情報保護學會論文誌
第19卷 第5號, 2009. 10

내부자에 의한 위장 공격을 방지하는 개선된 ID 기반 그룹 인증 및 키 합의 프로토콜*

박 혜 원,[1†] Asano Tomoyuki,[2] 김 광 조[1‡]
[1]KAIST, [2]Sony Corporation

Improved ID-based Authenticated Group Key Agreement Secure Against Impersonation Attack by Insider*

Hyewon Park,[1†] Tomoyuki Asano,[2] Kwangjo Kim[1‡]
[1]KAIST, [2]Sony Corporation

요 약

인터넷에 존재하는 많은 그룹 기반 시스템들은 안전한 통신을 위해 그룹 인증 및 키 합의 (AGKA)를 필요로 한다. 1984년에 Shamir가 ID 기반 암호시스템을 제안한 이후 이 시스템의 공개키 관리 연산 감소를 이용한 ID 기반 AGKA 프로토콜이 지속적으로 연구되고 있다. 이와 관련해서 2006년에 Zhou 등은 두 번의 라운드를 가지면서 통신 및 연산에서 매우 효율적인 ID 기반 AGKA 프로토콜을 제안하였다. 본 논문에서는 이 프로토콜이 임시 그룹 비밀 정보를 소유하고 있는 악성 내부자에 의한 위장 공격에 취약하다는 것을 보이고 이를 개선한 ID 기반 AGKA 프로토콜을 제안한다. 제안된 프로토콜은 악성 공격자가 임시 그룹 비밀 정보를 알고 있더라도 각 구성원의 개인 키를 포함하는 서명을 사용하여 위장 공격이 불가능하도록 설계되었다. 또한 Zhou 등의 프로토콜에서 수행되었던 불필요한 연산을 감소하여 성능 또한 향상시켰다.

ABSTRACT

Many conference systems over the Internet require authenticated group key agreement (AGKA) for secure and reliable communication. After Shamir [1] proposed the ID-based cryptosystem in 1984, ID-based AGKA protocols have been actively studied because of the simple public key management. In 2006, Zhou *et al.* [12] proposed two-round ID-based AGKA protocol which is very efficient in communication and computation complexity. However, their protocol does not provide user identification and suffers from the impersonation attack by malicious participants. In this paper, we propose improved ID-based AGKA protocol to prevent impersonation attack from Zhou *et al.*'s protocol. In our protocol, the malicious insider cannot impersonate another participants even if he knows the ephemeral group secret value. Moreover, our protocol reduces the computation cost from Zhou *et al.*'s protocol.

Keywords: AGKA, group key agreement, ID-based, Impersonation

I. Introduction

In many conference systems or applications, the communication between the conference participants is exchanged through insecure channel like the Internet. According to this property of the systems, not only

접수일(2009년 6월 9일), 게재확정일(2009년 8월 24일)

* 본 연구는 Sony Corporation에서의 인턴쉽 과정으로 수행되었으며, 본 논문의 일부는 2009 Symposium on Cryptography and Information Security(SCIS 2009)에 발표된 바 있습니다.

† 주저자, inde1776@kaist.ac.kr

‡ 교신저자, kimkj@cs.kaist.ac.kr

honest but malicious users can easily eavesdrop or interrupt the communication. Therefore, the conference participants need their private communication to be secure and reliable, and many solutions for the secure conference system have been proposed so far. Group key agreement (GKA) is one solution for secure communication that more than two entities establish a shared secret key for their communication. Since users can encrypt or decrypt the messages with this established key, the secure and reliable communication can be achieved. In GKA, no participant can predetermine the value of the established session key. Additionally, GKA with authentication mechanism is called authenticated group key agreement (AGKA) and provides mutual key authentication during group key agreement process.

After Shamir proposed ID-based cryptosystem [1], ID-based AGKA protocols [8-10,12,14,15] have been proposed with the advantage of simple public key management. ID-based cryptosystem uses an identity information as a public key, so it does not need public key infrastructure. Also, Burmester and Desmedt [2] proposed constant-round GKA protocol over the broadcast channel. Communication time is always constant in this protocol because the participants are only required to broadcast once when they want to send a message to all the other participants. Many researchers recently address the above two approaches to design their GKA protocols.

In this paper, we review and analyze Zhou *et al.*'s two-round ID-based constant round AGKA protocol [12] because their protocol is considered to be one of the most efficient ID-based AGKA protocol comparing with the previous protocols. After that, we propose an improved ID-based constant-round AGKA protocol. Our protocol prevents impersonation attack on Zhou *et al.*'s protocol. We also prove the security of our protocol under DBDH and CDH problems.

