JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 27, 915-932 (2011)

A Secure ID-Based Authenticated Group Key Exchange Protocol Resistant to Insider Attacks*

TSU-YANG WU, YUH-MIN TSENG AND CHING-WEN YU
Department of Mathematics
National Changhua University of Education
Chang-Hua City, 500 Taiwan

Recently, several identity (ID)-based authenticated group key exchange (IDAGKE) protocols from bilinear pairings were proposed. However, they all suffered from different types of insider (participants) colluding attacks. In this paper, we present a new IDAGKE protocol from bilinear pairings. In the random oracle model and under some security assumptions, we demonstrate that the proposed protocol is a provably secure IDAGKE protocol providing forward secrecy. Meanwhile, it is secure against insider attacks.

Keywords: authenticated group key exchange, identity-based, bilinear pairings, insider attacks, malicious participant

1. INTRODUCTION

An authenticated group key exchange (AGKE) protocol is an important security mechanism which provides secure group communications for members in cooperative and distributed applications. It allows that group members generate a common key, which can be used to encrypt and authenticate communicating messages in a group. Meanwhile, it also provides entity authentication. In the past, many AGKE protocols based on the traditional certificate-based public-key systems were proposed in [1-8].

In 1984, Shamir [9] first proposed the concept of identity (ID)-based public-key system. With compared to the traditional certificate-based public-key systems, Shamir's system may simplify the certificate management. However, it has a disadvantage that the user's private key must be generated by a single Key Generator Center (KGC). In 2001, Boneh and Franklin [10] presented a practical ID-based encryption system from bilinear pairings defined on elliptic curves. The security of their scheme is based on the discrete logarithm problem. In this case, the user's private key can be generated by several sub-centers using a threshold technique. Subsequently, the ID-based cryptographic schemes based on bilinear pairings have received much attention from cryptographic researchers. Recently, many related ID-based cryptographic schemes and protocols from bilinear pairings have been proposed in [11-24].

In 2004, Choi *et al.* [12] first presented an ID-based authenticated group key exchange (IDAGKE) protocol from bilinear pairings. Unfortunately, their protocol suffered from insider (participants) colluding attacks [23, 24]. Note that insider colluding attack means that some participants can collude to impersonate other participant in this session (or other session) by some ways. In [24], Shim also proposed a modification to overcome

Received October 19, 2009; revised May 25 & July 27, 2010; accepted September 14, 2010.
Communicated by Wen-Guey Tzeng.
* This research was partially supported by the National Science Council of Taiwan, R.O.C., under contract No. NSC 97-2221-E-018-010-MY3.

two insider colluding attacks. However, Choi *et al.* [19] proved that Shim's modification is still insecure against the other insider colluding attacks in 2008. They also presented an improvement to resist the mentioned insider colluding attacks. In 2009, we proved that their improvement [19] still suffered from an insider colluding attack in [21]. Meanwhile, these protocols above don't address the problem of malicious participants. Note that, a malicious participant is considered as a legitimate participant, but she/he is fully controlled by an adversary. Fortunately, Katz and Shin [25] presented a security model and a universal composability complier for this issue in authenticated group key exchange (AGKE) protocols. This provides the formal security definition of AGKE protocols in the existence of malicious participants. Adopting the Katz-Shin complier into an AGKE protocol, it can employ the explicit group key confirmation property to prevent malicious participants. However, the complier [25] requires an additional round and $O(n)$ signature verifications for each participant, where n is the number of group members.

In this paper, we propose a robust ID-based authenticated group key exchange (ID-AGKE) protocol from bilinear pairings. Particularly, our protocol is based on the Sakai-Kasahara's key construction [26]. In our proposed protocol, we adopt the concept of the Katz-Shin complier to resist malicious participants. For decreasing the computation cost of signature verifying, we use a batch verification technique called the small exponents test [27] into our protocol. In the random oracle model [28] and under some security assumptions, we will prove that our protocol is a provably secure IDAGKE protocol providing forward secrecy and achieves the Katz-Shin's security model. This means that the proposed protocol is secure against insider attacks and malicious participants disturbing. Finally, we make the comparisons between the previously proposed IDAGKE protocols and our proposed protocol.

The remainder of this paper is organized as follows. In section 2, we briefly review the concepts of bilinear pairings, key constructions, and related mathematical assumptions. The security model and notions of authenticated group key exchange is given in section 3. In section 4, we present our IDAGKE protocol. Security analysis of the proposed protocol is given in section 5. In section 6, we make the performance analysis and comparisons. Our conclusions and future work are drawn in section 7.

2. PRELIMINARIES

In this section, we depict compendiously the concepts of bilinear pairings, key constructions of ID-based public-key system from bilinear pairings, and the related mathematical assumptions.

2.1 Bilinear Pairings

Let G_1 and G_2 be two cyclic groups with a prime order q, where G_1 is additive and G_2 is multiplicative. Here, G_2 is a subgroup of the multiplicative group over a finite field. A bilinear pairing is defined as a map e: $G_1 \times G_1 \rightarrow G_2$. If the map e satisfies the following three properties, it is called an admissible bilinear map:

(1) Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_q^*$.
(2) Non-degenerate: There exists $P, Q \in G_1$ such that $e(P, Q) \neq 1$.

(3) Computable: For all $P, Q \in G_1$, there exists an efficient algorithm to compute the value $e(P, Q)$.

For the details of bilinear pairings, we can refer to [10, 16] for full descriptions.

2.2 Key Constructions

In general, there are two kinds of key constructions for ID-based public-key systems from bilinear pairings. We describe them as below:

- Construction I: In 2000, Sakai *et al.* presented the key Construction I in [29]. In this construction, there exists a special hash function called a map-to-point hash function H_G: $\{0, 1\}^* \rightarrow G_1$ [10]. For a user U with identity ID_U, the user's private key $DID_U = H_G(ID_U) \cdot s$, where s is the private key of Key Generation Center (KGC). Up to now, many ID-based cryptographic schemes and protocols based on this construction were proposed in [10-13, 16, 17, 19, 20, 22].
- Construction II: In 2003, Sakai and Kasahara proposed the key Construction II in [26]. In this construction, there exists a general hash function H: $\{0, 1\}^* \rightarrow Z_q^*$. For a user U with identity ID_U, the user's private key $DID_U = \frac{1}{s + H(ID_U)} \cdot P$, where s is the private key of KGC, and P is a generator of the group G_1. Until now, some ID-based cryptographic schemes based on this construction were proposed in [14, 15]. Note that the expensive map-to-point hash function is not required in this construction.

2.3 Related Mathematical Problems and Assumptions

For convenience to prove the security of our proposed protocol, we summarize some important mathematical hard problems and assumptions for bilinear pairings defined on elliptic curves as follows. For the details of them, we can refer to [15, 16] for full descriptions.

- Decision Diffie-Hellman problem (DDH) in the group G_2: Given $g = e(P, P)$, g^a, g^b, $g^c \in G_2$, for some $a, b, c \in_R Z_q^*$, the DDH problem (in G_2) is to distinguish (g, g^a, g^b, g^{ab}) from (g, g^a, g^b, g^c).
- DDH assumption: No probabilistic polynomial time (PPT) algorithm $\mathcal{A}$ can solve the DDH problem (in G_2) with a non-negligible advantage.
- k-CAA problem: Let k be a positive integer and $s \in_R Z_q^*$. Given $c_1, c_2, \ldots, c_k \in Z_q^*$, $P \in G_1$, $P_{pub} = s \cdot P$, and $\frac{1}{s + c_1} P$, $\frac{1}{s + c_2} P$, …, $\frac{1}{s + c_k} P$, the k-CAA problem is to compute $\frac{1}{s + c} P$, where $c \in Z_q^*$ and $c \notin \{c_1, c_2, \ldots, c_k\}$.
- k-CAA assumption: No PPT algorithm $\mathcal{A}$ can solve the k-CAA problem with a non-negligible advantage.
- Generalized k-CAA (Gk-CAA) problem: Let $k = n \cdot m$ be a positive integer and $s \in_R Z_q^*$. Given $c_1, c_2, \ldots, c_n, d_1, d_2, \ldots, d_m \in Z_q^*$, $P \in G_1$, $P_{pub} = s \cdot P$, $\frac{d_1}{s + c_1} P, \frac{d_2}{s + c_1} P, \ldots, \frac{d_m}{s + c_1} P$, $\frac{d_1}{s + c_2} P$, $\frac{d_2}{s + c_2} P$, …, $\frac{d_m}{s + c_2} P$, …, and $\frac{d_1}{s + c_n} P, \frac{d_2}{s + c_n} P, \ldots, \frac{d_m}{s + c_n} P$, the generalized k-

CAA problem is to compute $\frac{d}{s+c}P$, where $c, d \in Z_q^*$, $c \notin \{c_1, c_2, \ldots, c_n\}$, and $d \notin \{d_1, d_2, \ldots, d_m\}$.

