Tsinghua Science and Technology

Volume 19 | Issue 5 Article 1

2014

An Improved ID-Based Group Key Agreement Protocol

Kangwen Hu
School of Software, Beijing Institute of Technology, Beijing 100081, China.

Jingfeng Xue
School of Software, Beijing Institute of Technology, Beijing 100081, China.

Changzhen Hu
School of Software, Beijing Institute of Technology, Beijing 100081, China.

Rui Ma
School of Software, Beijing Institute of Technology, Beijing 100081, China.

Zhiqiang Li
School of Software, Beijing Institute of Technology, Beijing 100081, China.

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 01/13 pp421-428
Volume 19, Number 5, October 2014

An Improved ID-Based Group Key Agreement Protocol

Kangwen Hu, Jingfeng Xue, Changzhen Hu, Rui Ma, and Zhiqiang Li*

Abstract: ID-based constant-round group key agreement protocols are efficient in both computation and communication, but previous protocols did not provide valid message authentication. An improvement based on attack analysis is proposed in this paper. The improved method takes full advantage of the data transmitted at various stages of the protocol. By guaranteeing the freshness of authentication messages, the authenticity of the generator of authentication messages, and the completeness of the authenticator, the improved protocol can resist various passive and active attacks. The forward secrecy of the improved protocol is proved under a Katz-Yung (KY) model. Compared with existing methods, the improved protocol is more effective and applicable.

Key words: group key agreement protocol; ID; forward secrecy; nonsuper-singular elliptic curve

1 Introduction

The Identity-Based Cryptosystem (IBC) proposed by Shamir in 1984 is simpler than the PKI/CA that is currently widely used in key management[1]. In 2000, Joux proposed a tripartite key agreement with one round of communication using both Weil and Tate pairing[2]. Now, group key agreement protocols based on bilinear pairings of identity have elicited a lot of research. In 2002, Reddy first proposed an IDentity-based Authenti-cated Group Key Agreement (ID-AGKA) protocol with HOFT[3]. This protocol used Weil pairing, and provided implicit key authentication attribute; however, it just analyzed security attributes did not give rigorous proof. In 2003, Du et al. proposed a constant-round group key agreement protocol based on Burmester-Desmedt (BD) structure[4,5]. In 2004, Choi et al.[6] proposed a similar ID-AGKA and proved its security; it is efficient on both computation and communication but, because of a lack of entity authentication, it cannot resist internal impersonation or external attacks[7-9]. Many researchers have put forward a series of ways to improve Choi's ID-AGKA[8-13]. In this paper, a series of ID-AGKAs, represented by Choi's protocol, are analyzed and a new, improved scheme is proposed. This scheme can resist known internal impersonation attacks as well as external attacks.

2 Choi's ID-AGKA Protocol

In this section, we will briefly introduce the Choi's protocol; its detailed description is in Ref. [6].

Through the paper, we assume that G_1 is a cyclic additive group of big prime order q and G_2 is a cyclic multiplicative group of the same order q. P is G_1's generator. The discrete logarithm problem is intractable in both G_1 and G_2. $e\colon G_1 \times G_1 \to G_2$ is a valid bilinear map and satisfies the decisional bilinear Diffie-Hellman problem (DBDH) assumption[14]. $H\colon \{0,1\}^* \to Z_q$ and $H_1 : \{0,1\}^* \to G_1$ are two hash functions. In the security proof, H and H_1 are treated as random oracles.

Setup: The private Key Generation Center (KGC) randomly chooses a number $s \in Z_q^*$ as its master secret key, chooses G_1's generator P, computes $P_{\text{pub}} = sP$, and publishes system parameters params $= \{e, G_1, G_2, q, P, P_{\text{pub}}, H, H_1\}$.

Extract: The user U_{ID} with the identity of the ID sends the ID to the KGC. The KGC computes the public key $Q_{\text{ID}} = H_1(\text{ID})$ and the private key $S_{\text{ID}} =$

• Kangwen Hu, Jingfeng Xue, Changzhen Hu, Rui Ma, and Zhiqiang Li are with the School of Software, Beijing Institute of Technology, Beijing 100081, China. E-mail: xuejf@bit.edu.cn; lizq@bit.edu.cn.
∗ To whom correspondence should be addressed.
Manuscript received: 2014-04-15; revised: 2014-07-07; accepted: 2014-07-15

$sQ_{\rm ID}$. Then the KGC sends the $S_{\rm ID}$ to $U_{\rm ID}$ by a secure channel.

A group of users $\{U_1, U_2, \cdots, U_n\}$ want to establish a session key. $U_1, U_2, \cdots, U_n$ form a ring. For $1 \leqslant i \leqslant n$, U_{i-1} and U_{i+1} are the left and right users of U_i, $U_0 = U_n$, and $U_{n+1} = U_1$. $\langle Q_i, S_i \rangle$ are the public and private key pair of U_i.

