INFORMATICA, 2012, Vol. 23, No. 2, 315–334

Towards ID-Based Authenticated Group Key Exchange Protocol with Identifying Malicious Participants

Tsu-Yang WU[1], Yuh-Min TSENG[2]

[1] *School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, P.R. China*

[2] *Department of Mathematics, National Changhua University of Education Jin-De Campus, Chang-Hua City 500, Taiwan, R.O.C.*

e-mail: ymtseng@cc.ncue.edu.tw

Received: November 2010; accepted: August 2011

Abstract. An authenticated group key exchange (AGKE) protocol allows participants to construct a common key and provide secure group communications in cooperative and distributed applications. Recently, Choi *et al.* proposed an identity (ID)-based authenticated group key exchange (IDAGKE) protocol from bilinear pairings. However, their protocol suffered from an insider colluding attack because it didn't realize the security issue of withstanding insider attacks. Withstanding insider attacks mean that it can detect whether malicious participants exist in the group key exchange protocol. Nevertheless, an AGKE protocol resistant to insider attacks is still unable to find "who are malicious participants". In this paper, we propose an ID-based AGKE protocol with identifying malicious participants. In our protocol, we use a confirmed computation property to achieve identifying malicious participants. Certainly, it is also secure against insider attacks. In the random oracle model and under related mathematical hard problems, we prove that the proposed protocol a secure AGKE protocol with identifying malicious participants.

Keywords: authenticated group key exchange, identity-based, bilinear pairing, malicious participant, insider attack.

1. Introduction

1.1. *Motivation*

Shamir (1984) proposed the concept of identity (ID)-based public-key system. As compared to the certificate-based public-key systems, Shamir's ID-based public-key system may simplify certificate management. However, it has a disadvantage that the user's private key must be generated by the single Key Generator Center (KGC). Boneh and Franklin (2001) presented a practical ID-based encryption system from bilinear pairings defined on elliptic curves. The security of their system is based on the bilinear Diffie–Hellman problem. In their system, the user's private key can be generated by several subcenters using a threshold technique. Later on, ID-based cryptographic protocols based on

bilinear pairings have received much attention from cryptographic researchers. Recently, many ID-based cryptographic schemes based on bilinear pairings have been proposed. Those schemes include authentication (Tseng *et al.*, 2008), encryption (Ren *et al.*, 2010), key agreement (Chen *et al.*, 2007; Wu and Tseng, 2010), signature (Cha and Cheon, 2003; Yoon *et al.*, 2004; Tseng *et al.*, 2009; Liu and Huang, 2010) and group key exchange (Choi *et al.*, 2004; Shim, 2007; Choi *et al.*, 2008).

Owing to the popularity of group-oriented applications such as electronic conferences and collaboration works, secure group communication is an important research issue for network security. The secure group key exchange protocol design is critical for providing secure group communications over an insecure channel. Consider a situation that an emergency and some secure conferences must be held prior to a special time, such as military applications, rescue missions and emergency negotiations (Tzeng, 2002; Tseng, 2005). Delay or dissolution of the conference will have serious negative consequences. If a malicious participant attempts to disrupt the establishment of the group key, other honest participants will be unable to compute the same group key. If these honest participants cannot identify the malicious participant, the destruction of the conference could cause serious damage. Note that a malicious participant is considered as a legitimate participant who is fully controlled by adversary. However, these existing ID-based group key exchange protocols (Choi *et al.*, 2004; Shim, 2007; Choi *et al.*, 2008) suffer from insider colluding attacks (Wu and Tseng, 2009) and do not provide the functionality of identifying malicious participants.

1.2. *Related Work*

An authenticated group key exchange (AGKE) protocol is an important security mechanism. It allows participants to construct a common key and provide secure group communications in cooperative and distributed applications. It can be used to encrypt or 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 (Bresson *et al.*, 2001; Tzeng, 2002; Tseng, 2005; Burmester and Desmedt, 2005; Katz and Shin, 2005; Tseng, 2007; Tseng and Wu, 2010).

Tzeng (2002) proposed a provable secure group key exchange protocol with identifying malicious participants. This protocol employs two rounds to compute a group key following the detection of all malicious participants. This protocol is provable secure against passive attacks and attacks by impersonators. However, the message size sent by each participant is proportional to the number of participants.

Katz and Shin (2005) proposed a security model and a universal composability (UC) complier for AGKE protocols. This enhanced security model provides the formal security definition of AGKE protocols in the existence of malicious participants. The concept of the UC complier is to use the explicit key confirmation property to detect whether malicious participants exist in the group key establishment. The UC complier requires one additional round and n signature verifications, where n is the number of participants. However, it is still unable to find "who are malicious participants".

Tseng (2005) proposed an efficient non-authenticated group key exchange protocol with identifying malicious participants. The proposed protocol requires only a constant message size for each participant. That is, the message size sent by each participant is independent of the number of participants. By its very nature, a non-authenticated group key exchange protocol cannot provide participant and message authentication, so it must rely on an authenticated network channel. Furthermore, Tseng (2007) proposed an AGKE) protocol with identifying malicious participants while remaining efficient for message size sent by each participant.

Choi *et al.* (2004) presented two group key exchange (GKE) protocols using bilinear pairings. One is non-authenticated GKE, and the other is ID-based authenticated group key agreement (AGKE). Unfortunately, Zhang and Chen (2004) presented an impersonation attack on Choi *et al.*'s AGKE protocol. Shim (2007) also showed that Choi *et al.*'s AGKE protocol is not secure against an insider colluding attack. In which, three malicious participants can collude to impersonate an honest participant to the other participants in the group.

Shim (2007) suggested a modification to overcome insider colluding attacks. Recently, Choi *et al.* (2008) proved that Shim's suggestion still suffered from an insider colluding attack. Meanwhile, Choi *et al.* (2004) also presented an improvement of their original AGKE protocol to resist the mentioned insider colluding attacks. In their improved AGKE protocol, they applied an ID-based signature scheme on the broadcasting messages. They claimed that the transcript of the session is guaranteed to be fresh and insider attacks are impossible. In particular, they used batch verifications to reduce the computational cost in their modified AGKA protocol.

Actually, Wu and Tseng (2009) have shown that Choi *et al.*'s (2008) improved AGKE protocol is still insecure against other insider colluding attacks. Two malicious participants can collude to impersonate an honest participant to other participants in the group. Meanwhile, Wu and Tseng also proved that the adopted batch verification in their modified AGKE protocol suffers from a forgery attack. The forgery attack means that some malicious participants can collude to impersonate a non-involved user to generate valid multiple signatures to pass the batch verification (Kim *et al.*, 2011). Certainly, by applying the UC complier presented by Katz and Shin (2005) to Choi *et al.*'s improved AGKE protocol, it may enjoy the explicit key confirmation property to detect whether malicious participants exist in the group key establishment. However, the UC complier requires one additional round and n signature verifications. Even if the resulting ID-based AGKE protocol is secure against insider attacks, it cannot provide the functionality of identifying malicious participants.

1.3. *Contributions*

In this paper, we present an ID-based authenticated group key exchange protocol with identifying malicious participants. In the proposed protocol, each participant can confirm whether the broadcast values are correctly computed by other participants, called the confirmed computation property. We use this property to achieve implicit key confirmation

so that the proposed protocol can identify malicious participants. In particular, it does not require additional round. In the random oracle model (Bellare and Rogaway, 1993) and under the computational Diffie–Hellman as well as the decision bilinear Diffie–Hellman assumptions (Boneh and Franklin, 2001; Chen *et al.*, 2007), we show that the proposed ID-based AGKE protocol satisfies Katz and Shin's (2005) security model. In other words, it is a secure AGKE protocol providing forward secrecy and can resist insider attacks. The other point is that in the proposed protocol malicious participants can be identified. Finally, we compare our protocol with the previously proposed non-ID-based and ID-based AGKE protocols to demonstrate the advantage of the proposed protocol.

The remainder of this paper is organized as follows. In Section 2, we briefly review the concept of bilinear pairings as well as related security assumptions. The security model and notions of AGKE are given in Section 3. In Section 4, we present our ID-based AGKE protocol with identifying malicious participants. Security analysis of the proposed protocol is given in Section 5. In Section 6, we make the performance analysis and comparisons. Conclusions are drawn in Section 7.

2. Preliminaries

In this section, we depict compendiously the concepts of bilinear pairings, the related mathematical assumptions. Readers can refer to Boneh and Franklin (2001), Chen *et al.* (2007) for full descriptions.