The k-CAA problem and the Gk-CAA problem in the group G_1 are computationally equivalent [15]. It means that Gk-CAA problem is a hard problem in the group G_1.

2.4 Notations

The following notations are used throughout this paper:

- e: an admissible bilinear map, e: $G_1 \times G_1 \rightarrow G_2$.
- ID_i: the identity of participant U_i.
- P: a generator of the group G_1.
- s: the system private key $s \in Z_q^*$.
- P_{pub}: the system public key, $P_{pub} = s \cdot P$.
- g: a public value belongs to the group G_2, $g = e(P, P)$.
- $H()$: a one-way hash function, H: $\{0, 1\}^* \rightarrow Z_q^*$.
- $H_1()$: a one-way hash function, H_1: $\{0, 1\}^* \times G_2 \rightarrow Z_q^*$.
- $H_2()$: a one-way hash function, H_2: $\{0, 1\}^* \times G_2^2 \rightarrow Z_q^*$.

3. SECURITY MODEL AND NOTIONS

In this section, we briefly review the security model and notions for authenticated group key exchange protocol. The following definitions are referred to [2, 12, 25, 30].

Participants and Initialization We assume that each participant U_i has an unique identity $ID_i \in \{0, 1\}^l$, where l is a bit length. For simplicity, there is a fixed set $G = \{U_1, U_2, \ldots, U_n\}$ with polynomial-size of potential participants. Here, we allow that each participant $U_i \in G$ performs the protocol many times with different participants. In other words, each U_i has the unlimited number of instances to execute the AGKE protocol called oracles Π_i^t with $t \in N$. We denote the tth instance oracle of participant U_i as Π_i^t

An ID-based authenticated group key exchange protocol requires the following algorithms:

- Setup algorithm: The system private key s and the public parameters are generated by this algorithm.
- Extract algorithm: Each participant U_i obtains her/his private key DID_i by this algorithm, $DID_i \leftarrow Extract_s(ID_i)$.

Note that, the public parameters and the set of participants' identities $ID = \{ID_1, ID_2, \ldots, ID_n\}$ are known by all participants (including adversary).

Session Group key, Session ID, and Partner ID In each session, a new set G of k participating oracles is considered, where $1 \leq k \leq n$. By g_i for $i \in [1, k]$, we denote the index of the user related to the ith oracle involved in G, where this ith oracle is defined as $\Pi(G,$

i). Thus, for each $i \in [1, k]$ there exists $\Pi(G, i) = \Pi_{g_i}^t \in G$ for some $t \in N$. Every participating oracle $\Pi_U^t \in G$ computes the *session group key* SK_U^t. Each session is identified by a unique *session ID* $SID(\Pi_U^t)$. This value is know to all participating oracles in the same session. Similarly, each oracle $\Pi_U^t \in G$ obtains a *partner ID* $PID(\Pi_u^t)$ that contains the identities of participants (including U), formally $PID(\Pi_U^t) = \{ID_{U_{g_i}} | \Pi(G, i) \in G, \forall i = 1, 2, \ldots, k\}$. Note that the session ID and the partner ID are public information.

Related Notions We say that the oracle Π_i^t is *accepted*, when it can compute a valid session key. An oracle may terminate without accepting. In this case, it does not output any session key to all. Whether or no, an oracle has accepted or decided to terminate without accepting is a public information. We say that two oracles Π_i^t and Π_j^s are *partnered* if and only if it satisfies following three conditions: (1) Two oracles have accepted; (2) $ID_{U_i} \in PID(\Pi_j^s)$ and $ID_{U_j} \in PID(\Pi_i^t)$; (3) $SID(\Pi_i^t) = SID(\Pi_j^s)$.

Adversarial Model Formally, an adversary $\mathcal{A}$ is a probabilistic polynomial-time algorithm. We allow $\mathcal{A}$ to potentially control all communications completely in ID-based authenticated group key exchange (IDAGKE) protocol and may interact with the simulated group participants (*i.e.*, a challenger) by issuing the following oracle queries:

- Execute query: When the adversary $\mathcal{A}$ issues this query on *ID* to the oracles, the oracle Π_i^t returns the entire transcripts of an honest execution between the participants belong to *ID*. Note that $\mathcal{A}$ can choose the number of participants.
- Extract query: When the adversary $\mathcal{A}$ issues this query on ID_α to the oracles, the oracle Π_i^t returns the corresponding private key DID_α. In particular, ID_α is a non-target identity, $ID_\alpha \notin ID$.
- Send query: When the adversary $\mathcal{A}$ issues this query on messages to the oracles, the oracle Π_i^t returns the response according to the steps of the IDAGKE protocol.
- Reveal query: In this query, the adversary $\mathcal{A}$ can get a group session key from the oracle Π_i^t.
- Corrupt query: In this query, the adversary $\mathcal{A}$ can get a private key DID_i of ID_i from the oracle Π_i^t.
- Test query: The adversary $\mathcal{A}$ can issues a single *Test query* to a fresh oracle Π_i^t upon receiving this query, the oracle Π_i^t flips an unbiased coin b. If $b = 1$, Π_i^t returns the group session key. Otherwise, Π_i^t returns a random string.

In the above model, there are two types of adversaries according to their queries. **A passive adversary** is allowed to make *Execute*, *Reveal*, *Corrupt*, and *Test queries*; **an active adversary** is additionally to make *Extract* and *Send queries*. Here, we also use the *Execute query* to get more precise analysis, even though this query can be substituted for making the *Send query* repeatedly.

Freshness An oracle Π_i^t is called *fresh*, if the following conditions hold: (1) Π_i^t has accepted; (2) Nobody has been asked for a *Corrupt query* before Π_i^t accepts; (3) Neither Π_i^t nor its partners have been issued a *Reveal query*. Here, we assume that all oracles are considered *fresh*.

Secure AGKE A secure authenticated group key exchange protocol contains following four parts:

(1) Freshness: As mentioned above.
(2) The security of AGKE protocol: The security of ID-based authenticated group key exchange (IDAGKE) protocol is defined in the following game played with an active adversary $\mathcal{A}$ and an infinite set of oracles Π_i^t, where Π_i^t is the oracle of participant $U_i \in G$.
   1. The system private key, the public parameters, and the participants' private keys are generated in the initialization phase.
   2. The adversary $\mathcal{A}$ can issue some queries and gets back the answers corresponding to oracles.
   3. Finally, the adversary $\mathcal{A}$ outputs its guess b' for the coin b in the *Test* query and terminates.

