with other ID-based AGKA protocols.

Our paper organized as follows. In the following section, we discuss some preliminaries, such as Diffie-Hellman problem and bilinear pairing. In Section 3, we review CHL protocol with Burmester and Desmedt protocol which is the basic building block of the protocol. In Section 4, attacks on CHL protocol are reviewed. We present our improved ID-based AGKA protocol in Section 5, and compare and analyze it with other ID-based AGKA protocols in security and performance in Section 6. We finally conclude our paper in Section 7.

2 Preliminary

In this section, we state some assumptions briefly, such as Diffie-Hellman problems and admissible bilinear map. Also we define system setting for ID-based public key infrastructure which is used in CHL protocol and our protocol.

2.1 Diffie-Hellman Problem

1. Parameter Generator:

   A CDH parameter generator IG_{CDH} is a probabilistic polynomial time algorithm that takes a security parameter 1^k, runs in polynomial time, and outputs an additive group G of prime order q. A BDH parameter generator IG_{BDH} is a probabilistic polynomial time algorithm similar to CDH parameter, but outputs the description of two groups G_1 and G_2 of the same order q and an admissible bilinear map $e : G_1 \times G_1 \rightarrow G_2$.

2. Computational Diffie-Hellman (CDH):

   CDH problem in G is to compute abP when generator P of G and aP, bP for some $a, b \rightarrow Z_q^*$.

   $Pr[A(G, P, aP, bP) = abP$

   $|G \leftarrow IG_{CDH}(1^k); P \leftarrow G; a, b \leftarrow Z_q^*]$

3. Decisional Bilinear Diffie-Hellman (DBDH):

   DBDH problem in $[G_1, G_2, e]$ is to distinguish between tuples of the form

   $(P, aP, bP, cP, e(P, P)^{abc})$

   and $(P, aP, bP, cP, e(P, P)^d)$.

2.2 Admissible Bilinear Pairing

To define admissible bilinear map, some of notions have to be predefined. G_1 and G_2 are two groups of the same prime order q, more precisely, G_1 is an additive group and G_2 is a multiplicative group. P is an arbitrary generator of G_1. Assume that discrete logarithm problem (DLP) is hard in both G_1 and G_2. A mapping $e : G_1 \times G_1 \rightarrow G_2$ satisfying the following three properties is called an admissible bilinear map from a cryptographic point of view:

1. Bilinearity :

   $e(P_1, Q)e(P_2, Q) = e(P_1 + P_2, Q)$

   $e(P, Q_1)e(P, Q_2) = e(P, Q_1 + Q_2)$

   i.e. $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_q^*$.

2. Non-degeneracy : If a generator $P \in G_1$, then $e(P, P)$ is a generator of G_2. In other words, $e(P, P) \neq 1$.

3. Computable : There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

2.3 ID-based System Setting

CHL protocol is based on the ID-based public key infrastructure. It consists of a Key Generation Center (KGC) and users. KGC generates the system parameter,

$$\text{param} = < G_1, G_2, q, e, P, P_{pub}, H, H_1 >.$$

G_1 is an additive group and G_2 is a multiplicative group with order q. e is an admissible bilinear pairing and H and H_1 are the hash functions, $H : \{0, 1\}^* \rightarrow Z_q^*$ and $H_1 : \{0, 1\}^* \rightarrow G_1$.

Set Up: KGC chooses a random $s \in Z_q^*$ as the secret master key, and choose a random P. Then KGC computes $P_{pub} = sP$.

Private Key Extraction: With ID, KGC produces the public key, $Q_{ID} = H(ID)$, where H is hash function. The private key is $S_{ID} = sQ_{ID}$. When there are n users who are going to agree a shared key, each user has their own identity $ID_i, 1 \leq i \leq n$. Each user U_i who has ID_i as his identity has his own static key pair $< Q_i, S_i >$.

3 CHL Protocol

CHL protocol is considered to be a bilinear variant of the BD protocol [5]. In this section, we review the BD conference keying protocol and CHL protocol in brief.

3.1 BD Protocol

Burmester and Desmedt assumed the complete graph-type network that the users can broadcast messages to each other in their protocol. The indices are taken in a cycle, so next user of U_n is U_1 when n users are in a group. Diffie-Hellman key distribution system [1] is extended for the protocol. Let n users through U_1 to U_n to the set of users who are going to share a common secret key.