2.1. *Bilinear Pairings*

Here, G_1 and G_2 are two groups of the same large prime order q, where G_1 is an additive cyclic group, and G_2 is a multiplicative cyclic group. In particular, 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 conditions, 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 exist $P, Q \in G_1$ such that $e(P, Q) \neq 1$.
3. Computable: There exists an efficient algorithm to compute $e(P, Q)$ for all P, $Q \in G_1$.

2.2. *Related Security Hard Problems and Assumptions*

For convenience to prove the security of the proposed protocol, we summarize some hard problems and assumptions for bilinear pairings on elliptic curves as follows.

- Computational Diffie–Hellman (CDH) problem: Given $P, aP, bP \in G_1$ for some $a, b \in Z_q^*$, the CDH problem is to compute $abP \in G_1$.
- CDH assumption: No probabilistic polynomial time (PPT) algorithm can solve the CDH problem with a non-negligible advantage.

- Decision bilinear Diffie–Hellman (DBDH) problem: Given $P, aP, bP, cP, dP \in G_1$, for some $a, b, c, d \in Z_q^*$, the DBDH problem is to distinguish $(P, aP, bP, cP, dP, e(P,P)^{abc})$ from $(P, aP, bP, cP, dP, e(P,P)^d)$.
- DBDH assumption: No PPT algorithm can solve the DBDH problem with a non-negligible advantage.

2.3. *Notations*

The following notations are used throughout this paper:

- e: an admissible bilinear map, e: $G_1 \times G_1 \to G_2$.
- P: a generator of the group G_1.
- s: the system private key, where $s \in Z_q^*$.
- P_{pub}: the system public key, where $P_{pub} = s \cdot P$.
- ID_i: the identity of participant U_i.
- DID_i: the private key of participant U_i.
- H_G: a map-to-point hash function, H_G: $\{0,1\}^* \to G_1$.
- H_1: a one-way hash function, H_1: $\{0,1\}^* \times G_1 \to Z_q$.
- H_2: a one-way hash function, H_2: $\{0,1\}^* \times G_1^3 \to Z_q$.

3. Security Model and Notions

In this section, we briefly review the security model and notions for an authenticated group key exchange (AGKE) protocol. The following notations and definitions are referred to Bresson *et al.* (2001), Choi *et al.* (2004), Katz and Shin (2005), Bresson and Manulis (2008).

Participants and initialization. Assume that each participant U_i has a unique identity $ID_i \in \{0,1\}^l$, where l is the bit length. For simplicity, there is a fixed set $\mathcal{U} = \{U_1, U_2, \ldots, U_n\}$ which is a polynomial-size set of potential participants. Here, we allow that each participant $U_i \in \mathcal{U}$ performs the protocol many times with different participants. In other words, each U_i has the multiple instances to execute the protocol. We denote the instance t of participant U_i as a Π_i^t oracle, where t is a positive integer. The public parameters and the set of identities $ID = \{ID_1, ID_2, \ldots, ID_n\}$ are known by all participants (including adversary).

For an ID-based AGKE protocol, it requires two additional algorithms:

- **Setup algorithm.** The system private key and the public parameters are generated by this algorithm.
- **Extract algorithm.** Each participant $U_i \in \mathcal{U}$ can obtain the private key DID_i using this algorithm.

Session ID, partner ID, and related notions. The session ID of oracle Π_i^t is a set defined as $SID(\Pi_i^t)$. It equals the concatenation of all messages sent and received by Π_i^t during its execution. The partner ID of oracle Π_i^t is a set defined as $PID(\Pi_i^t)$. It contains the

identities of participants who want to establish a group session key with Π_i^t in the group. The session ID and the partner ID are public information. We say that an oracle Π_i^t is accepted, if it can compute a valid group 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 has decided to terminate without accepting is a public information. We say that two oracles Π_i^t and Π_j^s are partner if and only if it satisfies (1) two oracles have accepted; (2) $SID(\Pi_i^t) = SID(\Pi_j^s)$; (3) $PID(\Pi_i^t) = PID(\Pi_j^s)$.

Adversarial model. Formally, an adversary $\mathcal{A}$ is a probabilistic polynomial time algorithm. We allow $\mathcal{A}$ to potentially control all communications completely in an ID-based AGKE protocol via accessing to a set of oracles. Here, $\mathcal{A}$ can make different types of queries in the following:

- ***Extract***(ID_α). In this query, $\mathcal{A}$ can get a private key DID_α corresponding to identity ID_α, where $ID_\alpha \notin \mathcal{ID}$.
- ***Execute***$(\mathcal{ID})$. In this query, $\mathcal{A}$ can get a complete transcript of an honest executing between the participants belong to $\mathcal{ID}$. In particular, the number of group participants is chosen by $\mathcal{A}$.
- ***Send***(Π_i^t, M). When $\mathcal{A}$ makes this query with a message M to an oracle Π_i^t, this oracle performs the computations and responses the answers according to the protocol.
- ***Reveal***(Π_i^t). In this query, $\mathcal{A}$ can get a group session key from an oracle Π_i^t.
- ***Corrupt***(ID_i). In this query, $\mathcal{A}$ can get a private key DID_i of identity ID_i.
- ***Test***(Π_i^t). $\mathcal{A}$ can send only one *Test* query to an oracle Π_i^t. Upon receiving this query, this oracle flips an unbiased coin b. If $b = 1$, then Π_i^t returns the group session key. Otherwise, it returns a random string.

In the above model, there exist two types of adversaries. A **passive adversary** is allowed to make *Execute*, *Reveal*, *Corrupt*, and *Test* queries. An **active adversary** is allowed to make the above all types of queries. Though the *Execute* query can be substituted for making the *Send* query repeatedly, we also use this query to get more precise analysis here.

Secure AGKE. A secure AGKE protocol contains following four parts.

1. **Freshness.** An oracle Π_i^t is *fresh*, if one of the following is true:
   - Π_i^t has accepted a group session key.
   - Neither Π_i^t nor its partners have been asked a *Reveal* query.
   - No *Corrupt*(ID_V) query was asked before a *Send*(Π_j^s, M) query, where $ID_V \in PID(\Pi_i^t)$ and Π_j^s is a partner of Π_i^t.

   We assume that all oracles are considered ***fresh***.
2. **The security of AGKE protocol.** The security of ID-based AGKE protocol is defined in the following game played between an active adversary $\mathcal{A}$ and a set of oracles Π_i^t, where Π_i^t is an oracle of participant $U_i \in \mathcal{V}$.
   - **Initialization.** The system private key, public parameters, and participants' private keys are generated in this phase.

- **Queries.** $\mathcal{A}$ may make different types of queries and gets back the answers corresponding to oracles.
- **Guess.** Finally, $\mathcal{A}$ 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 a 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 by this oracle answering this query. The advantage of $\mathcal{A}$ in attacking a protocol $\mathtt{P}$ is defined as $Adv_{\mathcal{A},\mathtt{P}}(k) = |2 \cdot \Pr[Succ] - 1|$. We say that a protocol $\mathtt{P}$ is a secure ID-based AGKE protocol, if the advantage $Adv_{A,P}(k)$ is negligible.

3. **Forward secrecy.** We say that an ID-based AGKE protocol Ψ provides *forward secrecy*, if an adversary $\mathcal{A}$ cannot get some useful information about the previously established group session keys after $\mathcal{A}$ makes the *Corrupt* query in Ψ. The maximum advantage of $\mathcal{A}$ which attacks the protocol Ψ within running time t is defined as $Adv_{\Psi}^{AGKE-fs}(t, q_{ex}, q_s)$, where q_{ex} and q_s are maximum numbers of making the *Execute* and *Send* queries, respectively.
4. **Authentication.** We say that an ID-based AGKE protocol provides *implicit key authentication*, if the participants are assured that nobody other than its partners can learn the value of a particular session key. In particular, an adversary should not learn the key. Note that the property of implicit key authentication does not guarantee that partners have actually obtained the key.

Katz and Shin (2005) defined the concept of insider attacks for an AGKE protocol. We briefly review it as follows.

Insider attacks. An ID-based AGKE protocol is secure against *insider attacks* if it satisfies following conditions:

1. **Secure AGKE.**
2. **Withstanding insider impersonation attack.** We say that an adversary succeed in an *insider impersonation attack* if there exists a participant U_j and an oracle Π_i^t such that
   (1) The user U_j is not corrupted.
   (2) Π_i^t accepts and $ID_{U_j} \in PID(\Pi_i^t)$.
   (3) There does not exist an oracle Π_j^s of U_j such that $SID(\Pi_i^t) = SID(\Pi_j^s)$ and $PID(\Pi_i^t) = PID(\Pi_j^s)$.
   (4) Neither U_i nor U_j are corrupted at the time Π_i^t accepts.