Round 1: The U_i randomly chooses $a_i \in Z_q^*$ as its secret key and computes $P_i = a_i P$, $h_i = H(P_i)$, and $T_i = a_i P_{\rm pub} + h_i S_i$. Then the U_i keeps a_i secret and broadcasts $\langle P_i, T_i \rangle$.

Round 2: After U_i receives $\langle P_{i-1}, T_{i-1} \rangle$, $\langle P_{i+1}, T_{i+1} \rangle$, and $\langle P_{i+2}, T_{i+2} \rangle$, it will check whether they meet the following equation:

$$e\left(\sum_{k\in\{-1,1,2\}} T_{i+k}, P\right) = e\left(\sum_{k\in\{-1,1,2\}} (P_{i+k} + h_{i+k} Q_{i+k}), P_{\rm pub}\right).$$

If the above equation is not true, U_i will stop running the protocol and broadcast "failure". Otherwise, U_i will compute $D_i = e(a_i(P_{i+2} - P_{i-1}), P_{i+1})$ and broadcast D_i to the other members of the group.

Key computation: U_i computes its session key by the formula:

$$K_i = e(a_i P_{i-1} P_{i+1})^n D_i^{n-1} D_{i+1}^{n-2} \cdots D_{i-2} = e(P, P)^{a_1 a_2 a_3 + \cdots + a_{n-1} a_n a_1 + a_n a_1 a_2}.$$

3 Attacks on the Protocol

There are mainly four types of attacks on the Choi's protocol[6] and its variants[4,15].

(1) Replay impersonation attacks by malicious neighbors. Zhang and Chen[7] pointed out that U_{i-1} and U_{i-2}, two malicious neighbors of U_i, may collude to replay $\langle P_i, T_i \rangle$, which is U_i's authentication message in group G_A. The attackers can impersonate U_i in a new group, G_B, without being noticed.

(2) Impersonation attacks by colluding verifiers. Shim[8] pointed out that if U_{i-2}, U_{i-1}, and U_{i+1} (U_i's three malicious neighbors) collude, they can impersonate U_i without replaying $\langle P_i, T_i \rangle$ in a new group with randomly chosen a_i and D_i.

(3) External attacks that lead to group members computing different session keys. Li and He[9-11] pointed out that adversaries may divide the group into two subgroups and transmit different data to each. Thus, these two subgroups will get different session keys which means the protocol is broken.

(4) Passive attacks that compute session keys simply by monitoring transcripts. Li[10] pointed out that the protocol proposed by Liu and Xu[15] cannot resist the three attacks mentioned above and, according to the nature of bilinear pairings, as long as communication in the group is monitored, the attackers can compute session keys.

In fact, we find that as long as the adversary can pass verification in Round 1, it can break the protocol successfully. So these series of protocols[4,6,15] will not be able to resist man-in-the-middle attacks. If adversary $\mathcal{A}$ can control all transmitted/received data of U_1, and it can save data transmitted by U_1 in a normal agreement at sometime denoted by $\langle P_1', T_1', D_1' \rangle$. When U_1 is involved in a new agreement, adversary $\mathcal{A}$ can launch attacks as follows: U_1 will transmit $\langle P_1, T_1 \rangle$ according to the protocol in Round 1, $\mathcal{A}$ intercepts $\langle P_1, T_1 \rangle$ of U_1, broadcasts $\langle P_1', T_1' \rangle$ to the other members of the group and forwards $\langle P_i, T_i \rangle$ that transmitted by the other nodes to U_1. In Round 2, $\mathcal{A}$ replays D_i to the other members of the group and forwards D_i from the other nodes to U_1. After the agreement, the session key shared by $\mathcal{A}$ and U_1 is $K_1 = e(P, P)^{a_1 a_2 a_3 + \cdots + a_{n-1} a_n a_1 + a_n a_1 a_2}$; the session key shared by $\mathcal{A}$ and the other users is $K_1' = e(P, P)^{a_1' a_2 a_3 + \cdots + a_{n-1} a_n a_1 + a_n a_1 a_2}$. Thus, $\mathcal{A}$ can decrypt data that encrypted by any session keys of the two subgroups.

In summary, there are three reasons why impersonation attacks and external attacks work:

(1) The protocol does not authenticate the D_i in Round 2.

(2) The authenticators of $\langle P_i, T_i \rangle$ are U_i's neighbors only.

(3) The authentication data associates with protocol's current execution status little.

The reason for the success of passive attacks is that the temporary secret key is not used correctly in the session key formula.

Therefore, protocol improvement should be focused on the freshness of the authentication messages, the authenticity of the generator of the authentication messages, and the completeness of the authenticators.

(1) P_i broadcasted by each user in Round 1 can

identify current run of protocol, we can integrate all P_i values into the signature computation to ensure its freshness.

(2) The signature must be generated by a long-term private key to ensure the authenticity of the entities being authenticated.