Our paper organized as follows: In Section II, we review previous ID-based AGKA protocols. After introducing preliminaries in Section III, we review Zhou *et al.*'s two-round AGKA protocol and suggest how to do impersonation attack by malicious participants in Section IV. We present our improved ID-based AGKA protocol in Section V, and analyze in Section VI. We finally conclude our paper in Section VII.

II. Related Work

In this section, we briefly review some recent papers about ID-based constantround AGKA protocols. Choi *et al.* [8] proposed two-round ID-based AGKA protocol based on Burmester and Desmedt's GKA protocol in 2004. However, two papers showed impersonation attacks on this protocol: replay attack by Zhang and Chen [7] and insider colluding attack by Shim [13].

The protocol proposed by Kim *et al.* [9] requires only one communication round, but suffers from replay attack or passive attack because the equation for key computation can be computed from any other users.

Shi *et al.* [10] also proposed one-round AGKA protocol that used different type of ID-based public/private key pair with other protocols; however, Zhou *et al.* [12] showed insider attack that malicious insider can get the session key of any execution on this protocol.

Two AGKA protocols was proposed by Zhou *et al.*: one requires one communication round (ZSM-1) and the other requires two rounds (ZSM-2). ZSM-1 protocol requires much computation per each user and has key control problem. ZSM-2 protocol is efficient in computation, but suffers from impersonation attack by insider. We discuss the security of the ZSM-2 protocol in Section 4.

In 2008, Choi *et al.* [14] proposed an improved protocol from the previous one. This protocol can prevent passive attack or impersonation by additional signature and session identifiers.

Yao *et al.*'s AGKA protocol [15] requires 3 communication rounds, and each round is for identity authentication, key agreement, and key confirmation. This protocol also can prevent passive attack or impersonation.

III. Preliminaries

3.1 Security Model and Notions

Our security model follows Katz and Yung's [6] formal security model, which is extended version of Bresson *et al.*'s[4] model. Detailed definitions are described in [6].

Participants and Initialization. Each user U_i in a fixed, polynomial-size set $P=\{U_1,\dots,U_n\}$ of potential participants have the unique identity ID. We denote instance $s\in N$ of player U_i as Π_i^s.

In this model, an initialization phase occurs before the protocol runs at first. Then each participant U_i gets public/private keys (Q_i, S_i) by running an algorithm $G(1^k)$.

Adversarial Model. We assume that an adversary A can control all communications and ask an instance to release session key or long-term key. An adversary's queries are modeled by the following oracles.

$Send(U,i,M)$: Send message M to instance Π_U^i and outputs the reply generated by this instance.

$Execute(U_1,\dots,U_n)$: Execute the protocol between the players $U_1,\dots,U_n$ and outputs the transcript of execution.

$Reveal(U,i)$: Output the session key sk_U^i.

$Corrupt(U)$: Output the long-term secret key S_i.

$Test(U,i)$: A asks any of the above queries, and then asks $Test$ query only once. This query outputs a random bit b; if $b=1$, the adversary can access sk_U^i, and if $b=0$ he can only access random string.

A *passive adversary* can ask $Execute$, $Reveal$, $Corrupt$, $Test$ queries and an *active adversary* can ask all above queries including $Send$ query.

Protocol Security. The advantage of an adversary A in attacking protocol is defined as

$$Adv_A(k)=|2\Pr[Suc]-1|,$$

where Suc is the event that A's guess b' satisfies $b=b'$ for $Test$ query.

The GKA protocol is said to be secure if $Adv_A(k)$ is negligible for all probabilistic polynomial time (PPT) adversary A.

3.2 Bilinear Pairing

G_1 is an cyclic additive group and G_2 is a cyclic multiplicative group with same order q. Assume that discrete logarithm problem (DLP) is hard in both G_1 and G_2. A mapping $e: G_1\times G_1\rightarrow G_2$ which satisfies the following properties is called a bilinear pairing from a cryptographic point of view:

1) Bilinearity: $e(aP,bQ)=e(P,Q)^{ab}$ 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; that is, $e(P,P)\neq 1$.

3) Computable: There exists an efficient

algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

CDH Problem: A Computational Diffie-Hellman (CDH) parameter generator IG_{CDH} is a PPT algorithm takes a security parameter 1^k and outputs additive group G_1 with an order q.