   In the above game, the advantage of $\mathcal{A}$ is defined by the ability to distinguish the group session key from a random string. Let *Succ* be the event that $\mathcal{A}$ queries the *Test* oracle and correctly guesses the coin b, where the coin is used to answer the *Test query* in the *Test* oracle. The advantage of the adversary is defined as $Adv(k) = |2 \cdot \Pr[Succ] - 1|$. We say that an IDAGKE protocol is secure, if the advantage $Adv(k)$ is negligible for any adversary.
(3) Forward secrecy: We say that an ID-based authenticated group key exchange protocol Ψ provides *forward secrecy*. It means that the previous establishing session group keys in Ψ is preserved if an adversary $\mathcal{A}$ obtains private keys of the protocol participants in later protocol sessions. We define the advantage of $\mathcal{A}$ to attack Ψ running in time t by $Adv_{\Psi}^{AGKE\text{-}fs}(t, q_{ex}, q_s)$, where q_{ex} and q_s are the maximum numbers of issuing the *Execute* and *Send queries*, respectively.
(4) Authentication: We say that an ID-based AGKE protocol provides the *implicit key authentication*. If participants are assured that no one other than its partners can learn the value of a particular session key. In particular, any adversary should not learn the key. Note that, the property of implicit key authentication does not guarantee that partners have actually obtained the key.

In 2005, Katz and Shin [25] defined the concept of insider attacks for authenticated group key exchange protocol. In the following, we briefly review the definitions of insider attacks in the following.

Insider Attacks An ID-based authenticated group key exchange (IDAGKE) protocol is secure against *insider attacks*, if it satisfies following three conditions:

(1) Secure AGKE: As mentioned above.
(2) Secure against insider impersonation attack: Firstly, we say that an adversary $\mathcal{A}$ succeeds in an insider (participant) impersonation attack, if there exist a participant U_j and an instance Π_i^t such that (1) $\mathcal{A}$ impersonates the (uncorrupted) participant U_j to (accepting) the instance Π_i^t; (2) Neither U_i nor U_j are corrupted at the oracle Π_i^t accepts. We say an IDAGKE protocol secure against *insider impersonation attack* if the advantage of any probabilistic polynomial-time adversary succeeds in the above attack is negligible.

(3) Key agreement: We say that an IDAGKE protocol does not provide *key agreement*, if there exist two partnered instances Π_i^t and Π_j^s satisfying (1) Neither U_i nor U_j are corrupted; (2) Two group session keys $SK_i^t \neq SK_j^s$, where SK_i^t and SK_j^s are generated by two oracles Π_i^t and Π_j^s, respectively.

Malicious Participant We say that a participant U_m is a malicious participant in ID-based authenticated group key exchange protocol, if U_m is a legitimate participant in the protocol, nevertheless, she/he is fully controlled by a adversary.

Remark 1: If an ID-based authenticated group key exchange (IDAGKE) protocol is secure against insider attacks, then the IDAGKE protocol is able to resist malicious participant. Because of the condition "*key agreement*" considers explicit key confirmation property and the condition "*secure against insider impersonation attack*" concerns with mutual authentication property. For a malicious participant, it renders toothless in the protocol.

4. PROPOSED PROTOCOL

In this section, we propose a new ID-based authenticated group key exchange (IDAGKE) protocol with resisting insider attacks. In other words, the proposed IDAGKE protocol can withstand malicious participants. At first, we present the initialization phase of our protocol. The Key Generation Center (KGC) executes the *Setup algorithm* to generate the system private key s and the public parameters $Param = \{e, G_1, G_2, q, P, P_{pub}, g, H, H_1, H_2\}$, where $g = e(P, P)$. When a participant U with the identity ID_U wants to obtain her/his private key DID_U, U submits its identity ID_U to the KGC. Upon receiving the request of the participant U, the KGC runs the *Extract algorithm* to compute $DID_U = \frac{1}{s + H(ID_U)} \cdot P$ and returns it to U via a secure channel.

Let $\{U_1, U_2, \ldots, U_n\}$ be a set of participants who want to establish a group session key SK. Note that the indices are taken in a cyclic, *i.e.*, U_{n+1} is U_1 and U_0 is U_n. Assume that each participant U_i has a unique identity ID_i, U_i's public/private key is $(ID_i, DID_i = \frac{1}{s + H(ID_i)} P)$, PID is the concatenation of the identities of participants taking part in a session, *i.e.* $PID = ID_1 \| ID_2 \| \ldots \| ID_n$, and $M \in \{0, 1\}^*$ is a pre-known message by all participants which contains some conference information such as the conference title, date, and location. For authentication, we adopt Cui *et al.*'s ID-based signature scheme [15] into the proposed protocol. In addition, we also use the small exponents test [27] to strengthen the security of batch verification. The details of our protocol are described as follows,

Round 1 Each participant U_i picks a random number $a_i \in Z_q^*$ and computes $r_i = g^{a_i}$, $h_i = H_1(ID_i, M, PID, r_i)$, and $\sigma_i = (a_i + h_i) \cdot DID_i$. Then, each U_i broadcasts (ID_i, r_i, σ_i) to all other participants and keeps a_i as secret.

Round 2 Upon receiving $(ID_{i-1}, r_{i-1}, \sigma_{i-1})$ and $(ID_{i+1}, r_{i+1}, \sigma_{i+1})$, each participant U_i verifies them independently by the following equation. For $j \in \{i - 1, i + 1\}$,

$$g^{h_j} \cdot r_i \stackrel{?}{=} e(P_{pub} + H(ID_j) \cdot P, \sigma_j),$$

where $h_j = H_1(ID_j, M, PID, r_j)$. If the two verifications hold, each U_i uses r_{i-1} and r_{i+1} to compute $D_i = (r_{i+1}/r_{i-1})^{a_i}$. Note that the indices are taken in a cyclic, *i.e.* U_{n+1} is U_1 and U_0 is U_n. Thus, $D_1 = (r_2/r_n)^{a_1}$ and $D_n = (r_1/r_{n-1})^{a_n}$.

Then, each U_i chooses a random number $b_i \in Z_q^*$ and computes $t_i = g^{b_i}$, $k_i = H_2(ID_i, PID, R_1, D_i, t_i)$, and $\rho_i = (b_i + k_i)DID_i$, where $R_1 = r_1 \| r_2 \| \ldots \| r_n$. Finally, U_i broadcasts (ID_i, D_i, t_i, ρ_i) to all other participants.

Round 3 Upon receiving all (ID_j, D_j, t_j, ρ_j) for $j = 1, 2, \ldots, n$ and $j \neq i$, each participant U_i first selects a random vector $(v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$ with the security level $l_b = 160$ bits elements from Z_q and verifies them by the following equation:

$$g^{\sum_{j=1, j\neq i}^{n} v_j \cdot k_j} \cdot \prod_{j=1, j\neq i}^{n} t_j^{v_j} \stackrel{?}{=} e(P_{pub}, \sum_{j=1, j\neq i}^{n} (v_j \cdot \rho_j)) \cdot e(P, \sum_{j=1, j\neq i}^{n} (v_j \cdot H(ID_j) \cdot \rho_j)),$$

where $k_j = H_2(ID_j, PID, R_1, D_j, t_j)$ and $R_1 = r_1 \| r_2 \| \ldots \| r_n$. If the batch verification holds, each U_i computes the group session key

$$SK_i = (r_{i-1})^{na_i} \cdot D_i^{n-1} \cdot D_{i+1}^{n-2} \cdot \cdots \cdot D_n^{i-1} \cdot D_1^{i-2} \cdot \cdots \cdot D_{i-2}.$$

Then, each U_i selects a random number $c_i \in Z_q^*$ and computes $u_i = g^{c_i}$, $w_i = H_2(ID_i, PID, R_2, SK_i, u_i)$, and $\tau_i = (c_i + w_i) \cdot DID_i$, where $R_2 = D_1 \| D_2 \| \ldots \| D_n$. Finally, U_i broadcasts (ID_i, u_i, τ_i) to all other participants.