(3) All users in the group must contribute to authentication information to ensure the completeness of the authenticators.

In addition, we should minimize unnecessary computations and messages for efficiency.

4 Improvement of the Protocol and Analysis

4.1 Improvement of the protocol

Setup: System parameters and the protocol initialization are the same as the original protocol.

Round 1: Each user U_i randomly chooses $a_i \in Z_q^*$, computes $P_i = a_i P$, broadcasts P_i to others and keeps a_i secret.

Round 2: Upon the receipt of all data broadcasted by other members of the group, user U_i computes $D_i = e(a_i(P_{i+1} - P_{i-1}), P_{\text{pub}})$, $T_i = a_i P_{\text{pub}} + h_i S_i$; where $h_i = H(D_i \| P_{\text{ID}} \| S_{\text{ID}})$, $P_{\text{ID}} = P_1 \| \cdots \| P_n$, and $S_{\text{ID}} = \text{ID}_1 \| \cdots \| \text{ID}_n$. Then, it broadcasts $\langle D_i, T_i \rangle$ to other members of the group.

Key computation: The user U_i checks whether $e(T_j, P) = e(P_j + h_j Q_j, P_{\text{pub}})$, $(1 \leqslant j \leqslant n$ and $j \neq i)$ is true. If not, U_i aborts and broadcasts "failure". Otherwise, U_i computes session key K_i:

$$K_i = e(a_i P_{i-1}, P_{\text{pub}})^n D_i^{n-1} D_{i+1}^{n-2} \cdots D_{i-2} = e(P, P)^{(a_1a_2 + a_2a_3 + \cdots + a_na_1)s}.$$

The authentication mechanism Γ of the protocol is similar to Hess's signature scheme. The definition is as follows.

Signature generation: Given a secret key, compute $S_i = sH_1(\text{ID}_i)$, then $T = aP_{\text{pub}} + hS_i$; where $a \in_R Z_q^*$, $h = H(D_i \| P_{\text{ID}} \| S_{\text{ID}})$, and $\langle aP, T \rangle \leftarrow \Gamma_{\text{gen}}(S_{\text{ID}})$.

Signature verification: Given public key Q_{ID} and signature $\langle D_i, T_i \rangle$, check whether $e(T, P) = e(aP + hQ_{\text{ID}}, P_{\text{pub}})$ is true, where $h = H(D_i \| P_{\text{ID}} \| S_{\text{ID}})$; True or False $\leftarrow \Gamma_{\text{ver}}(Q_{\text{ID}}, \langle aP, T \rangle)$.

The batch validation reduces the number of calculations of bilinear pairings and improves verification efficiency[13]. Our improvement is similar to those in Refs. [4, 6, 16]. According to the nature and symmetry of bilinear pairings, $\prod_{i=1}^{n} D_i = 1$ is true. After each user receives all D_i, they should check D_i by this formula.

4.2 Proof of security

This protocol transmits messages through broadcasting. All participants (including the adversary who controls the network) will receive the same message. The KY model satisfies this feature[17]. The security definitions can be found in Refs. [6, 17]. To prove this protocol, we measure indistinguishability by the hybrid argument method[18]. We name this improved protocol IB-AGKA and will prove that it still provides forward secrecy.

Theorem 1 After the active rival issues q_{ex} *Execute* inquiries and q_s *Send* inquiries within time t, we define $\text{Adv}_{\text{IB-AGKA}}^{\text{ID-AGKA-fs}}(t, q_{\text{ex}}, q_s)$ as the maximum advantage of the attacker. We define Forger_Γ as a Probabilistic Polynomial Time (PPT) forger of authentication scheme Γ under the adaptively chosen ID attack, and $\text{Forger}_\Gamma^{\text{ID}}$ as a PPT forger of Γ under given ID attack. We take hash functions H and H_1 as random oracles.

$$\text{Adv}_{\text{IB-AGKA}}^{\text{ID-AGKA-fs}}(t, q_{\text{ex}}, q_s) \leqslant 2n(q_{\text{ex}} + q_s)\text{Adv}_{G_1,G_2,e}^{\text{DBDH}}(t) + \text{Adv}_\Gamma^{\text{Forge}}(t).$$

Here, $\text{Adv}_\Gamma^{\text{Forge}}(t)$ is the maximum advantage of any forger Forger_Γ running in time t.

Proof Let $\mathcal{A}$ be an active attacker, who can get advantage in attacking the protocol in two ways:

(1) Forging an authentication message or impersonating a user.

(2) Breaking the protocol without modifying any message.

Assuming that Adversary $\mathcal{A}$ breaks IB-AGKA by adaptive impersonation attack, we can construct a forger Forger_Γ C of an authentication scheme Γ by $\mathcal{A}$. This forger can produce a valid ternary $\langle U_i, D_i, T_i \rangle$ of authentication scheme Γ in the following ways.