When an algorithm A solves CDH problem with an advantage ϵ, the advantage is

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

where $P \in G_1$ and $a, b \in Z_q{}^*$.

DBDH Problem: A Bilinear Diffie-Hellman (BDH) parameter generator IG_{BDH} is a PPT algorithm takes a security parameter 1^k and outputs G_1 and G_2 and bilinear map e.

When an algorithm A solves Decisional BDH (DBDH) problem with an advantage ϵ, the advantage is

$$\left| \begin{array}{l} \Pr\left[A\left(P, aP, bP, cP, e(P,P)^{abc}\right) = 1\right] \\ -\Pr\left[A\left(P, aP, bP, cP, e(P,P)^{d}\right) = 1\right] \end{array} \right| \le \epsilon,$$

where $P \in G_1$ and $a, b, c, d \in Z_q{}^*$.

IV. ZSM-2 Protocol

4.1 Description

Here we focus on the two-round ID-based AGKA protocol, namely ZSM-2 protocol. Before the session starts, ID-based system setup [5] is done as follows:

Set Up. G_1 and G_2 are cyclic groups with order q, e is a bilinear pairing, P is an arbitrary generator of G_1, and H denotes a hash function, where $H: \{0,1\}^* \to Z_q^*$. Key Generation Center (KGC) chooses a random $s \in Z_q^*$ as the secret master key, and computes $P_{pub} = sP$.

$$param = \langle G_1, G_2, q, e, P, P_{pub}, H \rangle$$

Extraction. KGC generates the public/private key pair, $\langle Q_i = H(ID_i), S_i = sQ_i \rangle$.

There are n users, from U_1 to U_n, in a group who want to share a common secret key. U_1 is assumed to be an initiator of the group. Their protocol uses three hash functions, $H_4 : G_2 \to \{0,1\}^n$, $H_5 : \{0,1\}^n \to Z_q^*$, and $H_6 : G_1 \to \{0,1\}^n$. The protocol works as follows.

Round 1. Initiator U_1:

Pick $\delta \leftarrow G_2$, $r \leftarrow \{0,1\}^n$, $k_1 \leftarrow Z_P^*$

Compute $P_i = r \oplus H_4(e(S_1, Q_i) \cdot \delta)$,

where $2 \le i \le n$,

Compute & broadcast D_1

$$D_1 = \langle \delta, P_2, \ldots, P_n, X_1 = H_5(r) k_1 P, Y_1 = k_1 P_{pub}, L \rangle,$$

where L is a label containing users' association information.

Round 2. $U_i (2 \le i \le n)$:

Find appropriate P_i from D_1.

Then compute $r' = H_4(e(S_i, Q_1) \cdot \delta) \oplus P_i = r$,

Choose $k_i \leftarrow Z_P^*$ randomly.

Compute & broadcast D_i

$$D_i = \langle X_i, Y_i \rangle = \langle H_5(r) k_i P, k_i P_{pub} \rangle.$$

Key Computation. Each user computes

$$z_i = H_5(r)^{-1} X_i \, (1 \le i \le n)$$

Then verify the following equation. If fails, then the protocol halts.

$$e(P, \Sigma_{j=1}^n Y_j) = e(P_{pub}, \Sigma_{j=1}^n z_j)$$

Session Key. $K = K_i = H_6(z_1) \oplus \ldots \oplus H_6(z_n)$

4.2 Impersonation Attack

In ZSM-2 protocol, they did not consider about the existence of malicious participants. Also, their batch verification only executes if the message is correctly generated with secret value r, not if the message is sent by correct user. Therefore, the malicious insider who knows the secret value r can impersonate the other users, that is, impersonation attack by the insider will happen. The following is an attack on the protocol that the legitimated user U_m impersonates the user U_i.

Round 2. Malicious insider $U_m (i \neq m)$:
Inject the message which is sent to U_i.
Find appropriate P_m from D_1.
Compute $r' = H_4(e(S_m, Q_1) \cdot \delta) \oplus P_m = r$
Random $k_i \leftarrow Z_P^*$, $k_m \leftarrow Z_P^*$
Compute & broadcast D_i, D_m

$$D_i = \langle X_i, Y_i \rangle = \langle H_5(r) k_i P, k_i P_{pub} \rangle$$
$$D_m = \langle X_m, Y_m \rangle = \langle H_5(r) k_m P, k_m P_{pub} \rangle$$

Key Computation. All users succeed to verify D_i

$$e(P, \Sigma_{j=1}^{n} Y_j) = e(P_{pub}, \Sigma_{j=1}^{n} z_j).$$

Session Key. $K = K_i = H_6(z_1) \oplus \dots \oplus H_6(z_n)$

In Round 2 of the protocol, malicious user U_m can compute $\langle X_i, Y_i \rangle$ pair using r because the computation does not need any private information of U_i. Then all the other users believe that they agreed session group key with legitimate user U_i even though U_i does not exist. This attack can also occur with colluding of several malicious users.