   Note that the above conditions (1), (2), and (3) tell us that $\mathcal{A}$ impersonates U_j to Π_i^t. We say that an ID-based AGKE protocol is secure against ***insider impersonation attack***, if any adversary $\mathcal{A}$ with a negligible advantage succeeds in the above attack.
3. **Key agreement.** We say that an ID-based AGKE protocol does not provide *key agreement*, if there are partnered instances Π_i^t and Π_j^s such that (1) neither U_i nor U_j are corrupted; (2) $SK_i^t \neq SK_j^s$.

According to the definition of insider attacks, the condition "key agreement" considers the explicit key confirmation property, and the condition "secure against insider impersonation attack" concerns with the mutual authentication property.

Malicious participant. We say that a participant U_m is a malicious participant in an AGKE protocol, if U_m is a legitimate participant in the protocol, nevertheless, she/he is fully controlled by an adversary.

4. Proposed Protocol

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

Let $U = \{U_1, U_2, \ldots, U_n\}$ be a set of participants who want to establish a group session key SK. Note that the indices are subject to modulo n, that is U_{n+1} and U_0 denote U_1 and U_n, respectively. Assume that each participant U_i has a unique identity ID_i, PID is the concatenation of the identities of participants taking part in a session, i.e., $PID = ID_1 \| ID_2 \| \cdots \| ID_n$ 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. In addition, U_i's public key and private key pair is $(ID_i, DID_i = H_G(ID_i) \cdot s)$. The details of the proposed protocol are described as follows.

[**Round 1**] Each participant U_i randomly picks an integer $a_i \in Z_q^*$ and computes $P_i = a_i \cdot P,\ h_i = H_1(M \| PID \| ID_i, P_i)$, and $V_i = a_i \cdot H_G(ID_i) + h_i \cdot DID_i$. Finally, each U_i broadcasts (ID_i, P_i, V_i) to other participants.

[**Round 2**] Upon receiving $(ID_{i-1}, P_{i-1}, V_{i-1})$ and $(ID_{i+1}, P_{i+1}, V_{i+1})$, each participant U_i checks the following equation:

$$e\Big(P, \sum_{k\in\{-1,1\}} V_{i+k}\Big) \stackrel{?}{=} \prod_{k\in\{-1,1\}} e\big(P_{i+k} + h_{i+k} \cdot P_{pub}, H_G(ID_{i+k})\big),$$

where $h_{i+k} = H_1(M \| PID \| ID_{i+k}, P_{i+k})$ and $k \in \{-1, 1\}$.
If the above checking equation holds, each U_i uses the secret a_i to compute $D_i = e(P_{i+1} - P_{i-1}, P_{pub})^{a_i}$. Then, U_i generates a signature tuple on the message $PID \| ID_i \| D_i \| S$, where $S = P_1 \| P_2 \| \ldots \| P_n$ as follows: U_i chooses a random integer $r_i \in Z_q^*$ and computes $\alpha_i = r_i \cdot P,\ \beta_i = r_i \cdot (P_{i+1} - P_{i-1}),\ k_i = H_2(PID \| ID_i \| D_i \| S, P_{i+1} - P_{i-1}, \alpha_i, \beta_i), \gamma_i = r_i \cdot P_i + k_i a_i \cdot P_{pub}$. Finally, each U_i sends $\sigma_i = (ID_i, D_i, \alpha_i, \beta_i, \gamma_i)$ to all other participants.

[**Group session key computation**] Upon receiving all $\sigma_j = (ID_j, D_j, \alpha_j, \beta_j, \gamma_j)$ for $j = 1, 2, \ldots, n$ and $j \neq i$, each U_i checks

$$e(P, \gamma_j) \stackrel{?}{=} e(P_j, \alpha_j + k_j \cdot P_{pub}) \quad \text{and}$$

$$e(P_{j+1} - P_{j-1}, \gamma_j) \stackrel{?}{=} e(\beta_j, P_j) \cdot D_j^{k_j},$$

where $k_j = H_2(PID \| ID_j \| D_j \| S, P_{j+1} - P_{j-1}, \alpha_j, \beta_j)$ and $S = P_1 \| P_2 \| \ldots \| P_n$. For reducing the computational cost of verification, by the same batch verification method presented by Tseng *et al.* (2009), $e(P, \sum_{j=1}^{n} \gamma_j) \stackrel{?}{=} \prod_{j=1, j\neq i}^{n} e(P_j, \alpha_j + k_j \cdot P_{pub})$ can be used to replace $e(P, \gamma_j) \stackrel{?}{=} e(P_j, \alpha_j + k_j \cdot P_{pub})$ for $j = 1, 2, \ldots, n$ and $j \neq i$. If the above equations hold, each U_i not only believes that the σ_j is produced by U_j with the secret a_j, but also can confirm that

$$\begin{aligned} D_j &= e(P_{j+1} - P_{j-1}, P_{pub})^{a_j} = e\big((a_{j+1} - a_{j-1}) \cdot P, s \cdot P\big)^{a_j} \\ &= e(P, P)^{s a_j (a_{j+1} - a_{j-1})} \end{aligned}$$

for $j = 1, 2, \ldots, n$ and $j \neq i$. Therefore, each participant U_i can compute the same group session key $SK = e(a_i \cdot P_{i-1}, P_{pub})^n \cdot D_i^{n-1} \cdot D_{i+1}^{n-2} \cdots D_{i-2}$.

[**Malicious participant identifying**] If a participant U_m tries to send a wrong $\sigma_m = (ID_m, D_m, \alpha_m, \beta_m, \gamma_m)$ to disrupt the establishment of a group session key, then he will be determined as a malicious participant because two equations $e(P, \gamma_m) = e(P_m, \alpha_m + k_m \cdot P_{pub})$ and $e(P_{m+1} - P_{m-1}, \gamma_m) = e(\beta_m, P_m) \cdot D_m^{k_m}$ do not hold. Meanwhile, U_m will be deleted from the participant set U. Then other honest participants may rerun the protocol.

5. Security Analysis

In this section, we present security analysis of our presented protocol in the random oracle model (Bellare and Rogaway, 1993) and under the computational Diffie–Hellman as well as the decision bilinear Diffie–Hellman assumptions. First, we focus on ID and forgery attacks.

[**ID and forgery attacks**]

Here, we show that the proposed ID-based AGKE protocol is secure against ID and forgery attacks. Note that we apply the ID-based batch signature scheme in Yoon *et al.* (2004) to Round 1 of our proposed protocol. Hence, the signature security of Round 1 in our protocol is referred to Yoon *et al.* (2004). Therefore, we can claim that the signature in Round 1 is secure against ID and forgery attacks. In the following Lemma 1, we show that the signature in Round 2 is secure against ID and forgery attacks. Therefore, we obtain that the presented ID-based AGKE protocol is secure against ID and forgery attacks in Theorem 1.

Lemma 1. *In the random oracle model, assume that a probabilistic polynomial time adversary $\mathcal{A}$ with a non-negligible advantage can break the Round* 2 *of proposed ID-based AGKE protocol under the adaptive chosen message and ID attacks. Then, there exists a probabilistic polynomial time algorithm $\mathcal{C}_1$ with a non-negligible advantage that can solve the computational Diffie–Hellman problem.*

Proof. By assumption, $\mathcal{A}$ can execute adaptive chosen message and ID attacks to the Round 2 of presented ID-based AGKE protocol with a non-negligible advantage. By Cha and Cheon (2003), it implies that there exists a probabilistic polynomial time algorithm $\mathcal{A}_1$ with a non-negligible advantage for adaptive chosen message and fixed ID attacks.

For convenience to prove Lemma 1, we define the EUF-AGKE-ACMA game played between the challenger $\mathcal{C}_1$ and $\mathcal{A}_1$ as follows.

