Identity-based Constant Round Group Key Exchange Protocol via Secret-Share

C.J. CAO, J.F. MA

Key Laboratory of Computer Networks and Information Security
Xidian University
No.2 South TaiBai Road Xi'an, Shaanxi, 710071
CHINA
chjcao@mail.xidian.edu.cn, jfma@mail.xidian.edu.cn

Abstract: Group key management is one of the basic building blocks in securing group communication. A number of solutions to group key exchange have been proposed, but most of them are not scalable and require a number of rounds linear in the number of group members. We formally present a constant-round Identity-based protocol with forward secrecy for group key exchange, which is provably secure in the security model introduced by Bresson et al. Our protocol focuses on round efficiency and the number of communication round is only two. And, the protocol provides a batch verification technique, which simultaneously verifies the validity of messages from other group participants and greatly improves computational efficiency. Moreover, in our protocol, it is no necessary of always-online key generation center during the execution of the protocol compared to other Identity-based protocols.

Key-Words: - provable security, identity-based, group key exchange, constant round, secret-share.

1 Introduction

In many modern group-oriented and distributed applications, such as distributed simulation, multi-user games, and collaborative tools, scalable and reliable group communication is one of the critical problems. Regardless of the concrete application environment, security services are necessarily required to provide communication privacy and integrity. These are not possible without secure and efficient key distribution. A group key exchange protocol allows a group of participants to establish a common session key which is used to protect the sensible information.

Among the existing authentication systems, asymmetric technologies such as public key infrastructure (PKI) and Identity-based (ID-based) system are commonly adopted. The concept of ID-based cryptosystem was firstly proposed by Shamir[1]. Such a scheme has a unique property that a user's public key can be easily calculated from his identity, while the private key can be calculated for him by a trusted authority called key generation center (KGC). In a typical PKI system, a user should apply for his public key certificate from a certificate authority (CA) and other partners can use this certificate to authenticate the user. Whereas, in an ID-based system, the partner only needs to know the public identity, e.g. e-mail address, of the user. Thus, compared with certificate-based PKI system, an ID-based system greatly simplifies the procedure of key management.

Communication complexity has always been an important issue when designing group key exchange protocols. In 1998, Becker and Wille [2] derived several bounds on group key exchange protocols. Amongst these was the lower bound on the number of rounds which is only one, no matter how many users involved. A protocol that meets this bound would allow all messages to be sent simultaneously in one time unit, as long as parallel messages are possible. Until now no DH generalization is able to meet this bound and also provides forward secrecy, thus Becker and Wille leave it as an open problem whether any contributory group key exchange scheme can meet this bound.

Communication security is another very important issue when we design a group key exchange protocol. Only recently have Bresson, Chevassut, Pointcheval and Quisquater (BCPQ) given the first provably-secure model and protocol [3-5] for group key exchange setting. Their protocol is based on the protocol of Steiner et al.[6], and require n rounds to establish a key among a group of n users. The BCPQ model is an important step and is very helpful in analyzing and designing group key exchange protocols.

The purpose of this paper is to present a constant-round ID-based group key exchange protocol via

secret-share. Becker and Wille considered only unauthenticated key exchange protocols, which are insecure against active adversaries. However, we present a provably secure authenticated group key exchange protocol that exceeds the lower bound by only one and can provide forward secrecy.

1.1 Related Work

Group key exchange. A number of works [6-18] have considered the problem of extending the two-party Diffie-Hellman (DH) protocol [19] to the multiparty setting. A class of generic *n*-party DH protocols is defined in [16] and extended to provide implicit key authentication in [14], and one practical protocol of which is A-GDH.2. A tree based DH group key exchange protocol has been proposed by Kim et al. in [15] which is shown to be secure against passive adversaries. Also several papers have attempted to establish ID-based authenticated key exchange protocol. Joux presented an one-round tripartite key exchange protocol using pairings [9]. But it is vulnerable to "man-in-the-middle" attack. Zhang, Liu and Kim proposed a new ID-based authenticated three-party key exchange protocol, in which the authenticity is assured by a special signature scheme from pairing [11]. Recently an ID-based group key exchange protocol which uses the one way function trees and a pairing is proposed by Reddy et al with informal security analysis [20]. Barua et al. introduced an ID-based multi party key exchange scheme which uses ternary trees [12]. But the protocols of Reddy and Barua have $\lceil \log_2 n \rceil$ and $\lceil \log_3 n \rceil$ communications rounds respectively and are not scalable.

There are two kinds of famous constant-round group key agreement protocols, one is based on the BD scheme which was proposed by Burmester and Desmedt [8], and the other is based on secret sharing scheme. In PKC '04, Choi et al. presented an efficient ID-based group key exchange schemes from bilinear pairings which is an authenticated bilinear variant of BD scheme [13], but soon found to be flawed by Zhang and Chen [21]. Tzeng and Pieprzyk et al. have shown how secret sharing scheme can be exploited as a building block in group key establishment [22, 23]. And Bresson et al. proposed a O practical and simple group key exchange scheme which combines the ElGamal encryption scheme and the secret sharing technique [24]. Nevertheless, in the protocol of Pieprzyk and Li [23], confidence in fresh of the key depends on a random value supplied by a trusted third party, and forward secrecy does not provide. Also the scheme of Tzeng lacks of forward secrecy.