V. Our Scheme

The impersonation attack by insider on the protocol is possible because their batch verification is not enough to identify each user and only depends on secret value r. Therefore, we improve the protocol that modify the batch verification in the protocol to include user's private key S_i so malicious users cannot impersonate the U_i even though they get r. Our protocol uses new hash functions, $H_1 : G_2 \rightarrow \{0,1\}^{|q|}$, $H_2 : \{0,1\}^{|q|} \rightarrow Z_q^*$, and $H_3 : G_1 \rightarrow \{0,1\}^{|q|}$. The other notations are the same in ZSM-2 protocol. Our protocol runs as follows:

Round 1. Initiator U_1:
Pick $\delta, k_1 \leftarrow Z_q^*$, $r \leftarrow \{0,1\}^{|q|}$
Compute $P_i = r \oplus H_4(e(\delta S_1, Q_i))$,
where $2 \leq i \leq n$,
Compute & broadcast D_1

$$D_1 = \left\langle \begin{matrix} \delta, P_2, \dots, P_n, X_1 = H_2(r \| L) k_1 P, \\ Y_1 = k_1 P_{pub} + H_2(r \| L) S_1, L \end{matrix} \right\rangle,$$

where L is a label containing the session information, such as the concatenation of all the user ID's.

Round 2. $U_i (2 \leq i \leq n)$:
Find appropriate P_i from D_1.
Then compute $r' = H_4(e(\delta S_i, Q_1)) \oplus P_i \doteq r$,
Choose $k_i \leftarrow Z_q^*$ randomly.
Compute & broadcast D_i

$$D_i = \langle X_i, Y_i \rangle = \langle H_2(r \| L) k_i P, k_i P_{pub} + H_2(r \| L) S_i \rangle.$$

Key Computation. Each user computes

$$z = H_2(r \| L)^{-1} \Sigma_{i=1}^{n} X_i = \Sigma_{i=1}^{n} k_i P$$

Then verify the following equation. If fails, then the protocol halts.

$$e(P, \Sigma_{j=1}^{n} Y_j) = e(P_{pub}, z + H_2(r \| L) \Sigma_{j=1}^{n} Q_j)$$

Session Key. $K = K_i = H_3(z)$

In our AGKA protocol, three points are improved from ZSM-2 protocol. (i) We define $\delta \leftarrow Z_q^*$ and change the encryption of secret value r in round 1 that δ is multiplied by S_1 in G_1 group. The multiplication in G_2 group takes much more time than in G_1 group in practice so we can reduce the time to encrypt r in our protocol. (ii) Multiplication of z is combined in our protocol to reduce the computation overhead. In key computation process, we use hash function so key control of specific user is still impossible. (iii) The most important feature is that we modify the batch verification. In our protocol, each user broadcasts $\langle X_i, Y_i \rangle = \langle H_2(r\|L)k_iP, k_iP_{pub} + H_2(r\|L)S_i \rangle$ to verify users. This computation includes the private key of each users so malicious user cannot make this value arbitrary. The batch verification in our protocol can be done with the following equation.

$$\begin{aligned} e(P, \Sigma_{j=1}^{n} Y_j) &= e(P, \Sigma_{j=1}^{n}(k_jP_{pub} + H_2(r\|L)S_j)) \\ &= e(P, \Sigma_{j=1}^{n}(k_jsP) + \Sigma_{j=1}^{n}(H_2(r\|L)S_j)) \\ &= e(P_{pub}, \Sigma_{j=1}^{n}(k_jP) + \Sigma_{j=1}^{n}(H_2(r\|L)Q_j)) \\ &= e(P_{pub}, z + H_2(r\|L)\Sigma_{j=1}^{n} Q_j) \end{aligned}$$

VI. Analysis

In this section, we analyze our ID-based AGKA protocol from the security and performance points of view.

6.1 Security

Our goal is to show that our protocol is secure against all types of adversary under DBDH and CDH assumptions. We show the security proof of our protocol in two: encryption and signature schemes.