- **Initialization.** The algorithm $\mathcal{C}_1$ generates the public parameters $Param = \{G_1, G_2, e, q, P, P_{pub}, PID\}$ of proposed protocol and random values $P_i \in G_1$ for $i = 1, 2, \ldots, n$. Here, $\mathcal{C}_1$ acts as a challenger in this game. In addition, $\mathcal{C}_1$ needs to maintain three lists L_1, L_2, and L_3. Three lists are initially empty and are used to keep track of answers to *Extract*, $H_2()$, and *Issuing* queries, respectively. At the beginning of the game, $\mathcal{C}_1$ gives the public parameters and those P_i to $\mathcal{A}_1$ for $i = 1, 2, \ldots, n$.
- **Queries.** The challenger $\mathcal{C}_1$ is responsible to answer the queries which are made by $\mathcal{A}_1$ as follows:
  1. **Extract query.** Upon receiving this query with identity ID_i, the challenger $\mathcal{C}_1$ generates the corresponding private key DID_i by *Extract* algorithm and *Param*. Then, $\mathcal{C}_1$ returns DID_i to $\mathcal{A}_1$ and adds $(ID_i,\ DID_i)$ into the list L_1.
  2. $\mathbf{H_2}()$ **query.** Upon receiving the *hash* query request $H_2(\tau_i)$, where $\tau_i = (PID \| ID_i \| D_i \| S, P_{i+1} - P_{i-1}, \alpha_i, \beta_i)$ from the adversary $\mathcal{A}_1$, the challenger $\mathcal{C}_1$ searches τ_i in the list L_2. If τ_i is found, $\mathcal{C}_1$ returns the corresponding value. Otherwise, $\mathcal{C}_1$ returns a random value $k_{Ri} \in Z_q^*$ and adds (τ_i, k_{Ri}) into L_2.
  3. **Issuing query.** The adversary $\mathcal{A}_1$ chooses an identity ID_i and a random value $P_{bi} = P_{i+1} - P_{i-1} \in G_1$. Then, $\mathcal{A}_1$ sends them to $\mathcal{C}_1$. Upon receiving the *Issuing* query request for (ID_i, P_{bi}), $\mathcal{C}_1$ randomly selects two values $x_{ai}, r_i \in {}_R Z_q^*$ to compute $D_i = e(P_{bi}, P_{pub})^{x_{ai}}$, $\alpha_i = r_i \cdot P$, $\beta_i = r_i \cdot P_{bi}$, and $\gamma_i = r_i \cdot P_i + k_i x_{ai} \cdot P_{pub}$. Here, k_i is the simulated value of $H_2()$ query as the mentioned above. Finally, the challenger $\mathcal{C}_1$ returns $\sigma_i = (D_i, \alpha_i, \beta_i, \gamma_i)$ to $\mathcal{A}_1$ as the answer and adds $(ID_i, P_{bi}, \sigma_i = (D_i, \alpha_i, \beta_i, \gamma_i))$ into the list L_3.

Assume that the adversary $\mathcal{A}_1$ can forge a valid tuple $(ID_i', P_{bi}', \sigma_i' = (D_i', \alpha_i', \beta_i', \gamma_i'))$ with a non-negligible advantage, where P_{bi}' did not appear in any *Issuing* query. Following the Forking lemma in Pointcheval and Stern (2000), this lemma adopts the "oracle replay attack" using a polynomial replay of the attack with the same random tape and a different oracle. $\mathcal{A}_1$ can output two valid tuples $((ID_i', P_{bi}', \sigma_i' = (D_i', \alpha_i', \beta_i', \gamma_i'))$ and $((ID_i', P_{bi}', \sigma_i'' = (D_i'', \alpha_i', \beta_i', \gamma_i''))$ which pass the verification equation. Thus,

$$e(P, \gamma_i') = e(P_i, \alpha_i' + k_i' \cdot P_{pub}) \quad \text{and}$$
$$e(P, \gamma_i'') = e(P_i, \alpha_i' + k_i'' \cdot P_{pub}),$$

where k_i' and k_i'' are hash values from $H_2()$ query. By the bilinear pairing operations, we obtain $e(P, \gamma_i' - \gamma_i'') = e(P_i, (k_i' - k_i'') \cdot P_{pub})$.

Given $P_i = xP$ and $P_{pub} = yP$ for some $x, y \in Z_q^*$, it implies $e(P, \gamma_i' - \gamma_i'') = e(xP, (k_i' - k_i'') \cdot yP)$. By the property of bilinear pairing, we have $e(P, \gamma_i' - \gamma_i'') =$

$e(P,(k_i'-k_i'')\cdot xyP)$. Thus, $xyP=(\gamma_i'-\gamma_i'')/(k_i'-k_i'')$. Hence, the challenger $\mathcal{C}_1$ can easy obtain xyP from a CDH tuple $(P,P_i,P_{pub})=(P,xP,yP)$. It is a contradiction for the computation Diffie–Hellman assumption.

Theorem 1. *In the random oracle model and under the computational Diffie–Hellman assumption, the presented ID-based AGKE protocol is secure against ID and forgery attacks.*

Proof. In Round 1 of our proposed protocol, we use the ID-based batch signature scheme in Yoon *et al.* (2004) to provide participant authentication, thus it is secure against ID and forgery attacks. By Lemma 1, we have shown that the individual signature in Round 2 is also secure against ID and forgery attacks. By the same batch verification method presented by Tseng *et al.* (2009), we can reduce the computational costs of individual verifications while it remains security of withstanding ID and forgery attacks. Therefore, the presented ID-based AGKE protocol is secure against ID and forgery attacks.

[**Secure AGKE providing forward secrecy**]

In the following, we prove that the proposed ID-based AGKE protocol is a secure AGKE protocol providing forward secrecy under the decision bilinear Diffie–Hellman (DBDH) assumption and the security of the adopted ID-based signature schemes.

Theorem 2. *Assume that the hash functions H_G, H_1, and H_2 are random oracles. Then the proposed ID-based AGKE protocol Ψ is a secure AGKE providing forward secrecy under the decision bilinear Diffie–Hellman (DBDH) assumption. Precisely,*

$$Adv_{\Psi}^{AGKE-fs}(t,q_{ex},q_s)\leqslant 2nq_{ex}\cdot Adv_{G_1,G_2,e}^{DBDH}(t)+Adv_{\Psi}^{forge}(t),$$

where q_{ex} and q_s are maximum numbers of making the Execute and Send queries, respectively. Note that $Adv_{G_1,G_2,e}^{DBDH}(t)$ is the advantage of solving the DBDH problem, and $Adv_{\Psi}^{forge}(t)$ is the advantage of any forgers attacking the proposed ID-based AGKE protocol Ψ.

Proof. Assume that $\mathcal{A}$ is an active adversary in attacking the proposed ID-based AGKE protocol Ψ with a non-negligible advantage. We consider two possible attacking cases. Case 1 is that $\mathcal{A}$ can impersonate a participant (forging authentication transcripts) with the advantage. Case 2 is that $\mathcal{A}$ can break the protocol Ψ without modifying any transcripts with the advantage.

Case 1. We assume that $\mathcal{A}$ has an adaptive impersonation ability to break Ψ. Using $\mathcal{A}$, we can construct a forger $\mathcal{F}$ who generates two valid signature pairs (ID,aP,V) and $(ID,D,\alpha,\beta,\gamma)$ with respect to the proposed protocol Ψ as follows. The forger $\mathcal{F}$ generates all other public/private key pairs for the system and simulates the oracle queries of $\mathcal{A}$. This simulation is perfect indistinguishable from $\mathcal{A}$'s oracle queries

unless $\mathcal{A}$ makes the *Corrupt*(*ID*) query. If it occurs, then $\mathcal{F}$ terminates. Otherwise, if $\mathcal{A}$ outputs two valid signature pairs (ID, aP, V) and $(ID, D, \alpha, \beta, \gamma)$, $\mathcal{F}$ generates $\mathcal{A}$'s pairs, that is (ID, aP, V) and $(ID, D, \alpha, \beta, \gamma)$. Let *Forge* be the event that $\mathcal{A}$ can output valid signature pairs, and Pr[*Forge*] be the corresponding probability of this event. Then, the probability of $\mathcal{F}$ successfully generating two valid signature pairs satisfies $\Pr[Forge] \leqslant Adv_{F,\Psi}^{forge}(t) \leqslant Adv_{\Psi}^{forge}(t)$.

