SCIS 2009 The 2009 Symposium on Cryptography and Information Security
Otsu, Japan, Jan. 20-23, 2009
The Institute of Electronics, Information and Communication Engineers

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

Hyewon Park [*] Tomoyuki Asano [†] Kwangjo Kim [‡]

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: ID-Based, GKA, AGKA, constant-round

1 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 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 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 pre-determine 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, 9, 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 2, we review previous ID-based AGKA protocols. After introducing preliminaries in Section 3, we review Zhou *et al.*'s two-round AGKA protocol and suggest how to do impersonation attack by malicious participants in Section 4. We present our improved ID-based AGKA protocol in Section 5, and analyze in Section 6. We finally conclude our paper in Section 7.

2 Related Work

In this section, we briefly review some recent papers about ID-based constant-round 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.

[*] Information and Communications University, Munji-dong, Yuseong-gu, Daejeon, 305-732 Korea, inde1776@icu.ac.kr
[†] System Technologies Laboratories, Sony Corporation, 6-7-35 Kitashinagawa, Shinagawa-ku, Tokyo 141-0001, Japan
[‡] Information and Communications University, Munji-dong, Yuseong-gu, Daejeon, 305-732 Korea, kkj@icu.ac.kr

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.* [15]'s AGKA protocol 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.

3 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 $\mathcal{P} = \{U_1, ..., U_n\}$ 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 $\mathcal{A}$ 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.

- $Execute(U_1, ..., U_n)$: Execute the protocol between the players $U_1, ..., U_n$ 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.

- $Test(U, i)$: $\mathcal{A}$ 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 a 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 $\mathcal{A}$ in attacking protocol is defined as

$$\mathsf{Adv}_{\mathcal{A}}(k) = |2 \cdot Pr[Succ] - 1|,$$

where $Succ$ is the event that $\mathcal{A}$'s guess b' satisfies $b = b'$ for $Test$ query.

The GKA protocol is said to be secure if $\mathsf{Adv}_{\mathcal{A}}(k)$ is negligible for all probabilistic polynomial time (PPT) adversary $\mathcal{A}$.

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 $\mathcal{IG}_{CDH}$ is a PPT algorithm takes a security parameter 1^k and outputs additive group G_1 with an order q.

When an algorithm $\mathcal{A}$ solves CDH problem with an advantage ϵ, the advantage is

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

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

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

When an algorithm $\mathcal{A}$ solves Decisional BDH (DBDH) problem with an advantage ϵ, the advantage is

$$\begin{aligned} &|Pr[\mathcal{A}(P, aP, bP, cP, e(P, P)^{abc}) = 1] \\ &- Pr[\mathcal{A}(P, aP, bP, cP, e(P, P)^{d}) = 1]| \leq \epsilon, \end{aligned}$$

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

4 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 $H : \{0,1\}^* \rightarrow Z_q^*$. Key Generation Center (KGC) chooses a random $s \in Z_q^*$ as the secret master key, and a random generator P of G_1. Then KGC computes $P_{pub} = sP$.

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

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

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: $G_2 \rightarrow \{0,1\}^n$, H_5: $\{0,1\}^n \rightarrow Z_q^*$, and H_6: $G_1 \rightarrow \{0,1\}^n$. 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 $P_i = r \bigoplus H_4(e(S_1, Q_i) \cdot \delta)$ $(2 \leq i \leq n)$
Compute & broadcast D_1

$$D_1 = < \delta, P_2, ..., P_n, \\ X_1 = H_5(r) \cdot k_1 P, Y_1 = k_1 P_{pub}, L >,$$

where L is a label containing users' association information.

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

Find appropriate P_i from D_1.
Then compute $r' = H_4(e(S_i, Q_1) \cdot \delta) \bigoplus P_i = r$
Choose $k_i \leftarrow Z_P^*$ randomly.
Compute & broadcast D_i

$$D_i = < X_i, Y_i > = < H_5(r) \cdot k_i P, k_i P_{pub} >$$

Key Computation. Each user computes

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

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

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

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

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_k 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 $r' = H_4(e(S_m, Q_1) \cdot \delta) \bigoplus P_m = r$
Random $k_i \leftarrow Z_P^*$, $k_m \leftarrow Z_P^*$
Compute & broadcast D_i, D_m
$D_i = < X_i, Y_i > = < H_5(r) \cdot k_i P, k_i P_{pub} >$
$D_m = < X_m, Y_m > = < H_5(r) \cdot k_m P, k_m P_{pub} >$

Key Computation. All users succeed to verify D_i

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

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

In **Round 2** of the protocol, malicious user U_m can compute $< X_i, Y_i >$ 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.