6.1.1 Encryption

We first assume that an adversary A gains an advantage from attacking the encryption scheme $r \oplus H_1(e(\delta S_i, Q_i))$ without forging a signature.

Theorem 1. *The encryption scheme in above protocol is secure under the DBDH assumption in the Random Oracle Model (ROM). Namely:*

$$Adv_A \le 2q_{ex} Adv_G^{DBDH}$$

Proof. Let A be an active adversary and get advantage in attacking the encryption. We consider that A makes *Execute* query. The distribution of the transcript T and session group key K, where $2 \le i \le n$, is given by:

$$Real = \begin{bmatrix} \delta, k_1 \leftarrow Z_q^*,\ r \leftarrow \{0,1\}^{|q|}; \\ P_i = r \oplus H_1(e(\delta S_1, Q_i)); \\ r' = H_1(e(\delta S_i, Q_1)) \oplus P_i; \\ X_i = H_2(r\|L)k_iP,\ Y_i = k_iP_{pub} + H_2(r\|L)S_i; \\ T = \langle \delta, P_2, \dots, P_n, X_1, \dots, X_n, Y_1, \dots, Y_n \rangle; \\ K = H_3(z) \end{bmatrix}$$

Consider the distributions *Fake* defined as follows:

$$Fake = \begin{bmatrix} \delta, k_1, b_1, b_i \leftarrow Z_q^*,\ r \leftarrow \{0,1\}^{|q|}; \\ P_i = r \oplus H_1(e(\delta b_1 P_{pub}, b_iP)); \\ r' = H_1(e(\delta b_i P_{pub}, b_1P)) \oplus P_i; \\ X_i = H_2(r\|L)k_iP,\ Y_i = k_iP_{pub} + H_2(r\|L)S_i; \\ T = \langle \delta, P_2, \dots, P_n, X_1, \dots, X_n, Y_1, \dots, Y_n \rangle; \\ K = H_3(z) \end{bmatrix}$$

Let $\epsilon = Adv_G^{DBDH}$ and q_{ex} is the number of *Execute* queries issued by A. When choosing (T, K) pair randomly to ask *Test* query and getting b, A can distinguish $e(\delta S_1, Q_i)$ and $e(\delta b_1 P_{pub}, b_iP)$, and get bit b' from guessing with probability $\epsilon' (\le \epsilon)$ because he can obtain $b_1P, \dots, b_nP$ and $P_{pub} = sP$ is public. Therefore, we obtain the following equation:

$$\epsilon' = \left| \begin{array}{l} \Pr[T \leftarrow Real; K \leftarrow Real; A(T,K) = 1] \\ - \Pr[T \leftarrow Fake; K \leftarrow Fake; A(T,K) = 1] \end{array} \right| \le \epsilon$$

There is a *H−list* which contains all the messages that A queried before. Let *Ask* be the event that what A makes to the *Hash* query is on the *H−list* when A asks *Test* query. The advantage of A in correctly guessing the session key and breaking the encryption is:

$$
\begin{aligned}
Adv_A(k) &= |2\Pr[Suc]-1| = |2\Pr[b=b']-1| \\
&= 2\Pr[b=b'|\neg Ask]\Pr[\neg Ask] \\
&\qquad + 2\Pr[b=b'|Ask]\Pr[Ask]-1 \\
&= 2\Pr[b=b'|\neg Ask] + 2\Pr[b=b'|Ask]-1 \\
&= 2\Pr[b=b'|Ask] \\
&= 2q_{ex}\epsilon'
\end{aligned}
$$

A cannot gain the advantage without asking for it in ROM, so $2\Pr[b=b'|\neg Ask]-1=0$. By adapting a standard hybrid argument, we can have the result that the advantage of A breaking the encryption as follows:

$$Adv_A \leq 2q_{ex}Adv_G^{DBDH} \qquad \square$$

6.1.2 Signature

Second, we assume that A gains an advantage with forging a signature. In our protocol, we use an ID-based signature scheme Σ defined as follows:

Extract. Given an identity ID, compute public key $Q_{ID}=H(ID)$ and private key $S_{ID}=sQ_{ID}$.

Sign. Compute $Y=kP_{pub}+hS_{ID}$, where $k\in Z_q^*$, $h=H_2(r\|L)$; $\langle kP, Y\rangle \leftarrow \Sigma_{\geq n}(S_{ID})$.