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

$$Param = \left\{ \begin{array}{l} (G_1, G_2, e) \leftarrow \mathrm{KGC};\ P \leftarrow G_1;\ s \leftarrow Z_q^*;\ P_{pub} = s \cdot P; \\ DID_1 = H_G(ID_1) \cdot s, \ldots, DID_n = H_G(ID_n) \cdot s: \\ (G_1, G_2, e, P, P_{\mathrm{pub}}, PID) \end{array} \right\} \quad \text{and}$$

$$Real = \left\{ \begin{array}{l} a_1, \ldots, a_n,\ h_1, \ldots, h_n,\ r_1, \ldots, r_n,\ k_1, \ldots, k_n \leftarrow Z_q^*; \\ P_1 = a_1 P, \ldots, P_n = a_n P; \\ V_1 = a_1 H_G(ID_1) + h_1 DID_1, \ldots,\ V_n = a_n H_G(ID_n) + h_n DID_n; \\ D_1 = e(P_2 - P_n, P_{pub})^{a_1}, \ldots, D_n = e(P_1 - P_{n-1}, P_{pub})^{a_n}; \\ \alpha_1 = r_1 P, \ldots, \alpha_n = r_n P; \\ \beta_1 = r_1(P_2 - P_n), \ldots, \beta_n = r_n(P_1 - P_{n-1}); \\ \gamma_1 = r_1 P_1 + k_1 a_1 P_{pub}, \ldots, \gamma_n = r_n P_n + k_n a_n P_{pub}; \\ \mathcal{T} = (P_1, \ldots, P_n, V_1, \ldots, V_n, D_1, \ldots, D_n, \alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \\ \quad \gamma_1, \ldots, \gamma_n); \\ SK = e(a_1 P_n, P_{pub})^n \cdot D_1^{n-1} \cdot D_2^{n-2} \cdots D_{n-1}: (\mathcal{T}, SK) \end{array} \right\},$$

where $\mathcal{T}$ denotes the transcript, and SK is the group session key.

Note that $D_i = e(P_{i+1} - P_{i-1}, P_{pub})^{a_i} = \frac{e(a_i P_{i+1}, P_{pub})}{e(a_i P_{i-1}, P_{pub})} = \frac{e(a_i a_{i+1} P, P_{pub})}{e(a_{i-1} a_i P, P_{pub})}$. Then, we can define the distribution $Fake_1$ in the following

$$Fake_1 = \left\{ \begin{array}{l} d_{1,2}, a_1, \ldots, a_n, h_1, \ldots, h_n, r_1, \ldots, r_n, k_1, \ldots, k_n \leftarrow Z_q^*; \\ P_1 = a_1 P, \ldots, P_n = a_n P; \\ V_1 = a_1 H_G(ID_1) + h_1 DID_1, \ldots, V_n = a_n H_G(ID_n) + h_n DID_n; \\ D_1 = \frac{e(d_{1,2} P, P_{pub})}{e(a_n a_1 P, P_{pub})},\ D_2 = \frac{e(a_2 a_3 P, P_{pub})}{e(d_{1,2} P, P_{pub})}, \ldots, \\ \quad D_n = \frac{e(a_n a_1 P, P_{pub})}{e(a_{n-1} a_n P, P_{pub})}; \\ \alpha_1 = r_1 P, \ldots, \alpha_n = r_n P;\ \beta_1 = r_1(P_2 - P_n), \ldots, \\ \quad \beta_n = r_n(P_1 - P_{n-1}); \\ \gamma_1 = r_1 P_1 + k_1 a_1 P_{pub}, \ldots, \gamma_n = r_n P_n + k_n a_n P_{pub}; \\ \mathcal{T} = (P_1, \ldots, P_n, V_1, \ldots, V_n, D_1, \ldots, D_n, \alpha_1, \ldots, \alpha_n, \\ \quad \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_n); \\ SK = e(a_1 P_n, P_{pub})^n \cdot D_1^{n-1} \cdot D_2^{n-2} \cdots D_{n-1}: (\mathcal{T}, SK) \end{array} \right\}.$$

Since $\mathcal{A}$ can obtain all private keys DID_i and hash values h_i by making the *Corrupt* and *Hash* queries, it can compute all $a_i H_G(ID_i) = V_i - h_i DID_i$ for $i = 1, 2, \ldots, n$.

According to the discrete logarithm problem in the group G_1 is intractable, these values offer no information about a_i for $i = 1, 2, \ldots, n$.

In the following, we will show that the problem to distinguish *Real* from $Fake_1$ can be reduced to solve the DBDH problem. Here, we let $\varepsilon(t) = Adv^{DBDH}_{G_1,G_2,e}(t)$.

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

$$\begin{aligned}&\big|\Pr\big[(\mathcal{T}, SK) \leftarrow \textit{Real}\colon A(\mathcal{T}, SK) = 1\big]\\&\quad - \Pr\big[(\mathcal{T}, SK) \leftarrow \textit{Fake}_1\colon A(\mathcal{T}, SK) = 1\big]\big| \leqslant \varepsilon(t).\end{aligned}$$

Proof. As note, $D_i = e(P_{i+1} - P_{i-1}, P_{pub})^{a_i} = \frac{e(a_iP_{i+1}, P_{pub})}{e(a_iP_{i-1}, P_{pub})} = \frac{e(a_ia_{i+1}P, P_{pub})}{e(a_{i-1}a_iP, P_{pub})}$. Now, we use the symbol $\Gamma_{i,i+1}$ to substitute $e(P, P_{pub})^{a_ia_{i+1}}$. Hence, each D_i can be written into $D_i = \frac{\Gamma_{i,i+1}}{\Gamma_{i-1,i}}$ for $i = 1, 2, \ldots, n$ and the group session key $SK = (\Gamma_{n,1})^n \cdot D_1^{n-1} \cdot D_2^{n-2} \cdots D_{n-1}$, where $(\Gamma_{n,1})^n = e(P, P_{pub})^{na_na_1} = e(a_1P_n, P_{pub})^n$.

Giving the adversary $\mathcal{A}$ and considering the following algorithm $\mathcal{D}$ which takes $P_a = aP,\ P_b = bP,\ P_c = cP \in G_1$ as input for some $a, b, c \in Z_q^*$. D generates $(\mathcal{T}, SK)$ according to the distribution $Dist^1$ and then outputs whatever $\mathcal{A}$ outputs after running $\mathcal{A}(\mathcal{T}, SK)$. The $Dist^1$ is defined as follows:

$$Dist^1 = \left\{\begin{array}{l} w_1, \ldots, w_{n-2}, a_1, \ldots, a_n,\ h_1, \ldots, h_n, r_1, \ldots, r_n, k_1, \ldots, k_n \leftarrow Z_q^*;\\ P_1 = a_1P, \ldots, P_n = a_nP;\\ V_1 = a_1H_G(ID_1) + h_1DID_1, \ldots, V_n = a_nH_G(ID_n) + h_nDID_n;\\ \Gamma_{1,2} = g_{sab} \in G_2,\ \Gamma_{2,3} = e(P_b, P_{pub})^{w_1} \quad \text{for } j = 3 \text{ to } n-1;\\ \Gamma_{j,j+1} = e(P, P_{pub})^{w_{j-2}w_{j-1}};\ \Gamma_{n,1} = e(P_a, P_{pub})^{w_{n-2}};\\ D_1 = \frac{\Gamma_{1,2}}{\Gamma_{n,1}}, \ldots, D_n = \frac{\Gamma_{n,1}}{\Gamma_{n-1,n}};\\ \alpha_1 = r_1P, \ldots, \alpha_n = r_nP;\ \beta_1 = r_1(P_2 - P_n), \ldots,\\ \beta_n = r_n(P_1 - P_{n-1});\\ \gamma_1 = r_1P_1 + k_1a_1P_{pub}, \ldots, \gamma_n = r_nP_n + k_na_nP_{pub};\\ \mathcal{T} = (P_1, \ldots, P_n, V_1, \ldots, V_n, D_1, \ldots, D_n, \alpha_1, \ldots, \alpha_n,\\ \qquad \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_n);\\ SK = (\Gamma_{n,1})^n \cdot D_1^{n-1} \cdot D_2^{n-2} \cdots D_{n-1}\colon (\mathcal{T}, SK) \end{array}\right\}.$$

By the above distribution, let $\Gamma_{1,2} = e(P, P_{pub})^{ab} = e(P, P)^{sab}$, we refer to the resulting distribution as $Dist^1_{DBDH}$. It is obvious the distribution $Dist^1_{DBDH}$ is identical to *Real*. Because of

$$\begin{aligned}SK &= (\Gamma_{n,1})(\Gamma_{1,2})(\Gamma_{2,3}) \cdots (\Gamma_{n-2,n-1})(\Gamma_{n-1,n})\\&= e(P_a, P_{pub})^{w_{n-2}} \cdot e(P, P)^{sab} \cdot e(P_b, P_{pub})^{w_1} \cdots e(P, P_{pub})^{w_{n-4}w_{n-3}}\\&\quad \times e(P, P_{pub})^{w_{n-3}w_{n-2}}\\&= e(P, P)^{sw_{n-2}a + sab + sbw_1 + \cdots + sw_{n-4}w_{n-3} + sw_{n-3}w_{n-2}}.\end{aligned}$$