Key Confirmation Upon receiving all (ID_j, u_j, τ_j) for $j = 1, 2, \ldots, n$ and $j \neq i$, each participant U_i first selects a random vector $(v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$ with the security level $l_b = 160$ bits elements from Z_q and verifies them by the following equation:

$$g^{\sum_{j=1, j\neq i}^{n} v_j \cdot w_j} \cdot \prod_{j=1, j\neq i}^{n} u_j^{v_j} \stackrel{?}{=} e(P_{pub}, \sum_{j=1, j\neq i}^{n} (v_j \cdot \tau_j)) \cdot e(P, \sum_{j=1, j\neq i}^{n} (v_j \cdot H(ID_j) \cdot \tau_j)),$$

where $w_j = H_2(ID_j, PID, R_2, SK_i, u_j)$ and $R_2 = D_1 \| D_2 \| \ldots \| D_n$. If the batch verification holds, each U_i can confirm that her/his SK_i is equal to all SK_j for $j = 1, 2, \ldots, n$ and $j \neq i$. In this case, no malicious participant exists in this group key establishment. All participants can use the group session key SK_i to communicate securely. Otherwise, there exists at least one malicious participant in the group.

Remark 2: After running the proposed protocol, if no malicious participants exist in the group, all participants can compute the same group session key. Here, we demonstrate that all the computed keys $SK_i = (r_{i-1})^{na_i} \cdot D_i^{n-1} \cdot D_{i+1}^{n-2} \cdot \cdots \cdot D_{i-2}$ are identical for $i = 1, 2, \ldots, n$. Set $A_{i-1} = (r_{i-1})^{a_i} = g^{a_{i-1}a_i}$, $A_i = (r_{i-1})^{a_i} \cdot D_i = g^{a_i a_{i+1}}$, $A_{i+1} = (r_{i-1})^{a_i} \cdot D_i \cdot D_{i+1} = g^{a_{i+1}a_{i+2}}$, …, *etc.*, where $r_i = g^{a_i}$, $D_i = (r_{i+1}/r_{i-1})^{a_i}$, $g = e(P, P)$. Thus, we can obtain $SK_i = A_{i-1} \cdot A_i \cdot A_{i+1} \cdot \cdots \cdot A_{i-2}$. Indeed, all participants can compute the same group session key $SK = e(P, P)^{a_1a_2+a_2a_3+\ldots+a_na_1}$.

5. SECURITY ANALYSIS

In this section, we prove that proposed protocol can achieve the security requirements defined in section 3 in the random oracle model [28] and under some security assumptions defined in section 2.3.

5.1 Insider Impersonation Attack

Here, we will prove that the proposed protocol is secure against insider (participants) impersonate attack. Firstly, we want to show that the adopted batch verification in Round 3 and Key confirmation is secure in the following lemma. Note that for the security proof of Lemma 1 we use the similar way in [27].

Lemma 1 For the security level l_b = 160 bits, the batch verification of our proposed protocol in Round 3 and Key confirmation is secure. Precisely, the probability of accepting an invalid batch signature is only $2^{-l_b} = 2^{-160}$.

Proof: At first we prove that the batch verification in Round 3 is secure. It is easy to see that if $g^{k_j} \cdot t_j = e(P_{pub}, \rho_j) \cdot e(P, H(ID_j) \cdot \rho_j)$ holds for all $j \in [1, n]\backslash i$, then $g^{\sum_{j=1, j\neq i}^{n} v_j \cdot k_j} \cdot \prod_{j=1, j\neq i}^{n} t_j^{v_j} = e(P_{pub}, \sum_{j=1, j\neq i}^{n} (v_j \cdot \rho_j)) \cdot e(P, \sum_{j=1, j\neq i}^{n} (v_j \cdot H(ID_j) \cdot \rho_j))$ holds for any vector $(v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n) \in Z_q$. Without of loss generality, we let $j = 1, 2, \ldots, n - 1$. Now, we must consider the other direction that if the batch verification holds, then $g^{k_j} \cdot t_j = e(P_{pub}, \rho_j) \cdot e(P, H(ID_j) \cdot \rho)$ holds for all $j \in [1, n - 1]$ except with probability at most 2^{-l_b}.

Set $X_j = g^{k_j} \cdot t_j$ and $Y_j = e(P_{pub}, \rho_j) \cdot e(P, H(ID_j) \cdot \rho_j)$. Since X_j and Y_j are in the group G_2, we have $X_j = e(P, P)^{x_j}$ and $Y_j = e(P, P)^{y_j}$ for some $x_j, y_j \in Z_q$. Thus, the batch verification can be written as $\prod_{j=1}^{n-1} e(P, P)^{x_j} = \prod_{j=1}^{n-1} e(P, P)^{y_j}$. It implies $e(P, P)^{\sum_{j=1}^{n-1} z_j} = 1$, where $z_j = x_j - y_j$. Because the batch verification holds, the following equation must be true $\sum_{j=1}^{n-1} v_j \cdot z_j \equiv 0 \bmod q$ for the vector $(v_1, v_2, \ldots, v_{n-1}) \in Z_q$.

Assume that at least one individual equation does not hold. Without of loss generality, we let the first equation does not hold, *i.e.*, $j = 1$. This means that the value $z_1 \neq 0$. Since q is a prime, we can find an inverse α_1 of z_1 such that $z_1 \cdot \alpha_1 \equiv 1 \bmod q$. Hence, we have $v_1 \equiv -\alpha_1 \cdot \sum_{j=2}^{n-1} v_j \cdot z_j \bmod q$. Let E be the event that $X_1 \neq Y_1$ but the batch verification holds. Then, we can get a result that given a fixed vector $(v_2, \ldots, v_{n-1})$, there exists exactly one value v_1 such that the event E occurs. Therefore, the probability of E with v_1 is $\Pr[E \mid \Delta'] = 2^{-l_b}$, where $\Delta' = (v_2, \ldots, v_{n-1})$. If we select v_1 at random and the sum of all possible values of $(v_2, \ldots, v_{n-1})$, we obtain $\Pr[E] \leq \sum_{j=1}^{|\Delta'|} (\Pr[E \mid \Delta'] \cdot \Pr[\Delta'])$, where $|\Delta'|$ be the number of all possible values for Δ'. That is $\Pr[E] \leq \sum_{j=1}^{2^{l_b(n-2)}} (2^{-l_b} \cdot 2^{-l_b(n-2)}) = 2^{-l_b}$, where l_b = 160 bits. Following the similar method, the batch verification in Key confirmation is also secure. ❑

Then, combining Lemma 1 and the security of Cui *et al.*'s single signature [15] we

prove that the proposed protocol is secure against insider (participants) impersonation attack in Theorem 2.

Theorem 2 Under the generalized *k*-CAA assumption and the security level l_b, the proposed ID-based authenticated group key exchange (IDAGKE) protocol is secure against insider impersonation attack.

Proof: Firstly, we have known that Cui *et al.*'s single signature is secure against forgery attack under the generalized *k*-CAA assumption [15]. Since we only verify the single signature in Round 2, we can claim that Round 2 of our protocol is secure against insider impersonation attack. In Round 3 and Key confirmation, we use the batch verification technique to reduce the computational cost. By Lemma 1, we have proven that the batch verification of our protocol in Round 3 and Key confirmation is secure against forgery attack under the security level $l_b = 160$ bits. The probability of accepting an invalid signature is $2^{-l_b} = 2^{-160}$. In other words, even if $(n - 1)$ insider participants collude to impersonate other participant in the group key establishment, the generated batch signature tuple is still unable to pass the batch verification equation in Round 3 and Key confirmation. Therefore, the proposed IDAGKE protocol is secure against insider impersonation attack. ❑

5.2 Secure AGKE Providing Forward Secrecy

Now, we want to prove that the proposed ID-based authenticated group key exchange (IDAGKE) protocol is a secure AGKE protocol providing forward secrecy. Firstly, we focus on the following theorem. Note that we use the similar way in [12, 31] to prove Theorem 3.