Verification. Verify $e(P,Y)=e(P_{pub},kP+hQ_{ID})$, where $h=H_2(r\|L)$;
$True$ or $False \leftarrow \Sigma_{ver}(Q_{ID},\langle kP, Y\rangle)$.

Here we show the signature scheme Σ is secure against existential forgery on adaptively chosen ID attack as in the following theorem. The proof follows from [8,11].

Theorem 2. *Let the hash functions H and H_2 be random oracles and F_0 be a forger which performs an existential forgery under an adaptively chosen ID with running time t_0. The forger F_0 can ask queries to the $H, H_2, Extract$ and $Sign$ at most q_H, q_{H_2}, q_E, and q_S times, respectively. Suppose the advantage of F_0 is $\epsilon_0 \geq 10q_H(q_{S+1})(q_S+q_{H_2})/(q-1)$. Then there exists an attacker F that can solve the CDH problem within the expected time $t_2 \leq 120686q_{H_2}t_0/\epsilon_0$.*

We can prove Theorem 2 by proving the following Lemmas.

Lemma 1. *Let the hash functions H be random oracle and F_0 be a forger for an adaptively chosen ID with running time t_0 and advantage ϵ_0. Suppose F_0 can ask queries to the H at most q_H times. Then a forger F for a given ID has advantage $\epsilon_1 \leq \epsilon_0(1-1/q)/q_H$ with running time $t_1 \leq t_0$.*

Proof. F is given ID^*, and we assume that F_0 makes H, $Extract$ and $Sign$ queries at most once. F maintains a list L_H of $\langle ID_i, Q_i\rangle$ and interacts with F_0 after choosing $\alpha\in\{1,\dots,q_H\}$.

- When F_0 makes α-th H query on ID, F returns Q^* with H query for ID^* and inserts $\langle ID,Q^*\rangle$ into L_H if $ID=ID^*$. Otherwise, F returns result for ID, and inserts $\langle ID,Q\rangle$ into L_H.
- F_0 issues an $Extract$ query on Q_i. If $Q_i=Q^*$, then F outputs FAIL; otherwise, F returns S_i to F_0 as the result of $Extract$ query.
- When F_0 issues H_2 query on $r\|L$, F returns the result $H_2(r\|L)$.
- When F_0 makes $Sign$ query on ID_i, F returns $\langle ID_i, kP_i, Y_i\rangle$ to F_0.
- F_0 finally outputs $\langle ID', k'P, Y'\rangle$ then F finds $\langle ID', Q'\rangle$ in L_H. If $Q'=Q^*$, F outputs $\langle ID^*, k'P, Y'\rangle$, otherwise it fails.

Here, F succeeds the simulation with probability $1/q$ if $Q'\neq Q^*$ and $\langle ID', Q'\rangle$ is not in L_H because the output $\langle ID', k'P, Y'\rangle$ is independent of the information F_0 accumulated from the previous queries in this case. Therefore, the probability that F does not fail the simulation is $1/q_H(1-1/q)$.

Lemma 2. *Let the hash function H and H_2 be random oracles and F_0 be a forger for a given ID who has advantage $\epsilon_1 \geq 10(q_S+1)(q_S+q_{H_2})/q$ with running time t_1. Suppose F_0 can ask queries to the $H, H_2, Extract$ and $Sign$ at most q_H, q_{H_2}, q_E, and q_S times, respectively. Then there exists an attacker F can solve the CDH problem within expected time $t_2 \leq 120686 q_{H_2} t_1/\epsilon_1$.*

Proof. F sets the system parameters $param = \langle G_1, G_2, e, P, P_{pub}, ID, H, H_2 \rangle$, where $P_{pub} = xP$, and gives it to F_0. Given P, xP, and yP, F's goal is to compute xyP as CDH problem. F maintains two lists $L_H = \langle ID_i, a_i, Q_i \rangle$ and $L_Y = \langle ID_j, k_jP \rangle$, and interacts with F_0 as follows:

- When F_0 makes H query on ID, F returns $Q^* = yP$ for ID^*; otherwise F picks $a_i \in Z_q^*$ randomly, adds $\langle ID_i, a_i, Q_i \rangle$ to L_H, and returns $Q_i = a_iP$.
- F_0 issues an $Extract$ query on Q_i, if $Q_i = Q^*$ then F fails; otherwise, F finds $\langle ID_i, a_i, Q_i \rangle$ from L_H and returns $S_i = a_iP_{pub} = xQ_i$ to F_0.
- When F_0 issues H_2 query on $r \| L$, F picks $h_i \in Z_q^*$ randomly and returns it.
- When F_0 makes $Sign$ query on ID_i, F picks $k_i \in Z_q^*$ randomly, computes k_iP, and adds $\langle ID_i, k_iP \rangle$ to L_Y. Then F finds $\langle ID_i, a_i, Q_i \rangle$ from L_H, computes $Y_i = k_ixP + h_ia_ixP = k_iP_{pub} + h_iS_i$ and returns $\langle ID_i, k_iP, h_i, Y_i \rangle$ to F_0.