Forger_Γ C honestly generates the public/private key pair of all other users except U_i. C simulates the oracle inquiries of adversary $\mathcal{A}$ in the natural way; this results in a perfect simulation unless $\mathcal{A}$ issues Corrupt(U_i), in which case, C aborts. If $\mathcal{A}$ produces a new valid $\langle U_i, D_i, T_i \rangle$, we denote this event by *Forge* and make $\mathcal{A}$ pass the result to C. Thus, we believe that C is a successful forger of authentication scheme Γ. So the probability of C's success meets $\Pr_A[\text{Forge}] \leqslant \text{Adv}_{C,\Gamma}^{\text{Forge}}(t) \leqslant \text{Adv}_{\Gamma}^{\text{Forge}}(t)$.

Next, we assume that $\mathcal{A}$ can also break IB-AGKA without altering transcripts; thus, we can use IB-AGKA to solve the MDBDH problem. According to Ref. [6], MDBDH and DBDH are computationally equivalent, namely, $\text{Adv}_{G_1,G_2,e}^{\text{MDBDH}}(t) = \text{Adv}_{G_1,G_2,e}^{\text{DBDH}}(t)$. We first consider the case that $\mathcal{A}$ issues only a single *Execute* query *Execute*$(\text{ID}_1, \cdots, \text{ID}_n)$ and then extend this to multiple *Execute* queries. Let n be the number of users selected by $\mathcal{A}$; the distribution of transcripts T and the group session key K are give by:

$$\text{Params} = \left[\begin{array}{c} (G_1, G_2, e) \leftarrow \text{IG}_{\text{BDH}}(1^k);\ P \leftarrow G_1; \\ s \leftarrow Z_q^*;\ P_{\text{pub}} = sP \\ Q_1, \cdots, Q_n \leftarrow G_1;\ S_1 = sQ_1, \cdots, \\ s_n = sQ_n : (G_1, G_2, e, P, P_{\text{pub}}) \end{array}\right],$$

$$\text{Real} \stackrel{\text{def}}{=} \left[\begin{array}{c} a_1, \cdots, a_n, h_1, \cdots, h_n \leftarrow Z_q^*; \\ P_1 = a_1 P, \cdots, P_n = a_n P; \\ T_1 = a_1 P_{\text{pub}} + h_1 S_1, \cdots, \\ T_n = a_n P_{\text{pub}} + h_n S_n; \\ D_1 = \dfrac{e(a_1 P_2, P_{\text{pub}})}{e(a_1 P_n, P_{\text{pub}})}, D_2 = \dfrac{e(a_2 P_3, P_{\text{pub}})}{e(a_2 P_1, P_{\text{pub}})}, \\ \cdots, D_n = \dfrac{e(a_n P_1, P_{\text{pub}})}{e(a_n P_{n-1}, P_{\text{pub}})} \\ T = \langle P_1, \cdots, P_n, T_1, \cdots, T_n, D_1, \cdots, D_n \rangle \\ K = e(a_1 P_n, P_{\text{pub}})^n D_1^{n-1} \cdots D_{n-1} : (T, K) \end{array}\right].$$

where IG_{BDH} is a PPT algorithm that takes a security parameter 1^k, runs in polynomial time, and outputs two groups G_1 and G_2 of the same order q and an reasonable bilinear map e: $G_1 \times G_2 \rightarrow G2$.

Now we construct a serial of mixed distributions, $\text{Fake}_i\,(i = 1, \cdots, n)$ which is defined as follows:

$$\text{Fake}_i \stackrel{\text{def}}{=} \left[\begin{array}{c} r_{s,n,1}, a_1, \cdots, a_n, h_1, \cdots, h_n \leftarrow Z_q^*; \\ P_1 = a_1 P, \cdots, P_n = a_n P; \\ T_1 = a_1 P_{\text{pub}} + h_1 S_1, \cdots, \\ T_n = a_n P_{\text{pub}} + h_n S_n; \\ D_1 = \dfrac{e(a_1 P_2, P_{\text{pub}})}{e(r_{s,n,1} P, P)}, D_2 = \dfrac{e(a_2 P_3, P_{\text{pub}})}{e(a_2 P_1, P_{\text{pub}})}, \\ \cdots, D_n = \dfrac{e(r_{s,n,1} P, P)}{e(a_n P_{n-1}, P_{\text{pub}})} \\ T = \langle P_1, \cdots, P_n, T_1, \cdots, T_n, D_1, \cdots, D_n \rangle; \\ K = e(r_{s,n,1} P, P_{\text{pub}})^n D_1^{n-1} \cdots D_{n-1} : (T, K) \end{array}\right].$$