Next, we again examine the distribution $Dist^1$. Let $\Gamma_{1,2} = e(P_c, P_{pub}) = e(P,P)^{sc}$ for some $c \neq ab \in Z_q^*$. We refer to the resulting distribution as $Dist^1_{Random}$. It is obvious the distribution $Dist^1_{Random}$ is identical to $Fake_1$. Because of

$$\begin{aligned} SK &= (\Gamma_{n,1})(\Gamma_{1,2})(\Gamma_{2,3})\cdots(\Gamma_{n-2,n-1})(\Gamma_{n-1,n}) \\ &= e(P_a, P_{pub})^{w_{n-2}} \cdot e(P,P)^{sc} \cdot e(P_b, P_{pub})^{w_1} \cdots e(P, P_{pub})^{w_{n-4}w_{n-3}} \\ &\quad \times e(P, P_{pub})^{w_{n-3}w_{n-2}} \\ &= e(P,P)^{sw_{n-2}a+sc+sbw_1+\cdots+sw_{n-4}w_{n-3}+sw_{n-3}w_{n-2}}. \end{aligned}$$

Therefore, we obtain

$$\begin{aligned} &\big|\Pr[(\mathcal{T}, SK) \leftarrow \mathit{Real}\colon \mathcal{A}(\mathcal{T}, SK) = 1] \\ &\quad -\Pr[(\mathcal{T}, SK) \leftarrow \mathit{Fake}_1\colon \mathcal{A}(\mathcal{T}, SK) = 1]\big| \leqslant \varepsilon(t). \end{aligned}$$

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 $\mathcal{A}$ running in time t, we can get the following equations:

$$\begin{aligned} &\big|\Pr[(\mathcal{T}, SK) \leftarrow \mathit{Fake}_1\colon \mathcal{A}(\mathcal{T}SK) = 1] \\ &\quad - \Pr[(\mathcal{T}, SK) \leftarrow \mathit{Fake}_2\colon \mathcal{A}(\mathcal{T}, SK) = 1]\big| \leqslant \varepsilon(t), \\ &\vdots \\ &\big|\Pr[(\mathcal{T}, SK) \leftarrow Fake_{n-1}\colon \mathcal{A}(\mathcal{T}, SK) = 1] \\ &\quad -\Pr[(\mathcal{T}, SK) \leftarrow Fake_n\colon \mathcal{A}(\mathcal{T}, SK) = 1]\big| \leqslant \varepsilon(t). \end{aligned}$$

This implies

$$\begin{aligned} &\big|\Pr[(\mathcal{T}, SK) \leftarrow \mathit{Real}\colon \mathcal{A}(\mathcal{T}, SK) = 1] \\ &\quad - \Pr[(\mathcal{T}, SK) \leftarrow Fake_n\colon \mathcal{A}(\mathcal{T}, SK) = 1]\big| \leqslant n\varepsilon(t). \end{aligned}$$

In the distribution $Fake_n$, the values $d_{1,2}, d_{2,3}, \ldots, d_{n-1,n}, d_{n,1}$ are constrained by $\mathcal{T}$ according to the following n equations: $\log_g D_1 = s \cdot (d1, 2 - d_{n,1}), \log_g D_2 = s \cdot (d2, 3 - d_{1,2}), \ldots, \log_g D_n = s \cdot (d_{n,1} - d_{n-1,n})$, where $g = e(P,P)$. Only $n-1$ of there equations are linear independent. Due to $SK = e(P,P)^{sd_{1,2}+sd_{2,3}+\cdots+sd_{n,1}}$, we have $\log_g SK = sd_{1,2}+sd_{2,3}+\cdots+sd_{n,1}$. Since this final equation is linear independent from the set of equations above, the value of SK is independent of transcript $\mathcal{T}$. This implies that for a computationally-unbounded adversary $\mathcal{A}$:

$$\Pr\big[(\mathcal{T}, SK_0) \leftarrow \mathit{Fake}_n;\ SK_1 \leftarrow G_2;\ b \leftarrow \{0,1\} \mid \mathcal{A}(\mathcal{T}, SK_b) = 1\big] = 1/2.$$

Since $\varepsilon(t) = Adv^{DBDH}_{G_1,G_2,e}(t)$, we have the result that the advantage of $\mathcal{A}$ on the event $\neg \mathit{Forge}$ is bounded by $2n \cdot Adv^{DBDH}_{G_1,G_2,e}(t)$. Hence, we have

$$Adv^{AGKE-fs}_{\Psi}(t, 1, q_s) \leqslant 2n \cdot Adv^{DBDH}_{G_1,G_2,e}(t) + Adv^{forge}_{\Psi}(t).$$

For the case of $q_{ex} > 1$, a standard hybrid argument immediately shows that

$$Adv_{\Psi}^{AGKE-fs}(t, q_{ex}, q_s) \leqslant 2nq_{ex} \cdot Adv_{G_1,G_2,e}^{DBDH}(t) + Adv_{\Psi}^{forge}(t).$$

Under the decision bilinear Diffie–Hellman (DBDH) assumption, the advantage $Adv_{G_1,G_2,e}^{DBDH}(t)$ is negligible; By Theorem 1, the advantage $Adv_{\Psi}^{forge}(t)$ is negligible. Thus, the presented ID-based AGKE protocol is a secure AGKE providing forward secrecy.

[Insider attacks and malicious participant]

In our proposed protocol, each participant can confirm whether the broadcast values are correctly computed by other participants, called *confirmed computation property*. We use this property to accomplish *implicit key confirmation* so that the proposed protocol can also achieve *key agreement*. Hence, we first prove that our protocol provides the *confirmed computation property*.

In order to explain our protocol provides the *confirmed computation property*, we first give a note. In the proposed protocol, each participant U_i use its secret a_i, a public value $(P_{i+1} - P_{i-1}) = (a_{i+1} - a_{i-1}) \cdot P = P_{bi}$ for $(a_{i+1} - a_{i-1}) \in {}_RZ_q^*$, and public parameters to produce a tuple σ_i of some computation values including a special value D_i. If the verification equations "hold", other participants can confirm that D_i is produced by U_i with the secret a_i and equals to $e(P, P)^{sa_i(a_{i+1} - a_{i-1})}$. In other words, given an identity ID_i and a public value $P_{bi} = (P_{i+1} - P_{i-1})$ to an adversary $\mathcal{A}$, $\mathcal{A}$ with the secret a_i is unable to produce two different valid tuples $(ID_i, P_{bi}, \sigma_{i1})$ and $(ID_i, P_{bi}, \sigma_{i2})$ with two different values D_i. Hence, if no probabilistic polynomial time adversary $\mathcal{A}$ with the secret a_i has a non-negligible advantage to produce a valid tuple including a specific value $D_i \neq e(P, P)^{sa_i(a_{i+1} - a_{i-1})}$ on the inputs $P_{bi} = P_{i+1} - P_{i-1} \in G_1$, then the proposed ID-based AGKE protocol provides the *confirmed computation property*.

Lemma 2. *In the random oracle model and under the computational Diffie–Hellman assumption, no probabilistic polynomial time adversary $\mathcal{A}$ with the secret a_i has a non-negligible advantage can produce a valid tuple including specific values $D_i \neq e(P, P)^{sa_i(a_{i+1} - a_{i-1})}$ on the input $P_{i+1} - P_{i-1} \in G_1$.*

Proof. Without loss of generality, let the input $P_{bi} = P_{i+1} - P_{i-1} = (a_{i+1} - a_{i-1}) \cdot P$ for $(a_{i+1} - a_{i-1}) \in {}_RZ_q^*$. By contradiction proof, assume that the adversary $\mathcal{A}$ with the secret a_i can take random values r_i to produce a valid tuple $(ID_i, P_{bi}, \sigma_i = (D_i, \alpha_i, \beta_i, \gamma_i))$, where $D_i \neq e(P, P)^{sa_i(a_{i+1} - a_{i-1})}$. Certainly, $\mathcal{A}$ may use the same a_i and r_i to generate another valid tuple $(ID_i, P_{bi}, \sigma_i' = (D_i', \alpha_i, \beta_i, \gamma_i'))$ in the proposed scheme: $D_i' = e(P_{i+1} - P_{i-1}, P_{pub})^{a_i}$, $\alpha_i = r_i \cdot P, \beta_i = r_i \cdot (P_{i+1} - P_{i-1})$, $k_i' = H_2(PID \| ID_i \| D_i' \| S, P_{i+1} - P_{i-1}, \alpha_i, \beta_i)$, $\gamma_i' = r_i \cdot P_i + k_i' a_i \cdot P_{pub}$, where $S = P_1 \| P_2 \| \cdots \| P_n$.