Provable Security for Protocols. The basic idea of proving the security of a protocol in a model in which the parties have a random oracle and then instantiating that oracle with an appropriate cryptographic primitive originates in [25, 26]. In 1993, Bellare and Rogaway proposed a formal model for proving security of protocols in two party setting [27, 28]. A modular approach is presented by Bellare, Canetti and Krawczyk to design and analyze key exchange protocols [29]. The modularity is achieved by applying an authenticator to protocols which have been proven secure in a much simplified adversarial setting where authentication of communication links is not required. Based on these works, Bresson et al. defined a sound formalization for the authenticated group DH key exchange and provide provably secure protocols within this model [3-5]. This is an important step and is very helpful for protocol designer.

1.2 Our Contribution

The purpose of this paper is to present an ID-based group key exchange protocol with provable security with forward secrecy. Our protocol focuses on round efficiency and only needs two rounds. It provides a batch verification technique which simultaneously verifies the validity of messages from other group participants and greatly improves computational efficiency. In addition, the protocol is a contributory key exchange, hence it does not impose a heavy computational burden on a particular party. It should be noted that our protocol provides a methodology to design group key exchange protocol and most secret sharing scheme could be adopted to construct our protocol.

1.3 Outline

The rest of our paper is organized as follows. We review admissible bilinear map, cryptographic assumption and aggregate signature in Section 2 and the BCP model in Section 3. We present our ID-based group key exchange protocol in Section 4, prove its security in section 5 and analyze its efficiency in Section 6 respectively. Finally we conclude in Section 7.

2 Preliminaries

2.1 Admissible Bilinear Map

Let G_1 be a cyclic additive group of prime order q and G_2 be a cyclic multiplicative group of same order q. Let P be an arbitrary generator of G_1. We assume that discrete logarithm problem in both G_1 and G_2 are intractable. A map e: $G_1 \times G_1 \rightarrow G_2$ satisfying the following properties is called an admissible bilinear map:

- Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for all P, $Q \in G_1$ and $a, b \in Z_q^*$.
- Non-degeneracy: if P is a generator of G_1, then $e(P, P)$ is a generator of G_2. I.e. $e(P, P) \neq 1$.
- Computability: There exists an efficient algorithm to compute $e(P, Q)$ for all P, $Q \in G_1$.

2.2 Computational DH (CDH) Problem in G_1

Input: (P, aP, bP) for some $a, b \in Z_q^*$.

Output: abP.

The success probability of any probabilistic polynomial time adversary A in solving CDH problem in G_1 is defined to be:

$$Succ_{\mathcal{A},G_1}^{CDH} = \Pr ob\left[\mathcal{A}(P, aP, bP, abP) = 1 : a, b \in Z_q^*\right]$$

CDH assumption: There exists no algorithm running in expected polynomial time, which can solve the CDH problem with non-negligible probability. Namely, for any probabilistic polynomial time (PPT) adversary A, $Succ_{\mathcal{A},G_1}^{CDH}$ is negligible.[1]

2.3 Aggregate Signature

In the construction of our authenticated protocol, we use the bilinear aggregate signature scheme firstly introduced by Boneh et al. [30]. But the base signature scheme is the ID-based bilinear signature scheme proposed by Hess [31].

An aggregate signature scheme is a digital signature that supports aggregation. Concretely, given n signatures on n distinct messages from n distinct participants, it is possible to aggregate all these signatures into a single short signature. This single signature and the n original messages will convince the verifier that participant u_i indeed signed message m_i. The aggregate signature scheme is formally denoted as $\Lambda = \{G, K, Sig, Ver, ASig, AVer\}$, where $\{G, K, Sig, Ver\}$ is a standard digital signature scheme, which is called the base signature scheme. Here G is a randomized system parameters generator algorithm, K is a randomized key generation algorithm, Sig is a randomized signing algorithm and Ver is a deterministic algorithm. The aggregation signature algorithm and the aggregation verification algorithm are respectively $ASig$ and $AVer$. The aggregate signature is generated as follows:

$$\delta = ASig(\delta_1, \delta_2, \ldots, \delta_n),$$

where δ_i is the signature of message m_i relative to public key PK_i and δ is the single aggregate signature. The verification is done by checking whether

$$Aver(PK_1, PK_2, \ldots, PK_i; m_1, m_2, \ldots, m_n; ASig(\delta_1, \delta_2, \ldots, \delta_n)) = 1.$$

Note that we set the co-GDH gap groups are equivalent, so the computational co-DH and decisional co-DH problems reduce to the standard CDH and DDH problems [32].

3 The Model

The model described in this section is extended from one of Bresson et al. [3-5] which follows the approach of Bellare and Rogaway [27, 28, 33].

3.1 Adversarial Model

Let $U = \{U_1, U_2, \ldots, U_n\}$ and $ID = \{ID_1, ID_2, \ldots, ID_n\}$ be a set of n users and their identities respectively. Each user U_i has a unique identity ID_i, which is known to all the other users, and all these identities are distinct. Each user can execute the protocol multiple times with different partners: this is modeled by allowing each user an unlimited number of *instances* with which to execute the protocol. We denote instance t of U_i, called an oracle, as Π_i^t for an integer $t \in N$.