1. Each user U_i selects $r_i \in_R Z_q$, and computes and broadcasts his individual Diffie-Hellman exponentials

   $z_i = \alpha_i^r$ mod p.

2. U_i computes and broadcasts

   $X_i = (z_{i+1}/z_{i-1})^{r_i}$ (mod p)

3. U_i computes the key

   $K_i = z_{i-1}^{nr_i}\ X_i^{n-1}\ X_{i+1}^{n-2}\ \ X_{i-2}$ (mod p).

After operating above protocol, all users in a group have one common shared key K, where $K = K_i$.

$$K = \alpha^{r_1r_2+r_2r_3+\ldots+r_nr_1} \text{ (mod p)}$$

3.2 CHL Protocol

Let n users through U_1 to U_n to the set of users who are going to share a common secret key. System setup and extraction follows the subsection 2.2. U_i's long term public/private key pair is $< ID_i, S_i >$.

Round 1. Each user select random $a_i \in Z_q^*$ as his own secret key, then computes and broadcasts

$< P_i = a_iP, T_i = a_iP_{pub} + h_iS_i >$,

where $h_i = H(P_i)$.

Round 2. After receive $< P_{i-1}, T_{i-1} >$, $< P_{i+1}, T_{i+1} >$, and $< P_{i+2}, T_{i+2} >$, each user U_i verifies

$e(\sum_{k\in\{-1,1,2\}} T_k, P)$

$= e(\sum_{k\in\{-1,1,2\}}(P_k + h_kQ_k), P_{pub})$

If the verification is satisfied, then U_i computes and broadcasts

$D_i = e(a_i(P_{i+2} - P_{i-1}), P_{i+1})$.

Key Computation. Each U_i computes the session key,

$$K_i = e(a_iP_{i-1}, P_{i+1})^n D_i^{n-1} D_{i+1}^{n-2} \ldots D_{i-2}.$$

After operating above protocol, all users in a group have one common shared key K, where $K = K_i$.

$$K = e(P, P)^{a_1a_2a_3+\ldots+a_{n-1}a_na_1+a_na_1a_2}$$

4 Attacks on CHL Protocol

CHL protocol only adapts partial authentication because users only need $< P, T >$ pair of U_{i-1}, U_{i+1} and U_{i+2} for their authentication. This means the protocol is not fully authenticated. Zhang and Chen [15] showed that the impersonation attack on CHL protocol is possible when two malicious users have the previous authentication transcripts of the entity (ZC Attack), and Shim [19] showed that the insider colluding attacks is possible without previous transcripts (Shim Attack).

4.1 ZC Attack

In round 2 of CHL protocol , D_i computation can be modified as follows.

$D_i = e(a_i(P_{i+2} - P_{i-1}), P_{i+1})$

$= e(a_i(a_{i+2}P - a_{i-1}P), a_{i+1}P)$

$= e(a_iP, a_{i+1}P)^{a_{i+2}-a_{i-1}}$

$= e(P_i, P_{i+1})^{a_{i+2}-a_{i-1}}$

This means any two malicious users can impersonate an entity if they have the previous authentication transcripts of this entity. This attack is feasible because they only consider the partial authentication. To solving this problem, Zhang and Chen suggested to use time parameter as a solution to replay attack.

4.2 Shim Attack

Zhang and Chen showed any two malicious users who have the previous transcript can impersonate an entity. Some papers proposed solution of this attack,[15][14] However, Shim showed that three malicious users U_{i-1}, U_{i+1}, and U_{i+2} can collude and impersonate U_i anytime.

Round 1. They select random $a_i \in Z_q^*$ and $R \in G_1$, then computes and broadcasts

$< P_i = a_iP,\ T_i = R >$.

Round 2. Each user verify

$< P_{i-1}, T_{i-1} >, < P_{i+1}, T_{i+1} >, < P_{i+2}, T_{i+2} >$.

However, they don't have to verify because all others except U_{i-1}, U_{i+1}, and U_{i+2} doesn't know the invalidity of U_i. Then they computes and broadcasts D_i to impersonate U_i.

$D_i = e(a_i(P_{i+2} - P_{i-1}), P_{i+1})$.

Key Computation. Each U_i computes the session key K_i, and malicious users succeed in impersonating U_i to the other users.

