Group-oriented encryption secure against collude attack

Chunbo Ma[1,2], Jun Ao[1], and Jianhua Li[3]

[1]Information and Communication College,
Guilin University of Electronic Technology, Guilin, Guangxi, 541004, P. R. China
Cb_ma@263.net
[2] The State Key Laboratory of Information Security,
Beijing, 100049, P. R. China
[3] School of Information Security Engineering
Shanghai Jiao Tong University, Shanghai, 200030, P. R. China

Abstract. A group oriented encryption scheme is presented in this paper. In this scheme, a sender is allowed to encrypt a message using the group public key and send the ciphertext to the group. Any user in the group can independently decrypt the ciphertext via his private key. The scheme is secure against adaptively chosen ciphertext attack and collude attack.
Keywords. Group, Encrypt, Adaptively chosen ciphertext attack, Collude attack

1 Introduction

Group oriented encryption scheme [1] is such a scheme that a sender is allowed to send a ciphertext to a designated group and nobody besides the users of the group can decrypt the ciphertext using his private key. In this model of one-to-group, all the users share a common public key, i.e. the group public key and each user has his private key.

As a useful method of multi-parity communication, the group communication is playing an important role in some network-based applications. Since many users are involved and lots of resource is occupied, it is feasible for an attacker to take active or passive attacks through open networks. The research on group oriented encryption scheme is necessary and it is crucial to the security of network communications.

The notion of broadcast encryption is similarly to that of group oriented encryption. It was originally introduced by Fiat and Naor [2]. In such a scheme, the sender encrypts a message for some subset of receivers and sends the ciphertext by the broadcast over Internet. Any receiver in the designated subset can use his private key to decrypt the ciphertext. The different between these two schemes is that the public key of each user is common in group oriented encryption scheme, i.e. the group public key. There are some researches on broadcast encryption schemes [4][9].

Group oriented encryption scheme can be used in many aspects, such as TV subscription services, communications in LAN or VPN and so on. As a secure encryption scheme, it is necessary to withstand adaptively chosen ciphertext attack. Moreover, the security of key generation is worth paying attention to. In the scheme mentioned in [1], the key generation algorithm is vulnerable to collude attack. For example, two valid users can share their private keys and compute a new private key for any other user.

In this paper, we introduce the identity of the user to improve the scheme mentioned in [1]. The new scheme is secure against chosen ciphertext attack and the key generation withstands collude attack from the users of the group.

The rest of paper consists of following sections. In section 2, we introduce some related works on broadcast encryptions and group oriented encryption scheme. In section 3, we give the security model and some complexity assumptions. The proposed group oriented encryption scheme is presented in section 4. In section 5, we discuss the security of the proposed scheme in random oracle model. Finally, we draw the conclusions in section 6.

2 Related works

Fiat and Naor [2] first presented the concept of broadcast encryption and proposed a solution that is secure against a collusion of m users and the length of the ciphertext is $O(m\log^2 m\log n)$, where n is the number of users and m is the number of colluders who are revoked by the system. Further research, such as [3], supposed a method which ciphertext and keys don't rely on the m. The scheme has private key size of $O(\log^2 n)$. Halevy and Shamir [4] improved on the

broadcast encryption and presented their own solution. Other study about this scheme can be found in [5]. There are some other studies about broadcast encryption, such as [6][7][8].

Boneh et al. [9] presented two broadcast encryption systems for dynamic receivers. In their first scheme, both ciphertexts and private keys are of constant size, but the public key size in the system is linear in the total number of receivers. The second scheme is a generalization of the first. In addition, it is a collusion resistant broadcast scheme with $O(\sqrt{n})$ ciphertext and public key size, i.e. as the increasing of users, both ciphertext and public key size will increase also.

Ma et al. [1] designed a group-oriented encryption scheme. In such a scheme, anyone can encrypt a message using the group public key and distribute the ciphertext to the designated group. Any member in the group can independently decrypt the ciphertext via his private key. However, two valid users in this scheme can cooperate with each other to obtain a new and valid private key that can be used by any user. In other words, the scheme is vulnerable to collude attack.