Since two tuples $(ID_i, P_{bi}, \sigma_i = (D_i, \alpha_i, \beta_i, \gamma_i))$ and $(ID_i, P_{bi}, \sigma_i' = (D_i', \alpha_i, \beta_i, \gamma_i'))$ are valid, they satisfy following verification equations

$$e(P, \gamma_i) = e(P_i, \alpha_i + k_i \cdot P_{pub}) \quad \text{and} \quad e(P_{i+1} - P_{i-1}, \gamma_i) = e(\beta_i, P_i) \cdot D_i^{k_i};$$

$$e(P,\gamma_i') = e(P_i, \alpha_i + k_i' \cdot P_{pub}) \quad \text{and} \quad e(P_{i+1} - P_{i-1}, \gamma_i') = e(\beta_i, P_i) \cdot (D_i')^{k_i'},$$

where k_i is a hash value from the *Hash* oracle, and $k_i' = H_2(PID \| ID_i \| D_i' \| S, P_{i+1} - P_{i-1}, \alpha_i, \beta_i)$. Both the above equations

$$e(P,\gamma_i) = e(P_i, \alpha_i + k_i \cdot P_{pub}) \quad \text{and} \quad e(P,\gamma_i') = e(P_i, \alpha_i + k_i' \cdot P_{pub})$$

hold simultaneously. Given a CDH tuple $(P, P_i, P_{pub}) = (P, xP, yP)$ for some $x, y \in Z_q^*$, using the same method in Lemma 1, the value xyP will be derived from $(\gamma_i - \gamma_i')/(k_i - k_i')$. It is a contradiction for the computational Diffie–Hellman assumption. Therefore, we have $\gamma_i = \gamma_i'$ and $k_i = k_i'$. It is a contradiction for the Hash oracle assumption in the random oracle model.

Similarly, both the above equations

$$e(P_{i+1} - P_{i-1}, \gamma_i) = e(\beta_i, P_i) \cdot D_i^{k_i} \quad \text{and} \quad e(P_{i+1} - P_{i-1}, \gamma_i') = e(\beta_i, P_i) \cdot (D_i')^{k_i'}$$

hold simultaneously. According to $\gamma_i = \gamma_i'$ and $k_i = k_i'$, it is obvious that $(D_i/D_i')^{k_i} = 1$. Thus, we have $D_i = D_i'$. It is a contradiction for $D_i \neq e(P,P)^{sa_i(a_{i+1} - a_{i-1})}$. Hence, no adversary $\mathcal{A}$ with the secret a_i can produce a valid tuple $(ID_i, P_{bi}, \sigma_i = (D_i, \alpha_i, \beta_i, \gamma_i))$ such that $D_i \neq e(P,P)^{sa_i(a_{i+1} - a_{i-1})}$, where $P_{bi} = P_{i+1} - P_{i-1} \in G_1$. Therefore, the proposed ID-based AGKE protocol provides the confirmed computation property.

By Lemma 2, we have shown that our proposed ID-based AGKE protocol provides the *confirmed computation property*. This means that each participant U_i can confirm that each D_j is produced by U_j with the secret a_j and equals to $e(P,P)^{sa_j(a_{j+1} - a_{j-1})}$ for $j = 1, 2, \ldots, n$ and $j \neq i$. Hence, each U_i can compute the same group session key $SK = e(a_i \cdot P_{i-1}, P_{pub})^n \cdot D_i^{n-1} \cdot D_{i+1}^{n-2} \cdots D_{i-2}$ for $i = 1, 2, \ldots, n$. In this situation, the proposed ID-based AGKE protocol provides *implicit key confirmation* and achieves *key agreement*. According to Katz–Shin's security model (Katz and Shin, 2005), we obtain that our proposed protocol is secure against insider attacks in Theorem 3.

Theorem 3. *In the random oracle model and under the computational Diffie–Hellman as well as the decision bilinear Diffie–Hellman assumptions, the proposed ID-based AGKE protocol is secure against insider attacks.*

Proof. As mentioned in Section 3 and Katz–Shin's (2005) security model, an ID-based AGKE protocol is secure against insider attacks, if the following three conditions hold: (1) *the protocol is a secure AGKE*; (2) *it is secure against insider impersonation attack*; (3) *it provides key agreement*. By Theorems 1 and 2, we have proven that the proposed ID-based AGKE protocol is secure against insider impersonation attack and a secure AGKE and, respectively. By Lemma 2, we have proven that it provides key agreement. Thus, our proposed protocol is an secure ID-based AGKE one resistant to insider attacks.

6. Performance Analysis and Comparisons

In this section, we would like to analyze the computational cost of the proposed ID-based AGKE protocol with identifying malicious participants. Then, we compare our protocol with the previously proposed non-ID-based and ID-based AGKE protocols in terms of computational cost, round number and security properties. For convenience to evaluate the computational cost, we only consider some time-consuming operations and define the following notations:

- TG_e: The time of executing a bilinear map operation e: $G_1 \times G_1 \to G_2$.
- 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: $\{0,1\}^* \to G_1$.
- T_{exp}: The time of executing an exponentiation operation over a finite field F_p, where p is a large prime.
- T_{inv}: The time of executing an inverse operation over a finite field F_p.
- T_{mul}: The time of executing a multiplication operation over a finite field F_p.

Here, we first analyze the computational cost of the proposed protocol. In Round 1, $3TG_{\mathrm{mul}} + TG_H$ is required for computing (P_i, V_i). In Round 2, each participant requires $TG_e + 6TG_{\mathrm{mul}} + 2TG_H$ to verify $(ID_{i-1}, P_{i-1}, V_{i-1})$ and $(ID_{i+1}, P_{i+1}, V_{i+1})$, as well as to compute $(D_i, \alpha_i, \beta_i, \gamma_i)$. In the group session key computation phase, it requires $(3n-1)TG_e + nTG_{\mathrm{mul}}$ to verify all $(ID_j, D_j, \alpha_j, \beta_j, \gamma_j)$ and compute the common group key SK. As a result, $(3n+3)TG_e + (n+9)TG_{\mathrm{mul}} + 3TG_H$ is required for each participant in our protocol.

In Table 1, we compare our protocol with three previous AGKE protocols that include Tseng's protocol (Tseng, 2007), Choi *et al.*'s AGKE protocol (Choi *et al.*, 2008) and Choi *et al.*'s AGKE protocol with the UC-compiler in terms of public-key setting, round

Table 1

Comparisons between our protocol and the previously proposed non-ID-based/ID-based AGKE protocols

	Tseng's protocol (2007)	Choi *et al.*'s AGKE (Choi *et al.*, 2008)	Choi *et al.*'s AGKE with the UC-compiler	Our protocol
Public-key setting	Non-ID-based	ID-based	ID-based	ID-based
Rounds	2	2	3	2
Computational cost for each participant	$(8n-2)T_{\mathrm{exp}}+$ $(n+1)T_{\mathrm{inv}}+$ $(n+1)T_{\mathrm{inv}}$	$6TG_e+$ $(n+11)TG_{\mathrm{mul}}+$ $(n+3)TG_H$	$(6n-4)TG_e$ $+(3n+6)TG_{\mathrm{mul}}$ $+(3n-1)TG_H$	$(3n+3)TG_e+$ $(n+9)TG_{\mathrm{mul}}+$ $3TG_H$
Security	Provably secure	Existing attacks	Provably secure	Provably secure
Detecting malicious participants	Yes	No	Yes	Yes
Identifying malicious participants	Yes	No	No	Yes

number, performance and security properties. As depicted in Section 1.2, one recent non-ID-based AGKE protocol with identifying malicious participants was presented by Tseng (2007). Since Tseng's AGKE protocol is non-ID-based, each participant must verify other participants' certificates for participant authentication. The required computational costs for verifying certificates will be added, besides $(8n-2)T_{\exp}+(n+1)T_{\text{inv}}+(2n-2)T_{\text{mul}}$ in Table 1. Although Choi *et al.* claimed that their improved protocol (Choi *et al.*, 2008) is secure against insider colluding attacks, Wu and Tseng (2009) have shown that Choi *et al.*'s AGKE protocol is still insecure against other insider colluding attacks.