This attack shows that three malicious users can collude and impersonate user without replay attack. To prevent this attack, Shim suggested that each user should authenticated all participating entities for each round. From this solution, security of the protocol depends on the security of the signature scheme adapted to the protocol.

5 Our Proposed Scheme

In this section we propose an improved scheme of CHL protocol using random indexing. The term "Random Indexing" means that Key Generation Center (KGC) shuffle the order of users' indices by randomizing it. All user cannot know the indices of other users because KGC keeps them secret. The idea of improvement is given from that two attacks in Section 4 is possible only when malicious users know their index. In CHL

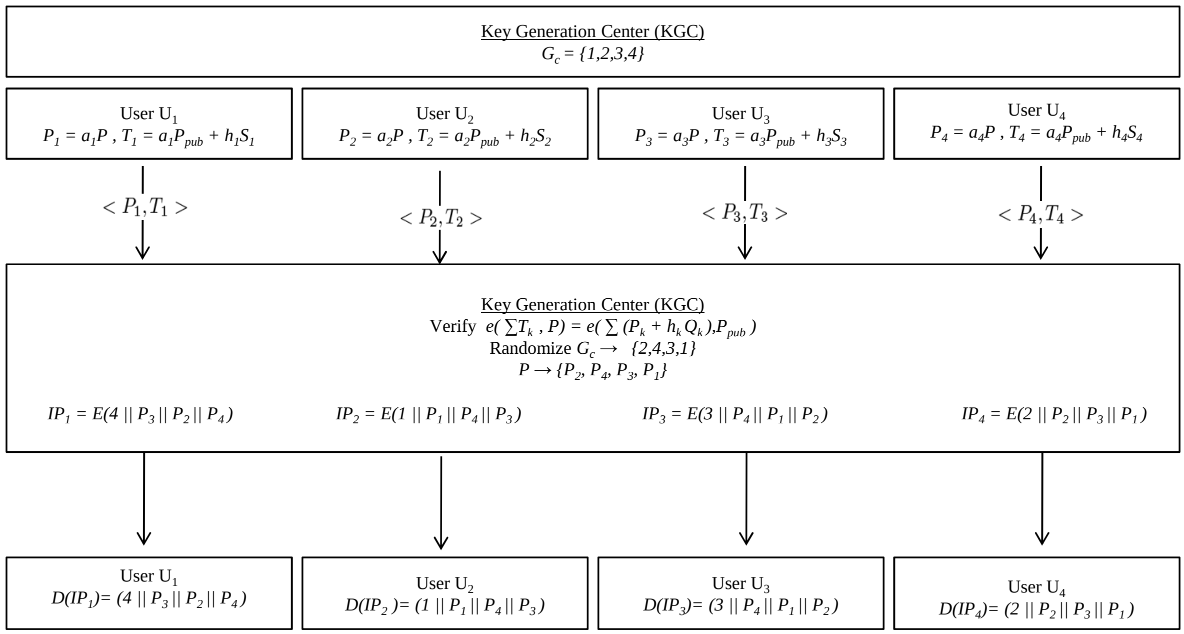

Figure 1: Rounds 1 and 2 of Our Improved Scheme with 4 Users

protocol, the malicious users can easily attack the user because they assumed the user index is fixed. Therefore, we propose that user cannot know his own index before round 2. We add the role of KGC to operate not only generating key but shuffling the user indexing. The diagram which shows round 1 and round 2 (mainly modified part) is in Figure 1.

Notations. All notations used in our scheme are same in CHL protocol. We only define one new notation, IP.

IP: IP is encrypted message which has new index of users with P_{i-1}, P_{i+1} and P_{i+2}. The message is encrypted using users public key. Only user who has his private key can decrypt IP.

Round 1. Each user select random $a_i \in Z_q^*$ as his own secret key, and computes

$$< P_i = a_iP,\ T_i = a_iP_{pub} + h_iS_i > .$$

Then send this $< P, T >$ pair to KGC.

Round 2. After receive all $< P, T >$ pairs from users, KGC verifies

$$e(\sum T_k, P)$$
$$= e(\sum (P_k + h_kQ_k), P_{pub})$$

where $k = 1, 2, \ldots, n$.