3 Background

3.1. Bilinear Maps

Let G_1 be a cyclic multiplicative group generated by g, whose order is a prime q and G_2 be a cyclic multiplicative group of the same order q. Assume that the discrete logarithm in both G_1 and G_2 is intractable. A bilinear pairing is a map $e: G_1 \times G_1 \to G_2$ and satisfies the following properties:

1. *Bilinear:* $e(g^a, p^b) = e(g, p)^{ab}$. For all g, $p \in G_1$ and $a, b \in Z_q$, the equation holds.
2. *Non-degenerate:* There exists $p \in G_1$, if $e(g, p) = 1$, then $g = \mathrm{O}$.
3. *Computable:* For g, $p \in G_1$, there is an efficient algorithm to compute $e(g, p)$.
4. *commutativity:* $e(g^a, p^b) = e(g^b, p^a)$. For all g, $p \in G_1$ and $a, b \in Z_q$, the equation holds.

Typically, the map e will be derived from either the Weil or Tate pairing on an elliptic curve over a finite field. Pairings and other parameters should be selected for efficiency and security.

3.2. General Scheme

A group oriented encryption is made up of five algorithms.

1. **Initialize.** Given the security parameter λ, the algorithm outputs the system parameters.
2. **KeyGen (A, p_i).** Inputs the designated group A and user $p_i \in A$. It outputs the group A's public key PK_A and user p_i 's private key (d_{i1}, d_{i2}, d_{i3}).
3. **KeyVer(** (d_{i1}, d_{i2}, d_{i3}) **,** p_i **).** Inputs a private key (d_{i1}, d_{i2}, d_{i3}) and a user p_i. If the key belongs to the user, the algorithm outputs TRUE, otherwise gives ERROR message.
4. **Encrypt (M, PK).** Let M be a message to be encrypted to the group A. Inputs a message M and a public key PK_A, the algorithm outputs (c_1, c_2, c_3, c_4, c_5). We refer to $C = (c_1, c_2, c_3, c_4, c_5)$ as the ciphertext of message M.
5. **Decrypt (C,** (d_{i1}, d_{i2}, d_{i3}) **).** Inputs the cighertext C and the private key (d_{i1}, d_{i2}, d_{i3}) of user p_i. If $p_i \in A$, then the algorithm can decrypt the ciphertext via the private key (d_{i1}, d_{i2}, d_{i3}) and output the plaintext M.

3.3. Security Notions

The indistinguishable chosen ciphertext attack (IND-CCA) [12] presented by Goldwasser and Micali has been widely used to analyze the security of an encryption scheme. In this model, several queries are available to the attacker to model his capability. Subsequently, Rackhoff and Simon [13] enhanced it and proposed adaptively chosen ciphertext attack (IND-CCA2). In this section, we define adaptively chosen ciphertext security of the group oriented encryption scheme. Security is defined using the following game between an *Attacker* and *Challenger*.

1. **Setup.** The *Challenger* initializes the system. The *Challenger* gives the *Attacker* the resulting system parameters and the public key PK. It keeps the SK to itself.
2. **Query phase 1.** The *Attacker* adaptively issues decryption queries $q_1 . q_2, \cdots, q_m$. The *Challenger* responds with **Decrypt (C, SK)**.

3. **Challenge.** Once the *Attacker* decides that **Query phase 1** is over it outputs two equal length messages (M_0, M_1) to the *Challenger*. The *Challenger* picks a random bit $e \in \{0,1\}$, and encrypts the message M_e. It gives ciphertext C^* as the challenge to the *Attacker*.
4. **Query phase 2.** The *Attacker* continues to adaptively issue decryption queries $q_{m+1}, \cdots, q_n$. The *Challenger* responds as in the phase 1. These queries may be asked adaptively as in **Query phase 1**. The decryption query $q_j = C^*$ is not permitted, where $0 \le j \le n$.
5. **Guess.** Finally, the *Attacker* outputs a guess $e' \in \{0,1\}$ for e and wins the game if $e' = e$.

The encryption scheme is secure against chosen ciphertext attack, if the *Attacker* has a negligible advantage $\varepsilon = \left| \Pr(e = e') - \frac{1}{2} \right|$ to win the game.

3.4. Complexity assumptions

— ***Computational Diffie-Hellman Assumption***

Given g^a and g^b for some $a, b \in Z_q^*$, compute $g^{ab} \in G_1$. A (τ, ε)-CDH attacker in G_1 is a probabilistic machine Ω running in time τ such that

$$Succ_{G_1}^{cdh}(\Omega) = \Pr[\Omega(g, g^a, g^b) = g^{ab}] \ge \varepsilon$$

where the probability is taken over the random values a and b. The CDH problem is (τ, ε)-intractable if there is no (τ, ε)-attacker in G_1. The CDH assumption states that it is the case for all polynomial τ and any non-negligible ε.