Initialization. In this phase, each user $U_i \in U$ gets his long-term public and private keys. ID-based protocol requires the following initialization phase.

1) The KGC randomly chooses a secret key $s \in Z_q$ as master key. The KGC computes $P_{pub} = sP$ and publishes it.
2) When each user with identity ID wants to obtain public/private key pair, the KGC uses its master secret key s to compute the corresponding private key S_{ID} and transmit it to the user through a secure channel.

[1] A function $\varepsilon(k)$ is negligible if for every $c>0$ there exists a $k_c>0$ such that for all $k>k_c$, $\varepsilon(k)<k^{-c}$.

Queries. Normally, the security of a protocol is related to the adversary's ability, which is formally modeled by queries issued by the adversary. We assume that a probabilistic polynomial time adversary A can completely control the communications and make queries to any instance. We now explain the capability that each kind of query captures.

- *Extract* (ID_i): This query allows the adversary to get the long-term private key corresponding to ID_i, where $ID_i \notin ID$.
- *Send* ($\prod_i^t$, M): This query allows the adversary to make the user ID_i run the protocol normally and send message M to instance $\prod_i^t$ which will return a reply.
- *Reveal* ($\prod_i^t$): This query models the adversary's ability to find session group keys. If an oracle has accepted, holding a session key K, then K is returned to the adversary. Note that we say that an oracle accepts when it has enough information to compute a session key. At any time an oracle can accept and it accepts at most once in executing an operation. As soon as an oracle accepts in executing an operation, the session key is defined.
- *Corrupt* (ID_i): This query models the attacks revealing the long-term private key S_i. This does not output any internal data of ID_i.
- *Test* ($\prod_i^t$): This query models the semantic security of a session key. This query is allowed only once by the adversary. A random bit b is chosen; if $b = 1$ then the session key is returned, otherwise a random value is returned.

In this model we consider two types of adversaries according to their attack types. The attack types are simulated by the queries issued by the adversaries. A passive adversary is allowed to issue Reveal, Corrupt, and Test queries, while an active adversary is additionally allowed to issue Send and Extract queries.

3.2 Security Notions

Definition 1 *(Partner IDS)*: Partner identities for instance $\prod_i^t$ which consists of the users (including ID_i himself) with whom $\prod_i^t$ intends to establish a session key. The Partner IDS of instance $\prod_i^t$ is denoted by PID ($\prod_i^t$)**.**

Definition 2 *(Session IDS)*: The Session IDS is a protocol specified function of all communication sent and received by $\prod_i^t$, which is denoted by SID ($\prod_i^t$).

Definition 3 *(Freshness)*: An oracle is called fresh (or holds a fresh key) if the following two conditions are satisfied. First, nobody in U has ever been asked for a Corrupt query from the beginning of the game. Second, in the current operation execution, $\prod_i^t$ has accepted and neither U_i nor his partners have been asked for a Reveal query.

Definition 4 *(Group key security, GK security)*: We say that event *Succ* occurs if the adversary issues Test query to a fresh oracle and correctly guesses the bit b (distinguishing the key from a random string). The advantage of an adversary A in attacking protocol P is defined as $Adv_A^P(k)=|2\Pr[Succ]-1|$.

A protocol P is GK secure, if the following two properties are satisfied:

- *Consistency*: In the presence of an adversary, all partner oracles accept the same key.
- *Secrecy*: For any PPT adversary A, $Adv_A^P(k)$ is negligible.

Definition 5 *(Perfect Forward Secrecy)*: A protocol provides perfect forward secrecy if an adversary does not get non-negligible knowledge information about session keys previously established when making Corrupt queries to all group members. We define $Adv_A^P(t, q_s, q_h)$ to be the maximal advantage of any active adversary attacking protocol P, running in time t and making q_s Send queries and q_h Hash queries.

Note that we do not define any notion of explicit authentication or, equivalently, confirmation that the other members of the group have computed the common key. However, explicit authentication in our protocol can be achieved at little additional cost. Previous work shows how to achieve explicit authentication for any group authenticated key exchange protocol using one additional round and minimal extra computation [5].

4 The Protocol ID-SS

In this section we present an ID-based authenticated group key exchange protocol via secret-share, which is denoted as ID-SS(Secret-Share). The trusted KGC is involved in this protocol. In the following, we assume that (1) the underlying group communication system is resistant to fail-stop failures, which means that the system should provide a consistent membership view to all group members and reliable and causally ordered multicasts; (2) unicast and multicast are reliable. We assume that any user can broadcast messages to others in the broadcast network. (3) there exists a authenticated secure channel between the user and (on-line or off-line) KGC for the distribution of the long-term private key.

***System setup*:** Given the security parameter q, the KGC chooses groups G_1 and G_2 of prime order q, a generator P of G_1, a bilinear map e: $G_1 \times G_2 \rightarrow G_2$. Let H_1: $\{0,1\}^* \rightarrow G_1$ be a map-to-point hash function, H_2: $G_1 \times Z_q \times \{0,1\}^* \times \{0,1\}^* \rightarrow Z_q$, H_3: $G_1 \rightarrow Z_q$ be other two hash functions and H_4 be a key derivation function. H_1, H_2, H_3 and H_4 are considered as random oracles. Also the KGC randomly selects $s \in Z_q$ as the master secret key and computes $P_{pub} = sP \in G_1$ that is made public. Then KGC publishes the following system parameters:

$$\{ e, G_1, G_2, q, P, P_{pub}, H_1, H_2, H_3, H_4 \}.$$

***Extract*:** Given a public identity $ID \in \{0,1\}^*$, The KGC computes $Q_{ID} = H_1(ID) \in G_1$ and associated private key $S_{ID} = sQ_{ID} \in G_1$ that is transmitted to the user.