Theorem 3 Assume that the hash functions H, H_1, and H_2 are random oracles. Then, the proposed IDAGKE protocol Ψ is a secure AGKE providing forward secrecy under the *k*-CAA and decision Diffie-Hellman (DDH) in the group G_2 assumptions. Precisely,

$$Adv_{\Psi}^{AGKE\text{-}fs}(t, q_{ex}, q_s) \le 2nq_{ex} \cdot Adv_{G_2}^{DDH}(t) + Adv_{\Phi}^{forge}(t),$$

where $Adv_{\Phi}^{forge}(t)$ is the advantage of any forger attacking the adopted signature scheme Φ and $Adv_{G_2}^{DDH}(t)$ is the advantage of solving the *DDH* problem in the group G_2.

Proof: Assume that $\mathcal{A}_2$ is an active adversary in attacking the proposed protocol Ψ with a non-negligible advantage. Here, we consider two possible attacking cases. One is that $\mathcal{A}_2$ can impersonate a participant (forging authentication transcripts). The other is that $\mathcal{A}_2$ breaks the protocol Ψ without modifying any transcripts with the advantage.

Case 1: We assume that $\mathcal{A}_2$ has an adaptive impersonation ability to break Ψ. Using the adversary $\mathcal{A}_2$, we can construct a forger $\mathcal{F}$ who generates a valid signature pair (ID, r, σ) with respect to the adopted signature scheme Φ in the following. The forger $\mathcal{F}$ generates all other public and private keys for the system. $\mathcal{F}$ simulates the oracle queries of $\mathcal{A}_2$. This simulation is perfect indistinguishable from $\mathcal{A}_2$'s oracle queries unless $\mathcal{A}_2$ makes the *Corrupt*(*ID*) query. If it occurs, then the forger $\mathcal{F}$ terminates. Otherwise, if $\mathcal{A}_2$ outputs a valid

signature pair (ID, r, σ), then $\mathcal{F}$ generates the $\mathcal{A}_2$'s pair (ID, r, σ). Let *Forge* be the event that $\mathcal{A}_2$ can output a valid signature pair and Pr[*Forge*] be the corresponding probability of this event. Then, the probability of $\mathcal{F}$ which successfully generates a valid signature pair satisfies $\Pr[Forge] \le Adv_{\mathcal{F}, \Phi}^{forge}(t) \le Adv_{\Phi}^{forge}(t)$.

Case 2: We assume that the adversary $\mathcal{A}_2$ breaks the proposed protocol Ψ without modifying any transcripts. Let n be the number of participants chosen by $\mathcal{A}_2$. Considering the case, the adversary $\mathcal{A}_2$ makes the *Execute*(ID_1, ID_2, …, ID_n) query once. The real execution of Ψ is given by

$$Param = \left\{ \begin{array}{l} (G_1, G_2, e) \leftarrow \mathrm{KGC};\ P \leftarrow G_1;\ s \leftarrow Z_q^*;\ P_{pub} = s \cdot P;\ g = e(P, P); \\ DID_1 = \dfrac{1}{s + H(ID_1)} P, \ldots, DID_n = \dfrac{1}{s + H(ID_n)} P : (G_1, G_2, e, P, P_{pub}, g) \end{array} \right\}$$

and

$$Real = \left\{ \begin{array}{l} a_1, \ldots, a_n, h_1, \ldots, h_n, b_1, \ldots, b_n, k_1, \ldots, k_n, c_1, \ldots, c_n, w_1, \ldots, w_n \leftarrow Z_q^*; \\ r_1 = g^{a_1}, \ldots, r_n = g^{a_n};\ \sigma_1 = (a_1 + h_1) DID_1, \ldots, \sigma_n = (a_n + h_n) DID_n; \\ D_1 = (r_2 / r_n)^{a_1},\ D_2 = (r_3 / r_1)^{a_2}, \ldots, D_n = (r_1 / r_{n-1})^{a_n}; \\ t_1 = g^{b_1}, \ldots, t_n = g^{b_n};\ \rho_1 = (b_1 + k_1) DID_1, \ldots, \rho_n = (b_n + k_n) DID_n; \\ u_1 = g^{c_1}, \ldots, u_n = g^{c_n};\ \tau_1 = (c_1 + w_1) DID_1, \ldots, \tau_n = (c_n + w_n) DID_n; \\ \mathsf{T} = (r_1, \ldots, r_n, \sigma_1, \ldots, \sigma_n, D_1, \ldots, D_n, t_1, \ldots, t_n, \rho_1, \ldots, \rho_n, u_1, \ldots, u_n, \tau_1, \ldots, \tau_n); \\ SK = (r_n)^{na_1} \cdot D_1^{n-1} \cdot D_2^{n-2} \cdot \cdots \cdot D_{n-1} : (\mathsf{T}, SK) \end{array} \right\},$$

where T denotes the transcript and SK is the group session key.

Note that $D_i = (r_{i+1}/r_{i-1})^{a_i} = \dfrac{e(a_i a_{i+1} P, P)}{e(a_{i-1} a_i P, P)}$. Here, we use a value $d_{1,2} \in Z_q^*$ to substitute the value $a_1 \cdot a_2$ and define a distribution $Fake_1$ as follows,

$$Fake_1 = \left\{ \begin{array}{l} d_{1,2}, a_1, \ldots, a_n, h_1, \ldots, h_n, b_1, \ldots, b_n, k_1, \ldots, k_n, c_1, \ldots, c_n, w_1, \ldots, w_n \leftarrow Z_q^*; \\ r_1 = g^{a_1}, \ldots, r_n = g^{a_n};\ \sigma_1 = (a_1 + h_1) DID_1, \ldots, \sigma_n = (a_n + h_n) DID_n; \\ D_1 = \dfrac{e(d_{1,2} P, P)}{e(a_n a_1 P, P)},\ D_2 = \dfrac{e(a_2 a_3 P, P)}{e(d_{1,2} P, P)}, \ldots, D_n = \dfrac{e(a_n a_1 P, P)}{e(a_{n-1} a_n P, P)}; \\ t_1 = g^{b_1}, \ldots, t_n = g^{b_n};\ \rho_1 = (b_1 + k_1) DID_1, \ldots, \rho_n = (b_n + k_n) DID_n; \\ u_1 = g^{c_1}, \ldots, u_n = g^{c_n};\ \tau_1 = (c_1 + w_1) DID_1, \ldots, \tau_n = (c_n + w_n) DID_n; \\ \mathsf{T} = (r_1, \ldots, r_n, \sigma_1, \ldots, \sigma_n, D_1, \ldots, D_n, t_1, \ldots, t_n, \rho_1, \ldots, \rho_n, u_1, \ldots, u_n, \tau_1, \ldots, \tau_n); \\ SK = (r_n)^{na_1} \cdot D_1^{n-1} \cdot D_2^{n-2} \cdot \cdots \cdot D_{n-1} : (\mathsf{T}, SK) \end{array} \right\}.$$

Since $\mathcal{A}_2$ can obtain all private keys DID_i and hash values h_i, k_i, and w_i by making the *Corrupt* and *Hash* queries, $\mathcal{A}_2$ can compute all $a_i DID_i = \sigma_i - h_i DID_i$, $b_i DID_i = \rho_i - k_i DID_i$, and $c_i DID_i = \tau_i - w_i DID_i$ for $i = 1, 2, \ldots, n$. By the modified k-CAA problem, we know that

these values offer no information about a_i, b_i, and c_i for $i = 1, 2, \ldots, n$.

In the following Claim, we want to prove that the problem of distinguishing two distributions *Real* from $Fake_1$ can be reduced to solve the decision Diffie-Hellman (DDH) problem in the group G_2. Here, we let $\varepsilon(t) = Adv_{G_2}^{DDH}(t)$.