If the verification is satisfied, then KGC shuffles the index ordering by random and send

$$IP = E_{Q_{ID}}(\text{new index } ||P_{i-1}||P_{i+1}||P_{i+2})$$

to all users.

Round 3. Each user decrypts IP using their private key S_{ID} and gets $P_{i-1}, P_{i+1}, P_{i+2}$ with his new index. Then user computes and broadcasts

$$D_i = e(a_i(P_{i+2} - P_{i-1}), P_{i+1}).$$

Key Computation. Each U_i computes the session key,

$$K_i = e(a_iP_{i-1}, P_{i+1})^n D_i^{n-1} D_{i+1}^{n-2} \ldots D_{i-2}.$$

After operating above protocol, all users in a group have one common shared key K, where $K = K_i$.

$$K = e(P, P)^{a_1a_2a_3 + \ldots + a_{n-1}a_na_1 + a_na_1a_2}$$

Table 1: Comparison

	CHL	Ours
Round	2	3
Index order	Fixed / Public	Randomized / Hidden
Authentication	Partial	Full
ZC Attack	Possible	Impossible
Shim Attack	Possible	Impossible

6 Analysis

Our scheme is considered to be an improved version of CHL protocol. The comparison between CHL protocol and ours is summarized in Table 1. The improved scheme requires 3 rounds, and authenticates all users fully. The order of indices is randomized. It can prevent ZC and Shim attacks which are reviewed in section 4. In this section, we analyze our scheme in detail especially for security and performance compared with the CHL protocol.

6.1 Security Analysis

Our improvement is focused on security enhancement. In CHL protocol, the malicious users can easily attack the user because they assumed the user index is fixed. We changed the role of KGC to operate not only generating key but shuffling the user indexing and authenticating users as in Figure 1. In round 2 of CHL protocol, each user only verify U_{i-1}, U_{i+1}, and U_{i+2}, so it was not fully authenticated. Our protocol proposes that KGC first verifies all $< P, T >$ pairs not only U_{i-1}, U_{i+1}, and U_{i+2}, then shuffles user index ordering by random in round 2. This new index order keeps secret. After that, KGC encrypts and sends new $P_{i-1}, P_{i+1}, P_{i+2}$ with his new index to all user. Each user U_i gets only his new index and P values, but malicious users cannot get the values because they don't know U_i's private key S_{ID}.

Section 4.1 mentioned ZC attack is feasible because they only consider the partial authentication. To prevent this attack, KGC verifies all users with $< P, T >$ pairs. Moreover, user index is randomly changed each time by KGC and users cannot know their new index before receiving IP_i. Therefore, replay attack using previous authentication transcript is impossible in our protocol.

In Section 4.2 also mentioned another insider colluding attack by Shim that three malicious users U_{i-1}, U_{i+1}, and U_{i+2} can collude and impersonate U_i without previous transcript, which means it is not replay attack. This attack is possible because the user indices are fixed and all users already know the indices and all $< P, T >$ which correspond to user indices. In our protocol, index is shuffled and ordering of indices is hidden by KGC. All users cannot know the index ordering and decrypt the received ID_i except the user who has the secret key S_{ID_i}. Therefore, malicious user cannot attack honest user.

As above, our protocol can prevent previous two attacks and increases the cryptographic strength of CHL protocol.

6.2 Performance Analysis

Table 2 compares our protocol with the previous ID-AGKA protocols using big-O notation. CHL protocol has only 2 rounds and requires only small time for message, computation, and pairing times. Compared with other protocols, CHL protocol is most efficient in performance. But, our protocol is modification of CHL protocol, so we compared our protocol with CHL protocol. The protocol operates 3 rounds, which requires one more round than CHL protocol so round time in big-O notation is as before. Pairing and computation time does not changed because there is no change with computing key. Message time is increased for randomizing index, but it is trivial. Synthetically, the protocol does not increase the computation and communication cost enormously.