5 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 $P_i = r \bigoplus H_1(e(\delta S_1, Q_i))$ $(2 \leq i \leq n)$

Compute & broadcast D_1

$$D_1 = < \delta, P_2, ..., P_n, X_1 = H_2(r||L) k_1 P, \\ Y_1 = k_1 P_{pub} + H_2(r||L) S_1, L >$$

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

Find appropriate P_i from D_1.
Then compute $r' = H_1(e(\delta S_i, Q_1)) \bigoplus P_i = r$
Choose $k_i \leftarrow Z_q^*$ randomly.
Compute & Broadcast D_i

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

Key Computation. Each user compute

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

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

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

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 to Q_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 $< H_2(r||L) \cdot k_i P, k_i P_{pub} + H_2(r||L)S_i >$ 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.

$$\begin{aligned} & e(P, \textstyle\sum_{j=1}^n Y_j) \\ &\quad = e(P, \textstyle\sum_{j=1}^n (k_j P_{pub} + H_2(r||L)S_j)) \\ &\quad = e(P, \textstyle\sum_{j=1}^n (k_j sP) + \sum_{j=1}^n (H_2(r||L)sQ_j)) \\ &\quad = e(P_{pub}, \textstyle\sum_{j=1}^n (k_j P) + \sum_{j=1}^n (H_2(r||L)Q_j)) \\ &\quad = e(P_{pub}, z + H_2(r||L) \textstyle\sum_{j=1}^n (Q_j)) \end{aligned}$$

6 Analysis of Our Protocol

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

6.1 Security Analysis

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.

1) Encryption

We first assume that an adversary $\mathcal{A}$ gains an advantage from attacking the encryption scheme $r \oplus H_1(e(\delta S_i, Q_j))$ 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:*

$$\mathsf{Adv}_{\mathcal{A}} \leq 2q_{ex} \cdot \mathsf{Adv}_G^{DBDH}.$$

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

$$Real = \left[\begin{array}{r} \delta, k_1 \leftarrow Z_q^*, r \leftarrow \{0,1\}^{|q|}; \\ P_i = r \oplus H_1(e(S_1, \delta Q_i)); \\ r' = H_1(e(S_i, \delta Q_1)) \oplus P_i; \\ X_i = H_2(r||L)k_i P, Y_i = k_i P_{pub} + H_2(r||L)S_i; \\ \mathcal{T} =< \delta, P_2, ..., P_n, X_1, ..., X_n, Y_1, ..., Y_n >; \\ K = H_3(z) \end{array} \right]$$

Consider the distributions *Fake* defined as follows:

$$Fake = \left[\begin{array}{r} \delta, k_1, b_1, b_i \leftarrow Z_q^*, r \leftarrow \{0,1\}^{|q|}; \\ P_i = r \oplus H_1(e(\delta b_1 P_{pub}, b_i P)); \\ r' = H_1(e(b_i P_{pub}, \delta b_1 P)) \oplus P_i; \\ X_i = H_2(r||L)k_i P, Y_i = k_i P_{pub} + H_2(r||L)S_i; \\ \mathcal{T} =< \delta, P_2, ..., P_n, X_1, ..., X_n, Y_1, ..., Y_n >; \\ K = H_3(z) \end{array} \right]$$

Let $\epsilon = \mathsf{Adv}_G^{DBDH}$ and q_{ex} is the number of $Execute$ queries issued by $\mathcal{A}$. When choosing $(\mathcal{T}, K)$ pair randomly to ask $Test$ query and getting b, $\mathcal{A}$ can distinguish $e(S_1, \delta Q_i)$ and $e(b_1 P_{pub}, \delta b_i P)$, and get bit b' from guessing with probability ϵ' $(\leq \epsilon)$ because he can obtain $b_1 P, ..., b_n P$ and $P_{pub} = sP$ is public. Therefore, we obtain the following equation.

$$\begin{aligned} \epsilon' = |&Pr[\mathcal{T} \leftarrow Real; K \leftarrow Real; \mathcal{A}(\mathcal{T}, K) = 1] \\ &Pr[\mathcal{T} \leftarrow Fake; K \leftarrow Fake; \mathcal{A}(\mathcal{T}, K) = 1]| \leq \epsilon \end{aligned}$$