— ***Decisional Diffie-Hellman Assumption[11]***

We say that an algorithm π that outputs $b \in \{0,1\}$ has advantage ε in solving the **Decisional Diffie-Hellman** (**DDH**) problem in G_1 if

$$|\Pr[\pi(g, g^a, g^b, g^c, e(g,g)^{abc}) = 0] - \Pr[\pi(g, g^a, g^b, g^c, T) = 0]| \ge \varepsilon$$

where the probability is over the random bit of π, the random choice of $a, b, c \in Z_q^*$, and the random choice of $T \in G_2$. The **DDH** problem is intractable if there is no attacker in G_1 can solve the **DDH** with non-negligible ε.

— ***k-Strong Diffie-Hellman (k-SDH) Assumption[10]***

Given $\{g, g^x, g^{x^2}, \cdots, g^{x^k}\}$ for a random number $x \in Z_q^*$, the attacker adaptively chooses random $c \in Z_q^*$ and computes $g^{(c+x)^{-1}}$. A (τ, ε)-k-SDH attacker in G_1 is a probabilistic machine Ω running in time τ such that

$$Succ_{G_1}^{k-sdh}(\Omega) = \Pr[\Omega(g, g^x, g^{x^2}, \cdots, g^{x^k}, c) = g^{(c+x)^{-1}}] \ge \varepsilon$$

We say the k-SDH problem is (τ, ε)-intractable if there is no (τ, ε)-attacker in G_1.

— ***k-Exponent (k-E) assumption [10].***

Given $\{g, g^x, g^{x^2}, \cdots, g^{x^k}\}$ for a random number $x \in Z_q^*$, compute $g^{x^{k+1}}$. A (τ, ε)-k-SDH attacker in G_1 is a probabilistic machine Ω running in time τ such that

$$Succ_{G_1}^{k-E}(\Omega) = \Pr[\Omega(g, g^x, g^{x^2}, \cdots, g^{x^k}) = g^{x^{k+1}}] \ge \varepsilon$$

We say the **k-E** problem is (τ, ε)-intractable if there is no (τ, ε)-attacker in G_1.

4 Group oriented encryption scheme

4.1 Initialize

Let G_1 be a cyclic multiplicative group generated by g, whose order is a prime q and G_2 be a cyclic multiplicative group of the same order q. A bilinear pairing is a map: $e: G_2 \times G_1 \to G_2$ that can be efficiently computed. Define three cryptographic hash functions:

$$H: \{0,1\}^* \to Z_q \qquad G: G_2 \to \{0,1\}^l \qquad H_1: G_1 \to Z_q^*$$

PKG chooses $a \in Z_q^*$ and $g_2 \in G_1$ uniformly at random, and computes $g_1 = g^a$. The master private key is a, and the master public keys are (g_1, g_2).

4.2. KeyGen

PKG chooses $k \in Z_q^*$ uniformly at random for the group A, and then publishes $PK_A = g^k$ and $VK_A = g^{a^2 k}$ as group A's public key. The member p_i's private key can be generated as follows:

1. PKG chooses $r_i \in Z_q^*$ uniformly at random.
2. compute and output $d_{i1} = g^{H(ID_i)r_i} g_2^{ar_i}$, $d_{i2} = g^{ar_i}$, and $d_{i3} = g^{ak} g^{H(ID_i)r_i}$.

The member p_i's private key is $d_i = \{d_{i1}, d_{i2}, d_{i3}\}$, where ID_i denotes the identity of p_i.

4.3. KeyVer

After receiving the private key distributed from PKG, the member p_i verifies the validity of the key by the following equation.

$$e(d_{i3}, g^a) \stackrel{?}{=} e(VK_A, g) e(g^{H(ID_i)}, d_{i2}) \tag{1}$$

If above equation holds, p_i accepts the private key, otherwise outputs ERROR message. We say the member p_i can verify the key since

$$\begin{aligned} e(d_{i3}, g^a) &= e(g^{ak} g^{H(ID_i)r_i}, g^a) \\ &= e(g, g^{a^2 k}) e(g^{H(ID_i)}, g^{ar_i}) \\ &= e(VK_A, g) e(g^{H(ID_i)}, d_{i2}) \end{aligned}$$

4.4. Encrypt

In order to encrypt a message $M \in \{0,1\}^l$ for the group A, the sender first chooses $s \in Z_q^*$ uniformly at random, and computes the ciphertext

$$c_1 = G(e(g_1, PK_A)^s) \oplus M \qquad c_2 = g^s \qquad c_3 = g_2^s \qquad c_4 = H_1(g^{ks}) \qquad c_5 = g^{(s+h)^{-1}}.$$

The ciphertext for message M is $c = (c_1, c_2, c_3, c_4, c_5)$, where $h = H(c_1 \| c_2 \| c_3 \| c_4)$. The sender sends the ciphertext to all the members in the group A by broadcast over Internet, where $a \| b$ denotes concatenation of a and b.