Claim For any algorithm $\mathcal{A}_2$ running in time t, we have

$$|\Pr[(\mathsf{T}, SK) \leftarrow Real: \mathcal{A}_2(\mathsf{T}, SK) = 1] - \Pr[(\mathsf{T}, SK) \leftarrow Fake_1: \mathcal{A}_2(\mathsf{T}, SK) = 1]| \le \varepsilon(t).$$

Proof: Note that $D_i = (r_{i+1}/r_{i-1})^{a_i} = \dfrac{e(a_i a_{i+1}P, P)}{e(a_{i-1}a_i P, P)} = \dfrac{e(P, P)^{a_i a_{i+1}}}{e(P, P)^{a_{i-1}a_i}}$. We use the symbol $\Gamma_{i,i+1}$ to substitute $e(P, P)^{a_i a_{i+1}}$. Hence, each D_i can be rewritten into $\dfrac{\Gamma_{i,i+1}}{\Gamma_{i-1,i}}$ and the group session key $SK = (r_n)^{na_1} \cdot D_1^{n-1} \cdot D_2^{n-2} \cdot \cdots \cdot D_{n-1}$ can be transformed into $(\Gamma_{n,1})^n \cdot D_1^{n-1} \cdot D_n^{n-2} \cdot \cdots \cdot D_{n-1}$, where $(r_n)^{na_1} = e(P, P)^{na_n a_1} = (\Gamma_{n,1})^n$.

Here, we use a technique to dispose related parameters, which are used to solve the DDH problem in G_2. Giving the adversary $\mathcal{A}_2$ and considering the following algorithm $\mathcal{D}$ which takes $r_a = g^a = e(P, P)^a$, $r_b = g^b = e(P, P)^b$, $r_c = g^c = e(P, P)^c \in G_2$ as input for some $a, b, c \in Z_q^*$. $\mathcal{D}$ generates (T, SK) according to the distribution $Dist^1$ and outputs whatever $\mathcal{A}_2$ outputs after running $\mathcal{A}_2(\mathsf{T}, SK)$. The distribution $Dist^1$ is defined as follows,

$$Dist^1 = \left\{ \begin{array}{l} \alpha_1, \ldots, \alpha_{n-2}, a_1, \ldots, a_n, h_1, \ldots, h_n, b_1, \ldots, b_n, k_1, \ldots, k_n, c_1, \ldots, c_n, w_1, \ldots, w_n \leftarrow Z_q^*; \\ r_1 = g^{a_1}, \ldots, r_n = g^{a_n}; \sigma_1 = (a_1 + h_1)DID_1, \ldots, \sigma_n = (a_n + h_n)DID_n; \\ \Gamma_{1,2} = g_{ab} \in G_2, \Gamma_{2,3} = (r_b)^{\alpha_1} \text{ for } j = 3 \text{ to } n - 1: \\ \Gamma_{j,j+1} = e(P, P)^{\alpha_{j-2}\alpha_{j-1}}, \Gamma_{n,1} = (r_a)^{\alpha_{n-2}}; D_1 = \dfrac{\Gamma_{1,2}}{\Gamma_{n,1}}, D_2 = \dfrac{\Gamma_{2,3}}{\Gamma_{1,2}}, \ldots, D_n = \dfrac{\Gamma_{n,1}}{\Gamma_{n-1,n}}; \\ t_1 = g^{b_1}, \ldots, t_n = g^{b_n}; \rho_1 = (b_1 + k_1)DID_1, \ldots, \rho_n = (b_n + k_n)DID_n; \\ u_1 = g^{c_1}, \ldots, u_n = g^{c_n}; \tau_1 = (c_1 + w_1)DID_1, \ldots, \tau_n = (c_n + w_n)DID_n; \\ \mathsf{T} = (r_1, \ldots, r_n, \sigma_1, \ldots, \sigma_n, D_1, \ldots, D_n, t_1, \ldots, t_n, \rho_1, \ldots, \rho_n, u_1, \ldots, u_n, \tau_1, \ldots, \tau_n); \\ SK = (\Gamma_{n,1})^n \cdot D_1^{n-1} \cdot D_2^{n-2} \cdot \cdots \cdot D_{n-1} : (\mathsf{T}, SK) \end{array} \right\}.$$

By the above distribution, let $r_a = e(P, P)^a$, $r_b = e(P, P)^b$, and $\Gamma_{1,2} = e(P, P)^{ab}$, we refer to the resulting distribution as $Dist^1_{DDH}$. It is obvious that the distribution $Dist^1_{DDH}$ is identical to *Real* as follows,

$$\begin{aligned} SK &= (\Gamma_{n,1})(\Gamma_{1,2})(\Gamma_{2,3}) \cdots (\Gamma_{n-2,n-1})(\Gamma_{n-1,n}) \\ &= e(P, P)^{\alpha_{n-2}a} \cdot e(P, P)^{ab} \cdot e(P, P)^{b\alpha_1} \cdot \cdots \cdot e(P, P)^{\alpha_{n-4}\alpha_{n-3}} \cdot e(P, P)^{\alpha_{n-3}\alpha_{n-2}} \\ &= e(P, P)^{\alpha_{n-2}a + ab + b\alpha_1 + \cdots + \alpha_{n-4}\alpha_{n-3} + \alpha_{n-3}\alpha_{n-2}}. \end{aligned}$$

Next, we again examine the distribution $Dist^1$. Let $r_c = e(P, P)^c = \Gamma_{1,2}$ for some $c \in Z_q^* \backslash \{ab\}$. We refer to the resulting distribution as $Dist^1_{\text{random}}$. It is obvious that the distribution $Dist^1_{\text{random}}$ is identical to $Fake_1$ as follows,

$$\begin{aligned} SK &= (\Gamma_{n,1})(\Gamma_{1,2})(\Gamma_{2,3})\cdots(\Gamma_{n-2,n-1})(\Gamma_{n-1,n}) \\ &= e(aP, P)^{\alpha_{n-2}a} \cdot e(P, P)^{c} \cdot e(P, P)^{b\alpha_1} \cdot \cdots \cdot e(P, P)^{\alpha_{n-4}\alpha_{n-3}} \cdot e(P, P)^{\alpha_{n-3}\alpha_{n-2}} \\ &= e(P, P)^{\alpha_{n-2}a+c+b\alpha_1+\cdots+\alpha_{n-4}\alpha_{n-3}+\alpha_{n-3}\alpha_{n-2}}. \end{aligned}$$

Therefore, we have

$$|\Pr[(\mathsf{T}, SK) \leftarrow Real: \mathcal{A}_2(\mathsf{T}, SK) = 1] - \Pr[(\mathsf{T}, SK) \leftarrow Fake_1: \mathcal{A}_2(\mathsf{T}, SK) = 1]| \le \varepsilon(t). \quad \square$$

By the same approach, other distributions $Fake_i$ can be defined for $i = 2, 3, \ldots, n$. Using the same construction of $Dist^1$, for any adversary $\mathcal{A}_2$ running in time t, we can get the following equations:

$$\begin{aligned} &|\Pr[(\mathsf{T}, SK) \leftarrow Fake_1: \mathcal{A}_2(\mathsf{T}, SK) = 1] - \Pr[(\mathsf{T}, SK) \leftarrow Fake_2: \mathcal{A}_2(\mathsf{T}, SK) = 1]| \le \varepsilon(t), \\ &\vdots \\ &|\Pr[(\mathsf{T}, SK) \leftarrow Fake_{n-1}: \mathcal{A}_2(\mathsf{T}, SK) = 1] - \Pr[(\mathsf{T}, SK) \leftarrow Fake_n: \mathcal{A}_2(\mathsf{T}, SK) = 1]| \le \varepsilon(t). \end{aligned}$$

This implies

$$|\Pr[(\mathsf{T}, SK) \leftarrow Real: \mathcal{A}_2(\mathsf{T}, SK) = 1] - \Pr[(\mathsf{T}, SK) \leftarrow Fake_n: \mathcal{A}_2(\mathsf{T}, SK) = 1]| \le n\varepsilon(t).$$

In the distribution $Fake_n$, the values $d_{1,2}, d_{2,3}, \ldots, d_{n,1}$ are constrained by T according to the following n equations: $\log_g D_1 = d_{1,2} - d_{n,1}$, $\log_g D_2 = d_{2,3} - d_{1,2}$, …, $\log_g D_n = d_{n,1} - d_{n-1,n}$, where $g = e(P, P)$. Only $n - 1$ of there equations are linear independent. Due to $SK = e(P, P)^{d_{1,2}+d_{2,3}+\ldots+d_{n,1}}$, we have $\log_g SK = d_{1,2} + d_{2,3} + \ldots + d_{n,1}$. Since this final equation is linear independent from the set of equations above, the value of SK is independent of transcript T. This implies that even for a computationally-unbounded adversary $\mathcal{A}_2$: $\Pr[(\mathsf{T}, SK_0) \leftarrow Fake_n; SK_1 \leftarrow G_2; b \leftarrow \{0, 1\} | \mathcal{A}_2(\mathsf{T}, SK_b) = 1] = 1/2$.