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

$$\begin{aligned} \mathsf{Adv}_{\mathcal{A}} &= 2 \cdot Pr[Succ] - 1 = 2 \cdot Pr[b = b'] - 1 \\ &= 2 \cdot Pr[b = b' | \neg Ask] Pr[\neg Ask] \\ &\quad +2 \cdot Pr[b = b' | Ask] Pr[Ask] - 1 \\ &= 2 \cdot Pr[b = b' | \neg Ask] + 2 \cdot Pr[b = b' | Ask] - 1 \\ &= 2 \cdot Pr[b = b' | Ask] = 2q_{ex} \cdot \epsilon' \end{aligned}$$

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

$$\mathsf{Adv}_{\mathcal{A}} \leq 2q_{ex} \cdot \mathsf{Adv}_G^{DBDH}$$

2) Signature

Second, we assume that $\mathcal{A}$ 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 $k \in Z_q^*$, $h = H_2(r||L)$;
$< kP, Y > \leftarrow \Sigma_{gen}(S_{ID})$.

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

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 $\mathcal{F}_0$ be a forger which performs an existential forgery under an adaptively chosen ID with running time t_0. The forger $\mathcal{F}_0$ 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 $\mathcal{F}_0$ is $\epsilon_0 \geq 10q_H(q_S+1)(q_S+q_{H_2})/(q-1)$. Then there exists an attacker $\mathcal{F}$ that can solve the CDH problem within the expected time $t_2 \leq 120686 q_{H_2} t_0/\epsilon_0$.*

We can prove **Theorem 2** by proving the following Lemmas.

Lemma 1. *Let the hash functions H be random oracle and $\mathcal{F}_0$ be a forger for an adaptively chosen ID with running time t_0 and advantage ϵ_0. Suppose $\mathcal{F}_0$ can ask queries to the H at most q_H times. Then a forger $\mathcal{F}$ 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. $\mathcal{F}$ is given ID^*, and we assume that $\mathcal{F}_0$ makes $H, Extract$, and $Sign$ queries at most once. $\mathcal{F}$ maintains a list L_H of $< ID_i, Q_i >$ and interacts with $\mathcal{F}_0$ after choosing $\alpha \in \{1, ..., q_H\}$.

- When $\mathcal{F}_0$ makes α-th H query on ID, $\mathcal{F}$ returns Q^* with H query for ID^* and inserts $< ID, Q^* >$ into L_H if $ID = ID^*$. Otherwise, $\mathcal{F}$ returns result for ID, and inserts $< ID, Q >$ into L_H.

- $\mathcal{F}_0$ issues an $Extract$ query on Q_i. If $Q_i = Q^*$ then $\mathcal{F}$ outputs FAIL; otherwise, $\mathcal{F}$ returns S_i to $\mathcal{F}_0$ as the result of $Extract$ query.

- When $\mathcal{F}_0$ issues H_2 query on $r||L$, $\mathcal{F}$ returns the result $H_2(r||L)$.

- When $\mathcal{F}_0$ makes $Sign$ query on ID_i, $\mathcal{F}$ returns $< ID_i, kP_i, Y_i >$ to $\mathcal{F}_0$.

- $\mathcal{F}_0$ finally outputs $< ID', k'P, Y' >$ then $\mathcal{F}$ finds $< ID', Q' >$ in L_H. If $Q' = Q^*$, $\mathcal{F}$ outputs $< ID^*, k'P, Y' >$, otherwise it fails.

Here, $\mathcal{F}$ succeeds the simulation with probability $1/q$ if $Q' \neq Q^*$ and $< ID', Q' >$ is not in L_H because the output $< ID', k'P, Y' >$ is independent of the information $\mathcal{F}_0$ accumulated from the previous queries in this case. Therefore, the probability that $\mathcal{F}$ 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 $\mathcal{F}_0$ 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 $\mathcal{F}_0$ 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 $\mathcal{F}$ can solve the CDH problem within expected time $t_2 \leq 120686 q_{H_2} t_1/\epsilon_1$.*

Proof. $\mathcal{F}$ sets the system parameters $\mathsf{param} = < G_1, G_2, e, P, P_{pub}, ID, H, H_2 >$ where $P_{pub} = xP$ and gives it to $\mathcal{F}_0$. Given P, xP, and yP, $\mathcal{F}$'s goal is to compute xyP as CDH problem. $\mathcal{F}$ maintains two lists $L_H =< ID_i, a_i, Q_i >$ and $L_Y =< ID_j, k_jP >$, and interacts with $\mathcal{F}_0$ as follows:

- When $\mathcal{F}_0$ makes H query on ID, $\mathcal{F}$ returns $Q^* = yP$ for ID^*; otherwise $\mathcal{F}$ picks $a_i \in Z_q^*$ randomly, adds $< ID_i, a_i, Q_i >$ to L_H, and returns $Q_i = a_iP$.