According to this construction method, we can obtain the distribution:

$$\text{Fake}_n \stackrel{\text{def}}{=} \left[\begin{array}{c} r_{s,n,1}, \cdots, r_{s,n-1,n}, a_1, \cdots, a_n, h_1, \cdots, \\ h_n \leftarrow Z_q^*;\ P_1 = a_1 P, \cdots, P_n = a_n P; \\ T_1 = a_1 P_{\text{pub}} + h_1 S_1, \cdots, \\ T_n = a_n P_{\text{pub}} + h_n S_n; \\ D_1 = \dfrac{e(r_{s,1,2} P, P_{\text{pub}})}{e(r_{s,n,1} P, P_{\text{pub}})}, D_2 = \dfrac{e(r_{s,2,3} P, P_{\text{pub}})}{e(r_{s,1,2} P, P_{\text{pub}})}, \\ \cdots, D_n = \dfrac{e(r_{s,n,1} P, P_{\text{pub}})}{e(r_{s,n-1,n} P, P_{\text{pub}})} \\ T = \langle P_1, \cdots, P_n, T_1, \cdots, T_n, D_1, \cdots, D_n \rangle \\ K = e(r_{s,n,1} P, P_{\text{pub}})^n D_1^{n-1} \cdots D_{n-1} : (T, K) \end{array}\right].$$

Adversary $\mathcal{A}$ can obtain all long-term secret keys S_i and hash values h_1 $(i = 1, \cdots, n)$ through multiple *Corrupt* and H inquiries; compute $a_i P_{\text{pub}} = T_i - hS_i = sa_i P$ (since $P_{\text{pub}} = sP$ is a global parameter, and $P_i = a_i P$ can be obtained from transcripts). Let $\varepsilon(t) = \text{Adv}_{G_1,G_2,e}^{\text{MDBDH}}(t)$, according to the MDBDH assumption; any distinguishing algorithms A running in time t can result:

$$|\Pr[(T, K) \leftarrow \text{Real} : A(T, K) = 1] - \Pr[(T, K) \leftarrow \text{Fake}_1 : A(T, K) = 1]| \leqslant \varepsilon(t).$$

The reason is that adversary $\mathcal{A}$ has to distinguish $e(P, P)^{sa_1 a_n}$ from $e(P, P)^{r_{s,n,1}}$, which satisies MDBDH.

Similarity:

$$|\Pr[(T, K) \leftarrow \text{Fake}_1 : A(T, K) = 1] - \Pr[(T, K) \leftarrow \text{Fake}_2 : A(T, K) = 1]| \leqslant \varepsilon(t),$$

$$\vdots$$

$$|\Pr[(T, K) \leftarrow \text{Fake}_{n-1} : A(T, K) = 1] - \Pr[(T, K) \leftarrow \text{Fake}_n : A(T, K) = 1]| \leqslant \varepsilon(t).$$

Let $e(P, P) = g \in G_2$. In experiment Fake_n, the value $r_{s,1,2}, \cdots, r_{s,n,1}$ are constrained by T according to the following n equations:

$$\begin{cases} \log_g D_1 = r_{s,1,2} - r_{s,n,1}, \\ \log_g D_2 = r_{s,2,3} - r_{s,1,2}, \\ \quad\vdots \\ \log_g D_n = r_{s,n,1} - r_{s,n-1,n}. \end{cases}$$

The coefficient matrix of the equation set is

$$\begin{bmatrix} 1 & 0 & \cdots & \cdots & -1 \\ -1 & 0 & 0 & \cdots & 0 \\ & & \vdots & & \\ 0 & \cdots & \cdots & -1 & 1 \end{bmatrix}.$$

Its rank is $n - 1; n - 1$ vectors are linearly independent. In distribution Fake_n, $K_{\text{Fake}_n} =$

$e(P,P)^{r_{s,1,2}+r_{s,2,3}+\cdots+r_{s,n,1}}$. If both sides of the equation are logarithmic, we get $\log_g K_{\text{Fake}_n} = r_{s,1,2} + r_{s,2,3} + \cdots + r_{s,n,1}$. Obviously, it is independent of T. This indicates that for any adversary $\mathcal{A}$, the equation is true:

$$\Pr[(T,K) \leftarrow \text{Fake}_n : A(T,K) = 1] =$$
$$\Pr[T \leftarrow \text{Fake}_n, K \leftarrow \text{Random}: A(T,K) = 1].$$

This means that adversary $\mathcal{A}$ cannot distinguish session key K_{Fake_n} from a random number of the same length by distribution Fake_n.

Similarly, under MDBDH assumption, any algorithm A running in time t, we can get:

$$|\Pr[T \leftarrow \text{Fake}_n; K \leftarrow \text{Fake}_n : A(T,K) = 1]-$$
$$\Pr[T \leftarrow \text{Fake}_{n-1}; K \leftarrow \text{Random}: A(T,K) = 1]| \leqslant \varepsilon(t),$$
$$\vdots$$
$$|\Pr[T \leftarrow \text{Fake}_1; K \leftarrow \text{Random}: A(T,K) = 1]-$$
$$\Pr[T \leftarrow \text{Real}; K \leftarrow \text{Random}: A(T,K) = 1]| \leqslant \varepsilon(t).$$

According to the above equations, we can get:

$$|\Pr[T \leftarrow \text{Real}; K \leftarrow \text{Real}: A(T,K) = 1]-$$
$$\Pr[T \leftarrow \text{Real}; K \leftarrow \text{Random}: A(T,K)=1]| \leqslant 2n\varepsilon(t).$$

This means, when the adversary $\mathcal{A}$ intercepts all transcripts of the subgroup he chooses, the advantage of distinguishing real session key K from a random value of the same length is $2n\varepsilon(t)$.