Finally, F_0 outputs a valid tuple $\langle ID^*, kP, h, Y \rangle$ where $\langle ID^*, kP \rangle$ is not in L_Y without accessing any oracles except H_2. If F replays with the same random tape but different choices of H_2 as in the *forking lemma* [3], then F_0 outputs two valid tuples $\langle ID^*, kP, h, Y \rangle$ and $\langle ID^*, kP, h', Y' \rangle$, where $h \neq h'$.

Here, F can computes $(Y-Y')/(h-h') = xyP$ as CDH problem if both of them are expected; otherwise, it fails. Therefore, the time for F is equal to the time for *forking lemma* and the time t_2 is bounded by $120686 q_{H_2} t_1/\epsilon_1$.

Combining Lemmas 1 and 2, we can obtain Theorem 2 with that the advantage of forger F in our protocol is negligible. □

6.2 Performance

Table 1 shows communication and computation cost of our protocol comparing with other ID-based AGKA protocols. We use the following notations:

n: Number of group members
#R: Total number of rounds
#U: Total number of unicast
#B: Total number of broadcast
#Exp: Total number of exponentiation

〔Table 1〕 Comparison of Performance

	[8]	[9]	[10]	[12]-1*	[12]-2**	[14]	[15]	Ours
#R	2	1	1	1	2	2	3	2
#U	0	0	$(n-1)^2$	0	0	0	0	0
#B	$2n$	n	0	n	n	$2n$	$3n$	n
#Exp	$n(n-1)$	0	0	0	0	$n(n-1)$	0	0
#G_1-M	$8n$	$n(n+4)$	n^2	0	$n(n+3)$	$11n$	$2n(n+3)$	$7n-1$
#G_2-M	$n(n-1)$	0	0	$2n(n-1)$	$2(n-1)$	$n(n-1)$	0	0
#$Pair$	$4n$	$n(4n-3)$	n	$2n(n-1)$	$3n$	$6n$	$n(n+5)$	$3n$

*: ZSM-1 **: ZSM-2

$\#G_1-M$: Total number of G_1 multiplication
$\#G_2-M$: Total number of G_2 multiplication
$\#Pair$: Total number of pairings

Our protocol has less multiplication cost than ZSM-2 protocol, and shows even the most efficient protocol in Table 1. Therefore, our proposed protocol can improve both the security and performance of ZSM-2 protocol.

VII. Conclusion

In this paper, we suggested a deterministic attack on the ZSM-2 protocol that a malicious insider who knows the secret value can impersonate the other user. To prevent this attack, we proposed an improved AGKA protocol which prevents impersonation attack by insider and reduces the computation cost. In our protocol, we used signature including user's private key, so an insider who even gets secret value cannot impersonate other users. Moreover, our protocol reduces multiplication cost in encryption and batch verification. An open problem is to provide perfect forward secrecy if all the previous transcripts and user's private keys are exposed, then the previous session key can be exposed. Except this problem, our protocol improve the security and performance of the previous protocol.

Reference

〔1〕 A. Shamir, "Identity-based cryptosystems and signature schemes," Proc. of Crypto 84, LNCS 196, pp. 47-53, 1984.

〔2〕 M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system," Proc. of EUROCRYPT'94, LNCS 950, pp. 275-286, May 1994.

〔3〕 D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," Journal of Cryptology, vol. 13, no. 3, pp. 361-396, Dec. 2000.

〔4〕 E. Bresson, O. Chevassut, D. Pointcheval, and J. Quisquater, "Provably authenticated group Diffie-Hellman key exchange," 8th ACM conference on Computer and Communications Security (CCS'01), pp. 255-264, Dec. 2001.

〔5〕 D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," Proc. of Crypto'01, LNCS 2139, pp. 213-229, 2001.

〔6〕 J. Katz and M. Yung, "Scalable protocols for authenticated group key exchange," Proc. of Crypto'03, LNCS 2729, pp. 110-125, 2003.