As mentioned in Section 1.2, by applying the UC complier presented by Katz and Shin (2005) to Choi *et al.*'s improved AGKE (Choi *et al.*, 2008), it can provide the functionality of detecting malicious participants. However, the UC complier requires one additional round and n signature verifications. Now, let us discuss the computational cost of Choi *et al.*'s ID-based AGKE protocol with the UC-complier. Note that the key construction of Choi *et al.*'s protocol is the same as one of an efficient ID-based signature scheme (Cha and Cheon, 2003). Hence, we assume that the ID-based signature scheme (Cha and Cheon, 2003) is used to Choi *et al.*'s protocol and the UC-complier. Thus, it totally requires $(6n-4)TG_e+(3n+6)TG_{\text{mul}}+(3n-1)TG_H$ for each participant. Note that the resulting protocol provides only detecting malicious participants. It is still unable to find "who are malicious participants".

By Table 1, to our best knowledge, all existing non-ID-based or ID-based group key exchange protocols with identifying malicious participants still require O(n) computational cost. This is a price to pay for providing the functionality of identifying malicious participants. Nevertheless, our protocol has better performance as compared to the previously proposed non-ID-based and ID-based AGKE protocols.

7. Conclusions

In this paper, we have proposed an ID-based authenticated group key agreement protocol with identifying malicious participants. As compared to the previously proposed non-ID-based and ID-based AGKE protocols, our protocol provides not only detecting malicious participants but also identifying "who are malicious participants" in the group key establishment. In the random oracle model and under the CDH as well as DBDH assumptions, we have proven that the presented protocol satisfies Katz–Shin's security model. It means that our presented protocol is a secure AGKE providing forward secrecy and can resist insider attacks.

Acknowledgements. This research was partially supported by National Science Council, Taiwan, R.O.C., under contract No. NSC97-2221-E-018-010-MY3.

References

Bellare, M., Rogaway, P. (1993). Random oracles are practical: a paradigm for designing efficient protocols. In: *Proceedings of ACM CCS'93*, pp. 62–73.

Boneh, D., Franklin, M. (2001). Identity-based encryption from the Weil pairing. In: *Proceedings of CRYPTO'01*, LNCS, Vol. 2139, pp. 213–229.

Bresson, E., Manulis, M. (2008). Contributory group key exchange in the presence of malicious participants. *IET Information Security*, 2(3), 85–93.

Bresson, E., Chevassut, O., Pointcheval, D., Quisquater, J.J. (2001). Provably authenticated group Diffie–Hellman key exchange. In: *Proceedings of ACM CCS'01*, pp. 255–264.

Burmester, M., Desmedt, Y. (2005). A secure and scalable group key exchange system. *Information Processing Letters*, 94(3), 137–14.

Cha, J.C., Cheon, J.H. (2003). An identity-based signature from gap Diffie–Hellman groups. In: *Proceedings of PKC'03*, LNCS, Vol. 2567, pp. 18–30.

Chen, L., Cheng, Z., Smart, N.P. (2007). Identity-based key agreement protocols from pairings. *International Journal of Information Security*, 6(4), 213–241.

Choi, K.Y., Hwang, J.Y., Lee, D.H. (2004). Efficient ID-based group key agreement with bilinear maps. In: *Proceedings of PKC'04*, LNCS, Vol. 2947, pp. 130–144.

Choi, K.Y., Hwang, J.Y., Lee, D.H. (2008). ID-based authenticated group key agreement secure against insider attacks. *IEICE Transactions Fundamentals*, E91-A (7), 1828–1830.

Katz, J., Shin, J.S. (2005). Modeling insider attacks on group key exchange protocols. In: *Proceedings of ACM CCS'05*, pp. 180–189.

Kim, K, Yie, I., Lim, S., Park, H. (2011). A method of finding bad signatures in an RSA-type batch verification. *Informatica*, 22(2), 189–201.

Liu, J., Huang, S. (2010). Identity-based threshold proxy signature from bilinear pairings. *Informatica*, 21(1), 41–56.

Pointcheval, D., Stern, J. (2000). Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3), 361–396.

Ren, Y., Gu, D., Wang, S., Zhang, X. (2010). New fuzzy identity-based encryption in the standard model. *Informatica*, 21(3), 393–408.

Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In: *Proceedings of CRYPTO'84*, LNCS, Vol. 196, pp. 47–53.

Shim, K.A. (2007). Further analysis of ID-based authenticated group key agreement protocol from bilinear maps. *IEICE Trans. Fundamentals*, E90-A (1), 295–298.

Tseng, Y.M. (2005). A robust multi-party key agreement protocol resistant to malicious participants. *The Computer Journal*, 48 (4), 480–487.

Tseng, Y.M. (2007). A communication-efficient and fault-tolerant conference-key agreement protocol with forward secrecy. *Journal of Systems and Software*, 80(7), 1091–1101.

Tseng, Y.M., Wu, T.Y., Wu, J.D. (2009). An efficient and provably secure ID-based signature scheme with batch verifications. *International Journal of Innovative Computing, Information and Control*, 5(11), 3911–3922.

Tseng, Y.M., Wu, T.Y. (2010). Analysis and improvement on a contributory group key exchange protocol based on the Diffie–Hellman technique. *Informatica*, 21(2), 247–258.

Tseng, Y.M., Wu, T.Y., Wu, J.D. (2008). A pairing-based user authentication scheme for wireless clients with smart cards. *Informatica*, 19(2), 285–302.

Tzeng, W.G. (2002). A secure fault-tolerant conference-key agreement protocol. *IEEE Trans. on Computers*, 51(4), 373–379.

Wu, T.Y., Tseng, Y.M. (2009). Comments on an ID-based authenticated group key agreement protocol with withstanding insider attacks. *IEICE Trans. on Fundamentals*, Vol. E92-A(10), 2638–2640.

Wu, T.Y., Tseng, Y.M. (2010). An ID-based mutual authentication and key exchange protocol for low-power mobile devices. *The Computer Journal*, 53(7), 1062–1070.

Yoon, H.J., Cheon, J.H., Kim, Y. (2004). Batch verifications with ID-based signatures. In: *Proceedings of ICISC'04*, LNCS, Vol. 3506, pp. 233–248.

Zhang, F., Chen, X. (2004). Attack on an ID-based authenticated group key agreement scheme from PKC 2004. *Information Processing Letters*, 91(4), 191–193.

Tsu-Yang Wu received the BS and the MS degrees in Department of Applied Mathematics, Tatung University, Taiwan, in 2003 and 2005, respectively. He received the PhD degree in Department of Mathematics, National Changhua University of Education, Taiwan, in 2010. He is currently an assistant professor in School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, China. His research interests include applied cryptography, pairing-based cryptography and information security.

Yuh-Min Tseng 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, IEICE and the Chinese Cryptology and Information Security Association (CCISA). In 2006, his paper obtained the Wilkes Award from *The British Computer Society*. He is also editors of several international journals: *Computer Standards & Interfaces*, *International Journal of Security and Its Applications, Wireless Engineering and Technology* as well as *ISRN Communications.* His research interests include cryptography, network security, computer network and mobile communications.

Tapatumu grįstas grupinis apsikeitimo raktu protokolas identifikuojantis piktavalius vartotojus

Tsu-Yang WU, Yuh-Min TSENG

Tapatumu grįstas grupinis apsikeitimo raktu protokolas vartotojams sugeneruoja bendrąjį raktą ir užtikrina saugų vartotojų grupės ryšį. Neseniai pasiūlytas Choi ir kt. protokolas yra nesaugus nuo galimų vidinių atakų, nes jis nenustato ar tarp grupinio apsikeitimo raktu vartotojų yra piktavalis vartotojas. Be to, protokolas negali nustatyti kuris vartotojas yra piktavalis. Straipsnyje pasiūlytas identifikatoriumi (ID) grįstas protokolas, kuris identifikuoja piktavalius vartotojus. Šiame protokole panaudojamas patvirtinantis požymis, kurio dėka identifikuojami piktavaliai vartotojai. Be to įrodyta, kad pasiūlytas protokolas yra saugus.