Because $\varepsilon(t) = \text{Adv}^{\text{MDBDH}}_{G_1,G_2,e}(t) = \text{Adv}^{\text{DBDH}}_{G_1,G_2,e}(t)$, $\Pr_A[\sim \text{Forge}] \leqslant 2n\text{Adv}^{\text{DBDH}}_{G_1,G_2,e}(t)$ is established.

In summary:

$$\text{Adv}^{\text{ID-AGKA-fs}}_{\text{IB-AGKA}}(t,1) \leqslant 2n\text{Adv}^{\text{DBDH}}_{G_1,G_2,e}(t) + \text{Adv}^{\text{Forge}}_{\Gamma}(t).$$

After issuing q_S *Send* inquires and q_E *Execute* inquires, the result is:

$$\text{Adv}^{\text{ID-AGKA-fs}}_{\text{IB-AGKA}}(t, q_{\text{E}}, q_{\text{S}}) \leqslant$$
$$2n(q_{\text{S}} + q_{\text{E}})\text{Adv}^{\text{DBDH}}_{G_1,G_2,e}(t) + \text{Adv}^{\text{Forge}}_{\Gamma}(t).$$

Detailed calculations regarding the forger's advantage are in Refs. [6, 19]. ■

4.3 Analysis of improvement

This improvement uses some of the ideas in Refs. [11, 13, 20]. In addition to its forward security, it can resist all kinds of known attacks:

(1) Active attacks[7-10]: U_i's malicious neighbors U_{i-1} and U_{i+1} save the message $\langle P_i', D_i', T_i' \rangle$, which was transmitted in the old group G_A by U_i. They send P_i' in Round 1 of the new group G_B, thus $a_i = a_i'$. According to the protocol, U_{i-1} and U_{i+1} in G_B can get P_i broadcasted by all users in the first round. If the collusion attackers want to pass verification in Round 2, they must construct the correct signature of D_i, which is $T_i = a_i P_{\text{pub}} + h_i S_i$. Although the attacker can compute $h_i = H(D_i \| P_{\text{ID}} \| S_{\text{ID}})$, it still needs to compute $a_i P_{\text{pub}}$ and obtain U_i's long-term private key S_i. Under the Elliptic Curve Discrete Logarithm Problem (ECDLP) assumption and secure transmission of the private key, neither can be obtained, i.e., the adversary cannot calculate T_i. The malicious neighbors cannot generate valid U_i signatures in new group, and the protocol's honest participants are able to detect the attacks.

(2) Passive attacks[10]: The adversary can obtain system parameters P_i, T_i, and D_i. According to $K_i = e(a_i P_{i-1}, P_{\text{pub}})^n D_i^{n-1} D_{i+1}^{n-2} \cdots D_{i-2}$, the adversary knows that $P_i = a_i P$, $P_{\text{pub}} = sP$, and $P_{i-1} = a_{i-1} P$. Under the Computational Bilinear Diffie-Hellman Problem (CBDHP) problem assumption, the adversary cannot compute $e(a_i P_{i-1}, P_{\text{pub}}) = e(P,P)^{s a_{i-1} a_i}$, so they cannot get the session key.

In addition to the Internet, LAN, and other common networks, these improvements also apply to environments where the node's calculation capability is weak, the node's buffer is small, and the channel bandwidth is low—such as Wireless Sensor Networks (WSNs). In this type of network, protocol data can be transmitted in batches. In Round 1, U_i first unicasts P_i to U_{i-1} and U_{i+1} and then broadcasts P_i to the other nodes. After U_i receives P_{i-1} and P_{i+1}, it computes D_i and broadcasts it. Thus, although the P_i values of the other nodes required in Round 2 for T_i calculation are received relatively late, we can first calculate session key K and verify T_i later. The improvement can avoid channel congestion caused by all node's simultaneous broadcasting. Nodes need not wait for all data to arrive and then calculate. The parallelism of system is best.