Let $U=\{U_1, U_2, \ldots, U_n\}$ be a set of users who want to establish a common session key and ID_i be the identity of U_i. Then the public and private key pair of U_i is $(ID_i, S_i = sQ_i)$. Now we describe the protocol in detail.

Round* 1: *Secret Generation

Each participant U_i picks randomly $r_i \in Z_q^*$, computes and broadcasts $O_i = r_iP$.

Round* 2: *Secret Distribution

On receiving O_j, each participant U_i picks randomly $K_i \in Z_q$ and computes a polynomial

$$f_i(x) = K_i + a_{i_1} x + a_{i_2} x^2 + \cdots + a_{i_n} x^{n-1}$$

passing points $(j, H_3(r_iO_j))$, $1 \le j \le n$, $j \ne i$ and $(0, K_i)$. Then computes

$$P_{ij} = f_i(n+j),\ 1 \le j \le n,\ j \ne i;$$
$$P_i = P_{i1} \| P_{i2} \| \ldots \| P_{in};$$
$$O = O_1 \| O_2 \| \ldots \| O_n;$$
$$h_i = H_2(P_i \| O \| K_i \| T_s \| G_{ID});$$
$$\delta_i = r_iP_{pub} + h_iS_i$$

and broadcasts (P_{ij}, δ_i), where T_s is the time stamp and G_{ID} is the identity of the group.

***Key Computation*:**

- *Subkey computation*: On receiving (P_{jl}, δ_l), $1 \le l \le n$, $l \ne j \ne i$, each participant U_i computes polynomial $f_j'(x)$ of degree n that passes $(n + l, P_{jl})$ and $(i, H_3(r_iO_j))$. Then $K_{ij} = f_j'(0)$.
- *Aggregate signature verification*: Each participant U_i firstly checks

$$e(\sum_{j=1}^{n} \delta_j, P) = e(\sum_{j=1}^{n} (O_j + h_jQ_j), P_{pub}).$$

- *Key Computation*: If the above aggregate signature is verified successfully, U_i computes

$$K = H_4(K_{i_1} + K_{i_2} + \cdots + K_{i_n})$$
$$= H_4(K_1 + K_2 + \ldots + K_n).$$

5 Security Analysis of ID-SS

Theorem 1. Suppose that the hash functions H_1, H_2, H_3, H_4 are random oracles. Then the protocol ID-SS is a secure GK protocol providing perfect forward secrecy under the CDH assumption. Concretely,

$$Adv_A^{ID-SS}(t, q_s, q_h)$$
$$\le 2 \cdot n \cdot Succ_{\Gamma}^{Forgery_{\Lambda}}(t) + 2 \cdot l \cdot q_h \cdot Succ_{\Psi}^{CDH}(t)$$

Proof. Firstly, we prove the correctness of the protocol. In other words, if all users follow the process of the protocol, they can compute a common group key. Because of $r_iO_j = r_ir_jP = r_jO_i$, user U_i can compute the polynomial $f_j(x)$ passing $(n + l, P_{jl})$, $1 \le l \le n$ and (i, r_iO_j) according to the messages related to user U_j. Then U_i computes $K_j = f_j(0)$. By verifying aggregate signature δ, U_i can check whether K_j is correct or not. So all participants can derive the same group key $K = H_4(K_1 + K_2 + \ldots + K_n)$.

Secondly, we prove that the protocol is a GK secure protocol in the presence of an adversary A. (1) assuming A modifies the flows, build a forger and (2) assuming that A does not modify the flows, build a CDH-solver. So we can construct a forger Γ and a CDH attacker Ψ from A respectively.

Forger Γ. Assume that A breaks the protocol ID-SS by forging a signature at least with the probability φ. We can construct a forger Γ that generates a valid message pair (ID, m, δ) from A. Γ receives ID as the input and accesses a (public) signing oracle. Γ randomly picks $i \in [1, n]$ and honestly generates all other public and private keys for the system. However for user U_i, Γ sets ID as U_i's public key. Then Γ starts running A as a subroutine and answers the oracle queries made by A as follows:

- When A makes Send(*, m) queries, Γ responds in a straightforward way. When A makes Send(*, m, δ) queries, Γ responds in a straightforward way using long-term keys to sing the flows except if A makes the query of the form Send $(\prod_j^t, m, \delta)$. If this occurs, Γ goes through the signing oracle and stores the response in a variable α.
- When A makes a Reveal query, Γ gives the session key to A.
- When A makes a Corrupt query, Γ answers in a straightforward way except if A makes the query of Corrupt (ID). If this occurs, Γ stops and outputs "Fail".
- When A makes a Hash query, Γ answers as a random oracle in a straightforward way.
- When A makes a Test query, since all the accepted session keys are known from Reveal

queries, the query can be answered with the correct session key.