Table 2: Comparison of ID-AGKA Protocols

	Round	Message	Computation	Pairing
Reddy(02')	O(lg n)	O(n lg n)	O(n lg n)	O(n lg n)
Barua(03')	O(lg n)	O(n)	O(n)	O(n lg n)
Du(03')	O(1)	O(n)	O(n^2)	O(n)
Choi(04')	O(1)	O(n)	O(n)	O(n)
Shi(05')	O(1)	O(n)	O(n^2)	O(n)
Ours(07')	O(1)	O(n)	O(n)	O(n)

7 Conclusion

In this paper, we reviewed CHL protocol and attacks on this protocol. Zhang and Chen attacked to CHL protocol using replay attack, and Shim attacked the protocol using insider colluding attack. They just suggested simple solution, and did not give detail scheme. We proposed the improved version of CHL protocol. In the improved ID-based AGKA scheme, KGC operates shuffling user index and verifying all users not only generating user private key. Our scheme prevents the replay attack and insider colluding attack on CHL protocol by randomizing user index so increases security power from original protocol. In fact, the protocol needs more of trivial computations than CHL protocol, but it is trivial and does not increase the computation and communication cost enormously. Therefore, our protocol improve the security of CHL protocol with maintaining the performance.

References

[1] W. Diffie and M. Hellman, New Direction in Cryptography, IEEE Transactions on Information Theory 22(6), pp. 644-654, 1976.

[2] A. Shamir., Identity-based Cryptosystems and Signature Schemes, Advances in Cryptology-Crypto 84, LNCS 196, pp.47-53, Springer-Verlag, 1984.

[3] E. Okamoto, Proposal for Identity-based Key Distribution System, Electronics Letters, pp.1283-1284, 1986.

[4] C. Günther, An Identity-based Key Exchange Protocol, Proc. Eurocrypt '89, pp. 29-37, Springer-Verlag, 1990.

[5] M. Burmester and Y. Desmedt, A Secure and Efficient Conference Key Distribution System, Proc.

of Eurocrypt '94, pp.267-275, LNCS 950, Springer-Verlag, 1994.

[6] L. Law, A. Menezes, M.Qu, J.Solinas and S. Vanstone, An efficient protocol for authenticated key agreement, Technical Report CORR 98-05, 1998.

[7] S. Blake-Wilson and A. Menezes, Authenticated Diffie-Hellman Key Agreement Protocols, SAC98, LNCS 1556, pp.339-361, Springer-Verlag, 1999.

[8] B. Song and K. Kim. Two-Pass Authenticated Key Agreement Protocol with Key Confirmation, Proc. of Indocrypt 2000, LNCS 1977, pp.237-249, Springer-Verlag, 2000.

[9] M. Girault, Self-certified public keys, editor, Advances in Cryptology-Crypto 2000, pp. 333-352, Springer-Verlag, 2000.

[10] Joux, A One Round Protocol for Tripartite Diffie-Hellman, InW. Bosma, editor, Proceedings of Algorithmic Number Theory Symposium. ANTS IV, LNCS 1838, pp.385-394, Springer-Verlag, 2000.

[11] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, Proc. of Crypto 01, LNCS 2139, pp.213-229, Springer-Verlag, 2001.

[12] K. C. Reddy and D. Nalla, Identity Based Authenticated Group Key Agreement Protocol, Proc. of Indocrypt '02, LNCS 2551, pp.215-233, Springer-Verlag, 2002.

[13] R. Barua, R. Dutta and P. Sarkar. Extending Joux's Protocol to Multi Party Key Agreement, Proc. of Indocrypt '03, LNCS 2904, pp.205-217, Springer-Verlag, 2003.

[14] X. Du, Y. Wang, J. Ge and Y. Wang, An Improved ID-based Authenticated Group Key Agreement Scheme, Cryptology ePrint Archive, Report 2003/260.

[15] F. G. Zhang and X.F. Chen, Attack on Two ID-based Authenticated Group Key Agreement Schemes, Cryptology ePrint Archive: Report 2003/259.

[16] K. Y. Choi, J. Y. Hwang and D. H. Lee, Efficient ID- based Group Key Agreement with Bilinear Maps, PKC'04, LNCS 2947, pp.130-144, Springer-Verlag, 2004.

[17] W. D. Benits Jr, R. Terada, An IBE Scheme to Exchange Authenticated Secret Keys, Cryptology ePrint Archive, 2004.

[18] Y.Shi, G.Chen, J. Li, ID-Based One Round Authenticated Group Key Agreement Protocol with Bilinear Pairings, Cryptology ePrint Archive: Report, ITCC.2005.169.

[19] K. A. Shim, Further Analysis of ID-Based Authenticated Group Key Agreement Protocol from Bilinear Maps, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E90-A(1):295-298, 2007.