Regardless of whether Tate or Weil pairing is used, when parameters P and Q are linearly dependent, their safety is not guaranteed. We denote, $Q = kP$ $(k \in_R Z_m^*,\ P \in E[m])$, thus, we have $e(P,Q) = e(P,kP) = e(P,P)^k = 1$ with Weil pairing, according to its identity element and bilinear nature[14]. Obviously, it must not occur in practice. In Ref. [6], Choi made an admissive bilinear map that satisfies $e(P,P) \neq 1$, that may need extra work on existing pairing. Although the hypersingular curve can map two related points to different groups[14,21-23] by distortion mapping, the nonhypersingular curve does not have this property. In the original protocol[6] and its improvement[9, 10, 13], the two bilinear parameters both belong to $\langle P \rangle$; if some user's temporary private key is multiple of some other's, then P and Q are linearly dependent. In our protocol, the second parameter of bilinear pairing is always P_{pub}. When we configure the global parameters, we can set it outside $\langle P \rangle$, such as $P_{\text{pub}} = sQ$, $Q \notin \langle P \rangle$. The session key is $K = e(P, Q)^{(a_1 a_2 + a_2 a_3 + \cdots + a_n a_1)s}$. Under the MDBDH assumption, the modification still holds forward secrecy. Thus, our protocol can be implemented by supersingular/nonsupersingular curves.

In addition, there are some other improvements to resist existing attacks. Shim[8] proposed to use the long-term private keys to sign T_i and D_i, which can resist the impersonation attacks mentioned in Refs. [7, 8]. However, Ref. [13] proposed that such long-term private keys cannot resist replay attacks. Park proposed that the user index can be randomized by KGC[12]; each user calculates D_i by new index[7,8]. This improvement can resist impersonation attacks to some extent, but relies too much on KGC, and massive encryption/decryption operations make KGC a bottleneck. Du et al.[25] proposed that all users maintain a counter together, increase it by 1 when agreement occurs, and participants use the product of the counter and the original long-term private key as the new private key to make every $\langle P_i, T_i \rangle$ different. Thus, users can detect replayed authentication message in Round 1, but system synchronization is heavy to maintain. Choi and Li proposed to integrate protocol messages of current run into the signature[10, 11, 13], and this is by far the most comprehensive method. However, Choi's improvement requires two signatures[13]; in fact, the signature of Round 2 also provides verification of messages in Round 1.

Table 1 provides a comprehensive comparison of several improvements. Our protocol does not need either extra KGC assistance or a global counter, and is compatible with non-supersingular curves. Compared with other improved methods, our protocol applies more widely.

Table 2 compares the amount of calculation operations required of individual participants, including point multiplication, point addition, hash calculation, and bilinear pairing calculation. The multiplication and exponentiation in the session key calculation are the same in each protocol and are not compared.

Because the data of Round 1 can be verified by KGC, the calculation payload of individual participant of Park's protocol is minimal. But, in Park's protocol, there is an extra public key encryption operation in Round 2. On the whole, our protocol is the most efficient.

5 Conclusions

Our proposed protocol takes full advantage of the data transmitted at various stages of the protocol and provides both entity authentication and freshness. We extend authenticators from three neighbors to all nodes in the group. Our protocol can resist various known internal impersonation attacks and external attacks. In our protocol, the global public key can be placed in the noncyclic group, thus the protocol can be implemented by supersingular/non-supersingular elliptic curve.

Asymmetric identity based group key agreement protocols and related security models have been proposed by some scholars[26, 27], which are different from previous methods of session key agreement. In this type of protocol, participants will negotiate a common group key, but get a different decryption key individually. It is a very significant research direction.

Acknowledgements

This work was supported by the Key Project of National Defense Basic Research Program of China (No. B1120132031).

Table 1 Comprehensive comparison of improved protocols.

Protocol	Rounds	Need KGC's assistance	Need a global counter	Compatible with non-supersingular curves	Number of signatures
Park and Choi[12]	3	Y	N	N	1
Du et al.[25]	2	N	Y	N	1
Choi[13]	2	N	N	N	2
Li and He[11]	2	N	N	N	1
This protocol	2	N	N	Y	1

Table 2 Comparison of the amount of calculation required by individual participants.

Protocol	Point multiplication	Point addition	Hash	Bilinear pairing
Park and Choi[12]	5	2	1	2
Du et al.[25]	$n+4$	$3n-1$	n	4
Choi[13]	$n+7$	$3n+7$	$n+3$	5
Li and He[11]	$n+4$	$n+1$	n	$2n-1$
This protocol	$n+4$	$3n$	n	4

References

[1] A. Shamir, Identity-based cryptosystems and signature schemes, *Lecture Notes in Computer Science*, vol. 196, no. 1, pp. 47-53, 1985.

[2] A. Joux, A one round protocol for tripartite Diffie-Hellman, *Lecture Notes in Computer Science*, vol. 1838, no. 1, pp. 385-394, 2000.

[3] K. Reddy and D. Nalla, Identity based authenticated group key agreement protocol, *Lecture Notes in Computer Science*, vol. 2551, no. 1, pp. 215-233, 2002.

[4] X. Du, Y. Wang, and J. Ge, An ID-based authenticated two round multi-party key agreement, Cryptology ePrint Archive, http://eprint.iacr.org/2003/247, 2010, Jul. 15.