If A has already issued the query of Send ($\prod_j^t$, m, δ) where δ is a valid signature on m with respect to ID and (m, δ) $\notin \alpha$, then Γ stops and outputs (m, δ) as a forgery. Otherwise, Γ simply aborts. So the probability $Succ_{\Gamma}^{Forgery_{\Lambda}}(t)$ of Γ outputting a forgery is the product of the probability that A generates a valid signature and the probability that A correctly guesses the value of i:

$$Succ_{\Gamma}^{Forgery_{\Lambda}}(t) \geq \varphi/n \quad (1)$$

CDH-attacker Ψ. Next, we assume that A breaks the protocol ID-SS without generating a forgery of signature. Thus from A, we can construct a CDH-attacker Ψ that breaks the protocol by solving an instance of the CDH problem.

Let l be an upper bound on the number of sessions invoked by A, then Ψ randomly chooses and $\gamma \in [1, l]$ representing a guess that as to which query of A activates the instance for which A will ask its Test query.

Ψ receives an instance (P, aP, bP) of the CDH problem as input and randomly selects i, $j \in [1, n]$. Then Ψ starts running A as a subroutine and answers the oracle queries made by A. We now describe the simulation of the oracle queries of A in detail.

- When A makes a Send(*, m) query, Ψ proceeds as in protocol ID-SS using a random value except if the query is Send($\prod_i$, m) or Send($\prod_j$, m) query in the γ^{th} session. If this occurs, Ψ sets $O_i = aP$, $O_j = bp$. When A makes Send(*, m, δ) queries, Ψ responds in a straightforward way using long-term keys to sing the flows except if the query is Send($\prod_i$, m, δ) or Send($\prod_j$, m, δ) query in the γ^{th} session. If this occurs, Ψ responds using a random value and long-term keys to sing the flows.
- When A makes a Corrupt query, Ψ answers with the corresponding long-term private key in a straightforward way.
- When A makes a Reveal query, Ψ answers in a straightforward way except if the session key is of the γ^{th} session. In the latter case, Ψ stops and outputs "Fail".
- When A makes a Hash query, Ψ answers as a random oracle in a straightforward way.
- When A makes a Test query, Ψ answers with a random string.

Since Ψ knows all the keys except for one execution of ID-SS, this simulation is perfectly indistinguishable from an execution of the real protocol ID-SS.

At some stage, A completes and returns a value b'. The probability that Ψ correctly guesses on which session key A will make the Test query is the probability that Ψ correctly guesses the value γ. That is $\mu=1/l$.

Let $askH$ be the event that A makes a Hash query on (K_1+K_2+…+K_n) and $Forge$ be the event that A forges a signature with regard to some participant's long-term public key. We emphasize that, in the random oracle model, A cannot get any advantage on a random value without asking for it. The success probability of Ψ is the probability that A asks the correct value to the hash oracle multiplied by the probability that Ψ correctly chooses the Hash query and multiplied by the probability that Ψ correctly guesses the value γ. That is:

$$Succ_{\Psi}^{CDH}(t) \geq \Pr[askH]/q_H l \quad (2)$$

Finally, we have:

$$\begin{aligned}
&\Pr[b=b'] \\
&= \Pr[b=b'|Forge]\Pr[Forge] + \Pr[b=b'|\neg Forge]\Pr[\neg Forge] \leq \Pr[b=b'|Forge] + \Pr[b=b'|\neg Forge]\Pr[\neg Forge] \\
&\leq \varphi + \Pr[b=b'|\neg Forge]\Pr[\neg Forge] \\
&\leq \varphi + \Pr[\neg Forge \wedge askH]\Pr[b=b'|\neg Forge \wedge askH] + \Pr[\neg Forge \wedge \neg askH]\Pr[b=b'|\neg Forge \wedge \neg askH] \\
&= \varphi + \Pr[b=b'|\neg Forge \wedge askH]\Pr[\neg Forge \wedge askH] + 1/2 \\
&\leq \varphi + \Pr[\neg Forge \wedge askH] + 1/2 \\
&\leq \varphi + \Pr[askH] + 1/2
\end{aligned} \quad (3)$$

Then from the definition $Adv_A^P(k) = |2\Pr[Succ]-1|$ and above three equations, we can get the result as follows:

$$\begin{aligned}
&Adv_A^{ID-SS}(t, q_s, q_h) \\
&\leq 2 \cdot n \cdot Succ_{\Gamma}^{Forgery_{\Lambda}}(t) + 2 \cdot l \cdot q_h \cdot Succ_{\Psi}^{CDH}(t)
\end{aligned}$$

We next show that the authentication scheme Λ is secure against existential forgery on adaptively chosen ID attack.

Lemma 1. Let G_1 be an additive group with order q and the map-to-point hash function H_1 be a random oracle. We assume that the PPT forger A

breaks the bilinear aggregate signature scheme Λ for an adaptively chosen *ID* with advantage ε_0 and running time t_0. Suppose that A makes at most q_{H_1} queries to the hash function H_1. Then from A, we can construct a PPT forger B for a given *ID* with advantage $\varepsilon_0 \leq \varepsilon_1(1-1/q)/q_{H_1}$ and running time $t_1 \leq t_0$.