Since $\varepsilon(t) = Adv_{G_2}^{DDH}(t)$, we have the result that the advantage of $\mathcal{A}_2$ on the event $\neg Forge$ is bounded by $2n \cdot Adv_{G_2}^{DDH}(t)$. Hence, we have $Adv_{\Psi}^{AGKE\text{-}fs}(t, 1, q_s) \le 2n \cdot Adv_{G_2}^{DDH}(t) + Adv_{\Phi}^{forge}(t)$. For the case of $q_{ex} > 1$, a standard hybrid argument immediately shows that $Adv_{\Psi}^{AGKE\text{-}fs}(t, q_{ex}, q_s) \le 2nq_{ex} \cdot Adv_{G_2}^{DDH}(t) + Adv_{\Phi}^{forge}(t)$. $\square$

According to Cui *et al.*'s single signature [15] and Lemma 1, we obtain that the advantage of forging the adopted signature scheme $Adv_{\Phi}^{forge}(t)$ is negligible. Under the decision Diffie-Hellman (DDH) assumption in the group G_2, the advantage $Adv_{G_2}^{DDH}(t)$ also is negligible. Thus, the proposed protocol Ψ is a secure authenticated group key exchange protocol providing forward secrecy.

5.4 Insider Attacks

Here, we prove that the proposed protocol is secure against insider attacks in the following Theorem 4. In other words, our protocol can resist malicious participants.

Theorem 4 In the random oracle model, the proposed ID-based authenticated group key exchange (IDAGKE) protocol is secure against insider attacks under the security level l_b, the k-CAA assumption, and the decision Diffie-Hellman (DDH) assumption in the group.

Proof: As mentioned in section 3, we say that an IDAGKE protocol is secure against insider attacks, if the protocol satisfies following three conditions: (1) it is a secure AGKE; (2) it is secure against insider impersonation attack; (3) it provides key agreement. By Theorems 2 and 3, we have proven that the proposed protocol can withstand insider impersonation attack and is a secure authenticated group key exchange protocol providing forward secrecy. Thus, the conditions (1) and (2) have satisfied.

In the key confirmation phase, each U_i can confirm that her/his group key SK_i is equal to all SK_j for $j = 1, 2, \ldots, n$ and $j \neq i$ by the batch verifying process. By Lemma 1, we have shown that the adopted batch verification technique is secure under the security level l_b. Thus, each participant can confirm that all computed group session keys from other participants are identical. This means that key agreement is achieved. Therefore, our proposed protocol is secure against insider attacks. ❑

6. PERFORMANCE ANALYSIS AND COMPARISONS

In this section, we want to analyze the computational cost for each participant in our proposed ID-based authenticated group key exchange (IDAGKE) protocol. Furthermore, we present the comparisons between our protocol and the previously proposed IDAGKE protocols in terms of key construction, the number of rounds, performance, and security property.

For convenience to evaluate the computational cost of our protocol, we only consider some time-consuming operations and define the following notations:

- TG_e: The time of executing a bilinear map operation e: $G_1 \times G_1 \rightarrow G_2$.
- TG_{mul}: The time of executing a scalar multiplication operation of point in G_1.
- TG_H: The time of executing a map-to-point hash function H_G: $\{0, 1\}^* \rightarrow G_1$.
- T_{exp}: The time of executing a modular exponentiation operation in Z_q.

In Round 1, $TG_{mul} + T_{exp}$ is required for computing (r_i, σ_i). In Round 2, each participant U_i requires $2TG_e + 3TG_{mul} + 4T_{exp}$ to verify (r_j, σ_j) for $j \in \{i - 1, i + 1\}$ and to compute (D_i, t_i, ρ_i). In Round 3, it requires $2TG_e + (2n - 1)TG_{mul} + (n + 2)T_{exp}$ for verifying all (t_j, ρ_j) and computing (SK_i, u_i, τ_i). Note that computing the key $SK_i = (r_{i-1})^{na_i} \cdot D_i^{n-1} \cdot D_{i+1}^{n-2} \cdot \cdots \cdot D_{i-2}$ requires T_{exp} due to $A_{i-1} = (r_{i-1})^{a_i}$, $A_i = A_{i-1} \cdot D_i$, $A_{i+1} = A_i \cdot D_{i+1}$, …, *etc*. so that $SK_i = A_{i-1} \cdot A_i \cdot A_{i+1} \cdot \cdots \cdot A_{i-2}$. In the key confirmation phase, $2TG_e + (2n - 2)TG_{mul} + nT_{exp}$ is required to verify all (u_j, τ_j) and to confirm all SK_j. As a result, it requires $6TG_e + (4n + 1)TG_{mul} + (2n + 7)T_{exp}$ for each participant U_i.

As described in section 1, we know that Choi *et al.*'s protocol [12] suffered from insider (participants) colluding attacks in [23, 24]. Although Shim also proposed a modification to overcome the insider colluding attacks in [24]. Actually, Shim's modification is still insecure against other insider colluding attacks in [19]. Recently, Choi *et al.* [19] also presented an improvement to resist the mentioned insider colluding attacks. However, we proved that their improvement still suffered from an insider colluding attack in [21]. In Table 1, we make the comparisons between our proposed protocol, Shim's modification [24], and Choi *et al*.'s improvement [19] in terms of key construction, number of rounds, performance, and security property. In section 5, we have demonstrated that our proposed

protocol is a provably secure ID-based authenticated group key exchange protocol with resisting malicious participants.

Table 1. Comparisons between our protocol and the previously proposed IDAGKE protocols.

	Shim's modification [24]	Choi *et al.*'s improvement [19]	Our protocol
Key construction	I	I	II
Rounds	2	2	3
Computational cost for each participant	$6TG_e + (2n + 6)TG_{mul} + (2n - 2)TG_H$	$6TG_e + (n + 11)TG_{mul} + (n + 3)TG_H$	$6TG_e + (4n + 1)TG_{mul} + (2n + 7)T_{exp}$
Security	Existing attacks [19]	Existing attacks [21]	Provably secure
Resistant to malicious participants	No	No	Yes

7. CONCLUSIONS AND FUTURE WORK

As compared to the certificate-based public-key systems, ID-based cryptographic protocols may simplify the certificate management. In this paper, we have proposed a secure ID-based authenticated group key exchange (IDAGKE) protocol based on key construction II. To reduce the cost of signature verifications, we adopt the batch verification technique into our protocol. In the random oracle model, we have proven that the proposed protocol satisfies the Katz-Shin's defined security model under the security level l_b, the k-CAA and decision Diffie-Hellman (DDH) in the group G_2 assumptions. Thus, our proposed protocol is a provably secure IDAGKE protocol with resistant to insider attacks. This means that our protocol can withstand malicious participants disturbing.