[5] M. Burmester and Y. Desmedt, A secure and efficient conference key distribution system, *Lecture Notes in Computer Science*, vol. 950, no. 1, pp. 275-286, 1995.

[6] K. Y. Choi, J. Y. Jwang, and D. H. Lee, Efficient ID-based group key agreement with bilinear maps, *Public Key Cryptography*, vol. 2947, pp. 130-144, 2004.

[7] F. Zhang and X. Chen, Attack on an ID-based authenticated group key agreement scheme from PKC, *Information Processing Letters*, vol. 91, no. 1, pp. 191-193, 2004.

[8] 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. E90, no. A1, pp. 295-298, 2007.

[9] G. Li and D. He, Analysis and improvement of group key agreement protocol ID-AGKA, *Computer Engineering*, vol. 35, no. 6, pp. 148-149, 2009.

[10] G. Li, *The Analysis and Design of Group Key Agreement Protocol*. Chengdu, China: Southwest Jiaotong University Press, 2008.

[11] G. Li and D. He, ID-based authenticated group key agree-ment protocol, *Computer Science*, vol. 36, no. 1, pp. 60-64, 2009.

[12] H. Park and K. Y. Choi, Improving ID-based authenticated group key agreement scheme at PKC2004, presented at the Symposium on Cryptography and Information Security, 2008.

[13] K. Y. Choi, ID-based authenticated group key agreement secure against insider attacks, *IEICE Trans, Fundamentals*, vol. E91, no. A9, pp. 1828-1830, 2008.

[14] I. Blake, G. Seroussi, and N. Smart, *Advances in Elliptic Curve Cryptography*. Cambridge University Press, 2005.

[15] C. Liu and Q. Xu, ID-based group key agreement protocol, in *The Proceeding of ChinaCrypt'2006*, Science and Technology Press of China, 2006, pp. 181-187.

[16] R. Dutta and R. Barua, Provably secure constant round contributory group key agreement in dynamic setting, *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2007-2025, 2008.

[17] J. Katz and M. Yung, Scalable protocols for authenticated group key exchange, *Lecture Notes in Computer Science*, vol. 2729, pp. 110-125, 2003.

[18] O. Goldreich, *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press, 2001.

[19] F. Hess, Exponent group signature schemes and efficient identity based signatures schemes based on pairings, Cryptoglogy ePrint Archive Report 202/012, http://eprint.iacr.org/2002/012, 2010, Jul. 10.

[20] L. Harn, M. Mehta, and W. J. Hsin, Integrating Diffie-Hellman key exchange into the digital signature algorithm (DSA), *IEEE Communications Letters*, vol. 8, no. 3, pp. 198-200, 2004.

[21] M. Scoot, Computing the tate pairing, *Lecture Notes in Computer Science*, vol. 3376, pp. 293-304, 2005.

[22] P. S. Barreto, H. Y. Kim, and B. Lynn, Efficient algorithms for pairing-based cryptosystems, *Lecture Notes in Computer Science*, vol. 2442, pp. 354-368, 2002.

[23] M. Scott, Scaling security in pairing-based protocols, Cryptology ePrint Archive Report 2005/139, http://eprint.iacr.org/2005/139, 2010, Jun. 12.

[24] C. Zhao and F. Zhang, Research and development on efficient pairing computations, *Journal of Software*, vol. 20, no. 11, pp. 3001-3009, 2009.

[25] X. Du, Y. Wang, and J. Ge, An improved ID-based authenticated group key agreement scheme, Cryptology ePrint Archive, http://eprint.iacr.org/2003/260, 2010, Jul. 15.

[26] Q. Wu, Y. Mu, and W. Susilo, Asymmetric group key agreement, *Lecture Notes in Computer Science*, vol. 5479, pp. 153-170, 2009.

[27] L. Zhang, Q. Wu, and B. Qin, Identity-based authenticated asymmetric group key agreement protocol, *Lecture Notes in Computer Science*, vol. 6196, no. 1, pp. 510-519, 2010.

Kangwen Hu is a master student in the School of Computer at the Beijing Institute of Technology. He received his BEng degree from Beijing Institute of Technology in 2006. His current research interest is software security.

Jingfeng Xue received the PhD degree from Beijing Institute of Technology in 2003. He is currently the professor with the School of Software, Beijing Institute of Technology. His current research interests is software security.

Changzhen Hu received the PhD degree from Beijing Institute of Technology in 1996. He is currently the professor with the School of Software, Beijing Institute of Technology. His current research interest is information security.

Rui Ma received the PhD degree from Beijing Institute of Technology in 2002. She is an associate professor with the School of Software, Beijing Institute of Technology. Her current research interests include software security and Internet of Things.

Zhiqiang Li received the MEng degree from Beijing Institute of Technology in 2003. He is a lecturer with the School of Software, Beijing Institute of Technology. His current research interests include software security and big data.