Lemma 2. Let the hash function H_1, H_2 be random oracles. Suppose that B is a PPT forger for a given *ID* with advantage $\varepsilon_1 \geq 10q_{H_1}(q_s+q_{H_2})/(q-1)$ and running time t_1. Suppose that B makes at most q_{H_1}, q_{H_2}, q_s and q_{ex} queries to the H_1, H_2, Send and Extract oracles respectively. Then from B, we can construct a PPT attacker C that can solve the CDH problem within time $t_2 \leq 120686\, q_{H_2} t_1/\varepsilon_1$.

The security analysis of Lemma 1 and Lemma 2 is similar to that of [13], for space limitation, we omit the proof of them. Then, we can directly obtain the following theorem from the above two Lemmas.

Theorem 2. Let H_1, H_2 be random oracles. Then the bilinear aggregate signature scheme Λ on G_1 is secure against existential forgery on adaptively chosen ID attack under the CDH assumption.

6 Comparison of Group Key Exchange Protocols

In this section, we compare our protocol to other well-known group key exchange protocols in efficiency and some desirable attributes. A class of generic n-party DH (GDH) protocols is defined in [16] and extended to provide implicit key authentication in [14]. One practical protocol of which is A-GDH.2. A tree based DH group key exchange protocol (TGDH), which is shown to be secure against passive adversaries, has been proposed by Kim, et al in [15]. Also several papers have attempted to establish ID-based authenticated key exchange protocol, such as the binary tree based ID-BT [10], the ternary tree based ID-TT [12] and the authenticated bilinear variant of Burmester and Desmedt scheme ID-BD [13], etc.

Efficiency of a protocol is measured by communication and computation costs. Communication cost involves counting total number of rounds needed and total number of messages transmitted through the network during a protocol execution. Computation cost counts total number of pairing-computations, exponentiations or scalar multiplications, etc. The efficiency comparison of group key exchange protocols is shown in Table 1. The notations in this table are described as follows:

- Round number: The total number of rounds.
- Message number: The total number of messages sent by users.
- Exponentiation / Scalar Multiplication (E /SM): The total number of exponentiations and scalar multiplications.
- Pairing: The total number of pairing-computations.

The efficiency of non-constant round protocols ID-BT, ID-TT, TGDH and A-GDH.2 is reduced obviously with the increase of value n. So they are not scalable. ID-BT and TGDH did not give formal security analysis. In particular, ID-BT is based on the two-party key exchange protocol proposed by Smart [34], but it was shown that the protocol did not provide the forward secrecy by Shim [35]. As mentioned above, *ID-BD cannot resist collusion attack*. Both binary tree based protocol ID-BT and ternary tree based protocol ID-TT have $O(\log n)$ interactions with KGC during the execution of the protocol to attain the temporary private keys (not the long-term private keys). So the extra communication and computation costs are required. While in our protocol all the participants extract their long-term private keys before the execution of the protocol, and subsequently there is no necessary of always-online KGC. The interaction with KGC also requires additional computation overheads. Hence our protocol does not impose a heavy computation and communication burden on KGC, otherwise it will become bottleneck of the system.

Table 1. Comparison of group key exchange protocols in efficiency

Protocol	Round number	Message number	Exp/SM	Pairing

ID-BT	$\lceil \log_2 n \rceil$	$3n\lceil \log_2 n \rceil$	$2n\lceil \log_2 n \rceil+4n-2$	$2n\lceil \log_2 n \rceil$
ID-TT	$\lceil \log_3 n \rceil$	$5(n-1)$	$9(n-1)$	$5n\lceil \log_3 n \rceil+3$
ID-BD	2	$2n$	$n(n+7)$	$4n$
TGDH[2]	$\lceil \log_2 n \rceil$	$2(n-1)$	$n(\lceil \log_2 n \rceil+1)$	-
A-GDH.2	n	n^2	$(n^2+3n)/2-1$	-
ID-SS	2	$2n$	$n(n+3)$	$2n$

As shown in Table 1, our protocol and ID-BD are two most efficient protocols in *communication* as compared with other protocols. In addition, it requires lower computation complexity than that of ID-BD and needs only 2 rounds. Also the fewest paring computations are needed in our protocol. It should be noted that TGDH and A-GDH.2 are not based on the identity system, so these two protocols do not require pairing-computations. TGDH has the lowest computation complexity than others, but unfortunately it requires $\log_2 n$ rounds. The following figure shows the total cost (include computation and communication costs) of *one group member* in diffident protocols.