〔7〕 F. Zhang and X. Chen, "Attack on an ID-based authenticated group key agreement scheme from PKC 2004," Information Proceeding Letters, vol. 91, issue. 4, pp. 191-193, Aug. 2004.

〔8〕 K. Choi, J. Hwang, and D. Lee, "Efficient ID-based group key agreement with bilinear maps," Proc. of PKC'04, LNCS 2947, pp. 130-144, 2004.

〔9〕 J. Kim, H. Kim, K. Ha, and K. Yoo, "One round identity-based authenticated conference agreement protocol," Proc. of ECUMN 2004, LNCS 3262, pp. 407-416, 2004.

〔10〕 Y. Shi, G. Chen, and J. Li, "ID-Based one round authenticated group key agreement protocol with bilinear pairings," Proc. of International Conference on Information Technology: Coding and Computing (ITCC'05), pp. 757-761, Apr. 2005.

〔11〕 H. Yoon, J. Cheon, and Y. Kim, "Batch verifications with ID-based signatures," Proc. of ICISC '04, LNCS 3506, pp. 233-248, 2005.

〔12〕 L. Zhou, W. Susilo, and Y. Mu, "Efficient ID-based authenticated group key agreement from bilinear pairings," Proc. of

Mobile Ad-hoc and Sensor Networks (MSN 2006), LNCS 4325, pp. 521-532, 2006.

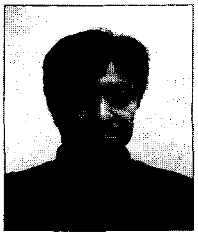
[13] K. Shim, "Further analysis of ID-Based authenticated group key agreement protocol from bilinear maps," IEICE Trans. Fundamentals, vol. E90-A, no. 1, pp. 295-298, Jan. 2007.

[14] K. Choi, J. Hwang, and D. Lee, "ID-Based authenticated group key agreement secure against insider attacks," IEICE Trans. Fundamentals, vol. E91-A, no. 7, pp. 1828-1830, July 2008.

[15] G. Yao, H. Wang, and Q. Jiang, "An authenticated 3-round identity-based group key agreement protocol," Proc. of the third International Conference on Availability, Reliability, and Security (ARES'08), pp. 538-543, Mar. 2008.

〈著者紹介〉

박혜원 (Hyewon Park) 학생회원
2007년 8월: 한국정보통신대학교 (ICU) 전산과 졸업
2007년 8월 ~ 현재: KAIST 정보통신공학과 석사과정
〈관심분야〉 정보보호



Asano Tomoyuki
1991년: 일본 요코하마 국립대학교 전자공학과 졸업
1993년: 일본 요코하마 국립대 대학원 전자공학 석사
1993년 ~ 현재: Sony Corporation 재직 중
2001년 ~ 2002년: Stanford University Visiting scholar
〈관심분야〉 암호 프로토콜

김광조 (Kwangjo Kim) 종신회원
1980년 2월: 연세대학교 공과대학 전자공학과 졸업
1983년 8월: 연세대학교 대학원 전자공학과 석사 (M/W 전공)
1991년 3월: 일본 요코하마 국립대 대학원 전자정보공학 박사 (암호학 및 정보보호 전공)
1979년 12월 ~ 1997년 12월: 한국전자통신연구원 부호1실장/책임연구원
1995년 1월 ~ 1997년 5월: 한국정보통신기술표준협회 일반 보안 기술 실무 작업반 의장
1996년 3월 ~ 1997년 8월: 충남대학교 컴퓨터과학과 겸임교수
1999년 12월 ~ 2000년 2월: 요코하마 국립대 및 동경대 방문 교수
1999년 1월 ~ 2004년 12월: 세계암호학회 이사
1998년 1월 ~ 2009년 2월: 한국정보통신대학교 정보통신대학원장 및 공학부장
2001년 3월 ~ 현재: 국제정보보호기술연구소 소장
2003년 1월 ~ 2005년 1월: IT 영재교육원 원장
2005년 1월 ~ 2008년 12월: Asiacrypt 조정위원회 의장
2005년 2월 ~ 2005년 5월: CSAIL@MIT 방문학자
2005년 6월 ~ 2005년 11월: UCSD 방문교수
2008년 1월 ~ 2008년 12월: 한국정보보호학회 수석부회장
2009년 3월 ~ 현재: 한국과학기술원 전산학과 및 정보통신공학과 교수
2009년 3월 ~ 현재: 한국정보보호학회 회장
〈관심분야〉 정보보호 이론 및 제반 응용