Recently, the certificateless cryptography [32, 33] was proposed to solve the key escrow issue in the ID-based public-key system. The key escrow means that the Key Generation Center (KGC) knows all users' private keys. In the certificateless public-key system, the KGC issues a partial key to user with respect to her/his identity and the user can generate an additional public/private key pair. When both the user's partial key and private key are known, the user can execute some cryptographic operations. Thus, the KGC cannot do any cryptographic operations on behalf of users. In the future, we will try to propose a secure certificateless authenticated group key exchange protocol.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments and constructive suggestions.

REFERENCES

1. G. Ateniese, M. Steiner, and G. Tsudik, "New multiparty authentication services and key agreement protocols," *IEEE Journal on Selected Areas in Communications*, Vol.

18, 2000, pp. 628-639.
2. E. Bresson, O. Chevassut, D. Pointcheval, and J. J. Quisquater, "Provably authenticated group Diffie-Hellman key exchange," in *Proceedings of the 8th ACM Conference on Computers and Communications Security*, 2001, pp. 255-264.
3. W. G. Tzeng, "A secure fault-tolerant conference-key agreement protocol," *IEEE Transactions on Computers*, Vol. 51, 2002, pp. 373-379.
4. Y. M. Tseng, "A robust multi-party key agreement protocol resistant to malicious participants," *The Computer Journal*, Vol. 48, 2005, pp. 480-487.
5. Y. M. Tseng, "An improved conference-key agreement protocol with forward secrecy," *Informatica*, Vol. 16, 2005, pp. 275-284.
6. Y. M. Tseng, "A communication-efficient and fault-tolerant conference-key agreement protocol with forward secrecy," *Journal of Systems and Software*, Vol. 80, 2007, pp. 1091-1101.
7. Y. M. Tseng, "A resource-constrained group key agreement protocol for imbalanced wireless networks," *Computers and Security*, Vol. 26, 2007, pp. 331-337.
8. Y. M. Tseng, "A secure authenticated group key agreement protocol for resource-limited mobile devices," *The Computer Journal*, Vol. 50, 2007, pp. 41-52.
9. A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of the 4th Annual International Cryptology Conference*, LNCS 196, 1984, pp. 47-53.
10. D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM Journal of Computing*, Vol. 32, 2003, pp. 586-615.
11. J. C. Cha and J. H. Cheon, "An identity-based signature from gap Diffie-Hellman groups," in *Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography*, 2003, pp. 18-30.
12. K. Y. Choi, J. Y. Hwang, and D. H. Lee, "Efficient ID-based group key agreement with bilinear maps," in *Proceedings of the 7th International Workshop on Theory and Practice in Public Key Cryptography*, 2004, pp. 130-144.
13. H. J. Yoon, J. H. Cheon, and Y. Kim, "Batch verifications with ID-based signatures," in *Proceedings of the 7th International Conference on Information Security and Cryptology*, 2005, pp. 233-248.
14. P. S. L. M. Barreto, B. Libert, N. McCullagh, and J. J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps," in *Proceedings of the 11th International Conference on the Theory and Application of Cryptology and Information Security*, LNCS 3788, 2005, pp. 515-532.
15. S. Cui, P. Duan, and C. W. Chan, "An efficient identity-based signature scheme with batch verifications," in *Proceedings of the 1st International Conference on Scalable Information Systems*, Article No. 22, 2006.
16. L. Chen, Z. Cheng, and N. P. Smart, "Identity-based key agreement protocols from pairings," *International Journal of Information Security*, Vol. 6, 2007, pp. 213-241.
17. Y. M. Tseng, T. Y. Wu, and J. D. Wu, "A pairing-based user authentication scheme for wireless clients with smart cards," *Informatica*, Vol. 19, 2008, pp. 285-302.
18. Y. M. Tseng, T. Y. Wu, and J. D. Wu, "Forgery attacks on an ID-based partially blind signature scheme," *International Journal of Computer Science*, Vol. 35, 2008, pp. 301-304.
19. K. Y. Choi, J. Y. Hwang, and D. H. Lee, "ID-based authenticated group key agreement secure against insider attacks," *IEICE Transactions on Fundamentals*, Vol. E91-

A, 2008, pp. 1828-1830.
20. Y. M. Tseng, T. Y. Wu, and J. D. Wu, "Towards efficient ID-based signature schemes with batch verifications from bilinear pairings," in *Proceedings of International Conference on Availability, Reliability and Security*, 2009, pp. 935-940.
21. T. Y. Wu and Y. M. Tseng, "Comment on an ID-based authenticated group key agreement protocol with withstanding insider attacks," *IEICE Transactions on Fundamentals*, Vol. E92-A, 2009, pp. 2638-2640.
22. T. Y. Wu and Y. M. Tseng, "An ID-based mutual authentication and key exchange protocol for low-power mobile devices," *The Computer Journal*, Vol. 53, 2010, pp. 1062-1070.
23. F. Zhang and X. Chen, "Attack on an ID-based authenticated group key agreement scheme from PKC 2004," *Information Processing Letters*, Vol. 91, 2004, pp. 191-193.
24. K. A. Shim, "Further analysis of ID-based authenticated group key agreement protocol from bilinear maps," *IEICE Transactions on Fundamentals*, Vol. E90-A, 2007, pp. 295-298.
25. J. Katz and J. S. Shin, "Modeling insider attacks on group key exchange protocols," in *Proceedings of the 12th ACM Conference on Computers and Communications Security*, 2005, pp. 180-189.
26. R. Sakai and M. Kasahara, "ID-based cryptosystems with pairing on elliptic curve," *Cryptology ePrint Archive*, Report 2003/054.
27. A. L. Ferrara, M. Green, and S. Hohenberger, "Practical short signature batch verification," in *Proceedings of the Cryptographers' Track at the RSA Conference on Topics in Cryptology*, 2009, pp. 309-324.
28. M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, 1993, pp. 62-73.
29. R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems based on pairing," in *Proceedings of Symposium on Cryptography and Information Security*, Article No. C20, 2000.
30. E. Bresson, M. Manulis, and J. Schwenk, "On security models and compilers for group key exchange protocols," in *Proceedings of the 2nd International Workshop on Security*, 2007, pp. 292-307.
31. J. Katz and M. Yung, "Scalable protocols for authenticated group key exchange," *Journal of Cryptology*, Vol. 20, 2007, pp. 85-113.
32. S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proceedings of the 9th International Conference on the Theory and Application of Cryptology and Information Security*, LNCS 2894, 2003, pp. 452-473.
33. B. C. Hu, D. S. Wong, Z. Zhang, and X. Deng, "Certificateless signature: A new security model and an improved generic construction," *Designs, Codes and Cryptography*, Vol. 42, 2007, pp. 109-126.

Tsu-Yang Wu (吳祖揚) received the B.S. and the M.S. degrees in Department of Applied Mathematics, Tatung University, Taiwan, in 2003 and 2005, respectively. He received the Ph.D. degree in the Department of Mathematics, National Changhua University of Education, Taiwan, in 2010. His research interests include applied cryptography, pairing-based cryptography, and computer network.

Yuh-Min Tseng (曾育民) received the B.S. degree in Computer Science and Engineering from National Chiao Tung University, Taiwan, in 1988; and the M.S. degree in Computer and Information Engineering from National Taiwan University in 1990 and the Ph.D. degree in Applied Mathematics from National Chung-Hsing University in 1999. He is currently a Professor in the Department of Mathematics, National Changhua University of Education, Taiwan. He is members of IEEE Computer Society, IEEE Communications Society and the Chinese Cryptology and Information Security Association (CCISA). His research interests include cryptography, network security, computer network and mobile communications. In 2006, his paper obtained the Wilkes Award from *The British Computer Society*. He is also editors of several international Journals.

Ching-Wen Yu (游靜玟) received the B.S. and the M.S. degrees in Department of Applied Mathematics, Tatung University, Taiwan, in 2003 and 2005, respectively. Her research interests include applied cryptography, network security and pairing-based cryptography.