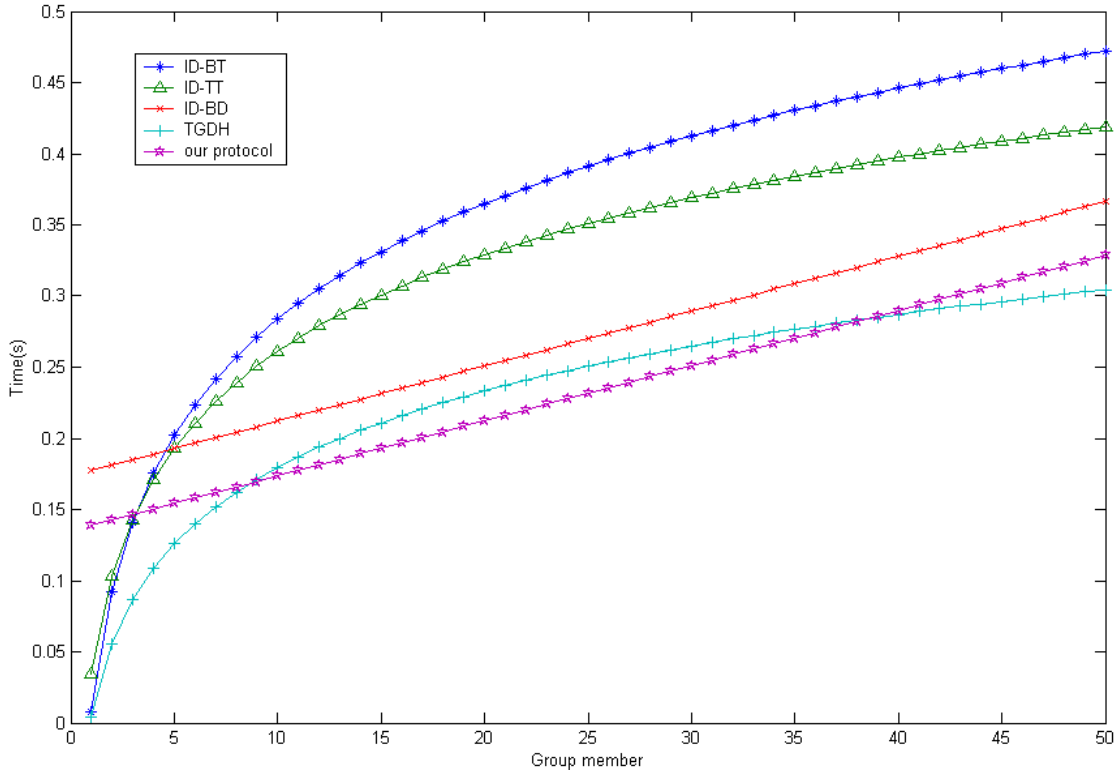

Fig. 1 Comparison of group key exchange protocols in efficiency

From the fig.1, we can see that when the number of group member is less than 38, our protocol has the most efficient performance, and it is a little worse than TGDH with the increase of group members. Thus comes to the conclusion, in a dynamic peer group (for example, wireless mesh networks), in which the group member is relatively small, our protocol is the most suitable one. However, for a lager group, we should choose a layered-protocol, like TGDH and ID-BT, etc.

7 Conclusion

In this paper, we formally present a constant-round ID-based protocol with forward secrecy for group key exchange, which is provably secure in the security model under the intractability of CDH problem. Our protocol focuses on round efficiency and needs only two communication rounds to

[2] Since there exists authenticated channels assumed in TGDH, we have to give the unauthenticated version of the protocol.

compute a common group key. So it is much more efficient than others. In particular, the protocol provides a batch verification technique, which can greatly improve computational efficiency. The protocol is a contributory key exchange, hence it does not impose a heavy computational burden on a particular party. Moreover, our protocol needs not always-online KGC, which saves a large amount of computation and communication costs. It should be noted that our protocol provides a methodology to the design of group key exchange protocol and most secret sharing scheme could be adopted to construct our scheme. However, the new protocol is for static groups and the approaches which manage the new users joining (or old users leaving) the group should be further studied.

References:

[1] A. Shamir. Identity-Based Cryptosystems and Signature Schemes. *in Advances in Cryptology-Crypto'84,* 1985. Berlin: Springer-Verlag.

[2] K. Becker and U. Wille. Communication Complexity of Group Key Distribution. *in 5th Conference on Computer and Communications Security*, 1998: ACM Press.

[3] E. Bresson, O. Chevassut and D. Pointcheval, Provably Authenticated Group DH Key Exchange-The Dynamic case. *In Proceedings of Asiacrypt '02*, **LNCS2248**, 2001, pp. 290-309.

[4] E. Bresson, O. Chevassut and D. Pointcheval, Dynamic group Diffie-Hellman key exchange under standard assumptions. *Advances in Cryptology - Eurocrypt'02, Proceedings*, Vol. **2332**, 2002, pp. 321-336.

[5] E. Bresson, O. Chevassut, D. Pointcheval, et al., Provably Authenticated Group DH Key Exchange. *In Proceedings of ACM CCS '01*, Vol. No., 2001, pp. 255-264.

[6] 6. M. Steiner, G. Tsudik and M. Waidner, Key agreement in dynamic peer groups. *Ieee Transactions on Parallel and Distributed Systems*, Vol. **11** No. 8, 2000, pp. 769-780.

[7] I. Ingemarsson, D.T. Tang and C.K. Wong, A Conference Key Distribution-System. *Ieee Transactions on Information Theory*, Vol. **28** No. 5, 1982, pp. 714-720.

[8] M. Burmester and Y. Desmedt. A Secure and Efficient Conference Key Distribution System(LNCS 950). *in Advances in Cryptography - Eurocrypt '94*. 1995: Berlin: Springer-Verlag.

[9] A. Joux, A one round protocol for tripartite Diffie-Hellman. *Algorithmic Number Theory*, Vol. **1838**, 2000, pp. 385-393.

[10] D. Nalla and K.C. Reddy, Identity Based Authenticated Group Key Exchange Protocol.Springer-Verlager. *In Proceedings of Indocrypt '02*, 2002, pp. 215-233.