- $\mathcal{F}_0$ issues an $Extract$ query on Q_i, if $Q_i = Q^*$ then $\mathcal{F}$ fails; otherwise, $\mathcal{F}$ finds $< ID_i, a_i, Q_i >$ from L_H and returns $S_i = a_iP_{pub} = xQ_i$ to $\mathcal{F}_0$.

- When $\mathcal{F}_0$ issues H_2 query on $r||L$, $\mathcal{F}$ picks $h_i \in Z_q^*$ randomly and returns it.

- When $\mathcal{F}_0$ makes $Sign$ query on ID_i, $\mathcal{F}$ picks $k_i \in Z_q^*$ randomly, computes k_iP, and adds $< ID_i, k_iP >$ to L_Y. Then $\mathcal{F}$ finds $< ID_i, a_i, Q_i >$ from L_H, computes $Y_i = k_ixP + h_ia_ixP = k_iP_{pub} + h_iS_i$ and returns $< ID_i, k_iP, h_i, Y_i >$ to $\mathcal{F}_0$.

Finally, $\mathcal{F}_0$ outputs a valid tuple $< ID^*, kP, h, Y >$ where $< ID^*, kP >$ is not in L_Y without accessing any oracles except H_2. If $\mathcal{F}$ replays with the same random tape but different choices of H_2 as in the *forking lemma* [3], then $\mathcal{F}_0$ outputs two valid tuples

$$< ID^*, kP, h, Y > \text{ and } < ID^*, kP, h', Y' >$$

where $h \neq h'$.

Here, $\mathcal{F}$ can computes $(Y - Y')/(h - h') = xyP$ as CDH problem if both of them are expected; otherwise, it fails. Therefore, the time for $\mathcal{F}$ 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 $\mathcal{F}$ in our protocol is negligible.

6.2 Performance Analysis

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
$\#G_1$-M:	Total number of G_1 group multiplication
$\#G_2$-M:	Total number of G_2 group 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 AGKA protocol can improve both of the security and performance of ZSM-2 protocol.

Table 1: Comparison of Performance

Protocol	[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

7 Conclusion

In this paper, we suggested a deterministic attack on the ZSM-2 protocol that a malicious insider who knows the secret value r 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 r 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.

References

[1] A. Shamir, Identity-based Cryptosystems and Signature Schemes, Proc. of Crypto 84, LNCS 196, pp.47-53, Springer-Verlag, 1984.

[2] M. Burmester and Y. Desmedt, A Secure and Efficient Conference Key Distribution System, Proc. of EUROCRYPT'94, LNCS 950, pp. 275-286. Springer, May 1994.

[3] D. Pointcheval and J. Stern, Security Arguments for Digital Signatures and Blind Signatures, J. of Cryptology, vol. 13, pp. 361-396, 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), pages 255-264. ACM Press, 2001.

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

[6] J. Katz and M. Yung, Scalable Protocols for Authenticated Group Key Exchange, Proc. of Crypto'03, LNCS 2729, pp.110-125, Springer, 2003.

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

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

[9] J. S. Kim, H. C. Kim, K. J. Ha, and K. Y. Yoo, One Round Identity-Based Authenticated Conference Agreement Protocol, Proc. of ECUMN 2004, LNCS 3262, pp.407-416, Springer-Verlag, 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), vol.I, pp.757-761, 2005.

[11] H. Yoon, J. H. Cheon, and Y. Kim, Batch verifications with ID-based signatures, Proc. of ICISC '04, LNCS 3506, pp.233-248, Springer-Verlag, 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, Springer-Verlag, 2006.

[13] K. A. Shim, Further Analysis of ID-Based Authenticated Group Key Agreement Protocol from Bilinear Maps, IEICE Trans. Fundamentals, vol.E90-A, no.1, pp.231-233, 2007.

[14] K. Y. Choi, J. Y. Hwang and D. H. Lee, ID-Based Authenticated Group Key Agreement Secure against Insider Attacks, IEICE Trans. Fundamentals, vol.E91-A, no.7, pp.1828-1830, 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, ACM, 2008.