[11] F. Zhang, S. Liu and K. Kim, ID-Based One Round Authenticated Tripartite Key Exchange Protocol with Pairings. *Cryptology ePring Archive*, Report 2002/122, 2002.

[12] R. Barua, R. Dutta and P. Sarkar, Extending Joux's protocol to multil party key agreement - (Extended abstract). *Progress in Cryptology -Indocrypt'03*, Vol. **2904**, 2003, pp. 205-217.

[13] K.Y. Choi, J.Y. Hwang and D.H. Lee, Efficient ID-based group key agreement with bilinear maps. *Public Key Cryptography - Pkc'04, Proceedings*, Vol. **2947**, 2004, pp. 130-144.

[14] G. Ateniese, M. Steiner and G. Tsudik, New multiparty authentication services and key agreement protocols. *Ieee Journal on Selected Areas in Communications,* Vol. **18** No. 4, 2000, pp. 628-639.

[15] Y. Kim, A. Perrig and G. Tsudik, Simple and Fault-Tolerant Key Exchange for Dynamic Collaborative Groups. *In 7th ACM Conference on Computer and Communications Security*, 2000, pp. 235-244.

[16] M. Steiner, G. Tsudik and M. Waidner, Diffie-Hellman Key Distribution Extended to Groups. *In ACM CCS '96*, 1996.

[17] Y.F. Chung, F. Lai and T.S. Chen, Group Key Management Method for the Distributed System. *WSEAS TRANSACTIONS on COMMUNICATIONS*, Vol. 6 No. 4, 2007, pp. 559-564.

[18] L. Zhu, Y. Cao, L. Liao, et al., Secure Group Key Distribution Protocol Based on Huffman One-way Key Chain Tree. *WSEAS TRANSACTIONS on*

COMPUTERS, Vol. 5 No. 6, 2007, pp. 1208-1213.

[19] W. Diffie and M.E. Hellman, New Directions in Cryptography. IEEE Transactions on Information Theory, Vol. **22** No. 6, 1976, pp. 644-654.

[20] K.C. Reddy and D. Nalla, Identity based authenticated group key agreement protocol. *Progress in Cryptology - Indocrypt'02, Proceedings,* Vol. **2551**, 2002, pp. 215-233.

[21] F.G. Zhang and X.F. Chen, Attack on an ID-based authenticated group key agreement scheme from PKC 2004. *Information Processing Letters*, Vol. **91** No. 4, 2004, pp. 191-193.

[22] W.G. Tzeng, A secure fault-tolerant conference-key agreement protocol. *Ieee Transactions on Computers*, Vol. **51** No. 4, 2002, pp. 373-379.

[23] J. Pieprzyk and C.H. Li, Multiparty key agreement protocols. *Iee Proceedings-Computers and Digital Techniques*, Vol. **147** No. 4, 2000, pp. 229-236.

[24] E. Bresson and D. Catalano, Constant round authenticated group key agreement via distributed computation. *In Proceedings of Public Key Cryptography - Pkc'04*, Vol. **2947**, 2004, pp. 115-129.

[25] O. Goladreich, S. Goldwasser and S. Micali, On the Cryptographic Applications of Random Functions, *in Proceeding of Crypto '84*. 1985, Sringer-Verlag. pp. 276-288.

[26] O. Goladreich, S. Goldwasser and S. Micali, How to Construct Random Functions. *Journal of the ACM*, Vol. **33** No. 4, 1986, pp. 792-807.

[27] M. Bellare and P. Rogaway, Random Oracles are practical: A Paradign for Designing Efficient Protocols. I*n the First ACM Conference on Computer and Communications Security*, 1993.

[28] M. Bellare and P. Rogaway, Entity Authentication and Key Distribution. Advances in Cryptography, -CRYPTO'93, **LNCS 773**, 1994, pp. 232-249.

[29] M. Bellare, R. Canetti and H. Krawczyk. A Modular Approach to the Design and Analysis of Authentication and Key-Exchange Protocols. *in Proc. of the 30th Annual Symp. on the Theory of Computing*, 1998, New York: ACM Press.

[30] D. Boneh, C. Gentry, B. Lynn, et al., Aggregate and verifiably encrypted signatures from bilinear maps. *Advances in Cryptology-Eurocrypt'03*, Vol. **2656**, 2003, pp. 416-432.

[31] F. Hess. Efficient Identity Based Signature Scheme Based on Pairings. in Selected Areas in Cryptography the 9th Annual Int'l Workshop, SAC'02. (LNCS 2595) 2003. New York: Springer-Verlag.

[32] R. Dutta, R. Barua and P. Sarkar, Pairing-Based Cryptography : A Survey, *Cryptology ePrint Archive*, Report 2004/064, 2004.

[33] M. Bellare and P. Rogaway, Provably secure session key distribution: the three party case. *In Proceedings of the 27th ACM Symposium on the Theory of Computing*, 1995, pp. 57-66.

[34] N.P. Smart, Identity-based authenticated key agreement protocol based on Weil pairing. *Electronics Letters*, Vol. **38** No. 13, 2002, pp. 630-632.

[35] K.A. Shim, Further analysis of ID-based authenticated group key agreement protocol from bilinear maps. *Ieice Transactions on Fundamentals of Electronics Communications and Computer Sciences*, Vol. **E90a** No. 1, 2007, pp. 295-298.