4.5. Decrypt

After receiving the encrypted message $c = (c_1, c_2, c_3, c_4, c_5)$, the user $p_i \in A$ decrypts the ciphertext as follows:

1. compute $T = e(c_2, d_{i3}) e(c_3, d_{i2}) / e(c_2, d_{i1})$.
2. compute $M = c_1 \oplus G(T)$.

We say the **Decrypt** is correct, since

$$\begin{aligned} T &= e(c_2, d_{i3}) e(c_3, d_{i2}) / e(c_2, d_{i1}) \\ &= e(g^s, g^{ak} g^{H(ID_i)r_i}) e(g_2^s, g^{ar_i}) / e(g^s, g^{H(ID_i)r_i} g_2^{ar_i}) \\ &= e(g^s, g^{ak}) e(g^s, g^{H(ID_i)r_i}) e(g_2^s, g^{ar_i}) / e(g^s, g^{H(ID_i)r_i}) e(g^s, g_2^{ar_i}) \\ &= e(g, g)^{aks} \end{aligned}$$

Then p_i gets the message M= $c_1 \oplus G(T)$.

5 Security

A group oriented encryption scheme is presented in [1]. This scheme is secure against adaptively chosen ciphertext attack. However, when two or more users work together, they can

forge a valid private key that can be used by any one. We assume that two users $p_i, p_j \in A$ will cooperate with each other to forge a private key. User p_i 's private key is $\{ d_{i1} = g^{r_i} g_2^{ar_i}, d_{i2} = g^{ar_i}, d_{i3} = g^{ak} g^{r_i} \}$, and p_j 's private key is $\{ d_{j1} = g^{r_j} g_2^{ar_j}, d_{j2} = g^{ar_j}, d_{j3} = g^{ak} g^{r_j} \}$. They compute as follows.

$$d_{k1} = (d_{i1} \cdot d_{j1})^{1/2} = g^{(r_i + r_j)/2} g_2^{a(r_i + r_j)/2}$$
$$d_{k2} = (d_{i2} \cdot d_{j2})^{1/2} = g^{a(r_i + r_j)/2}$$
$$d_{k3} = (d_{i3} \cdot d_{j3})^{1/2} = g^{ak} g^{(r_i + r_j)/2}.$$

It means that they have forged a valid private key $\{d_{k1}, d_{k2}, d_{k3}\}$ and this key can be used to decrypt any ciphertext encrypted for the group.

In order to overcome this shortcoming, we present an improved scheme that is secure against collude attack in section 4. We say there exists collude attack if two or more users cooperate with each other to compute a new valid private key for designated user. In the improved scheme, we assume that p_i 's private key is $\{ d_{i1} = g^{H(ID_i)r_i} g_2^{ar_i}, d_{i2} = g^{ar_i}, d_{i3} = g^{ak} g^{H(ID_i)r_i} \}$ and p_j 's private key is $\{ d_{j1} = g^{H(ID_j)r_j} g_2^{ar_j}, d_{j2} = g^{ar_j}, d_{j3} = g^{ak} g^{H(ID_j)r_j} \}$. These two users still perform above computing

$$d_{k1} = (d_{i1} \cdot d_{j1})^{1/2} = g^{(H(ID_i)r_i + H(ID_j)r_j)/2} g_2^{a(r_i + r_j)/2}$$
$$d_{k2} = (d_{i2} \cdot d_{j2})^{1/2} = g^{a(r_i + r_j)/2}$$
$$d_{k3} = (d_{i3} \cdot d_{j3})^{1/2} = g^{ak} g^{(H(ID_i)r_i + H(ID_j)r_j)/2},$$

and send the result $\{d_{k1}, d_{k2}, d_{k3}\}$ to the user $p_k \in A$ as private key. After receiving the private key, the user p_k can decrypt a ciphertext via this key. Since the private key is related to the identity of the receiver, we say that the private key that sent to p_k is not valid if it is not generated by using p_k 's identity ID_k. The user p_k can verify the key as we have mentioned in section 4.3. To the private key $\{d_{k1}, d_{k2}, d_{k3}\}$, the user has ability to detect the collude attack since

$$e(d_{k3}, g^a) \neq e(VK_A, g) e(g^{H(ID_k)}, d_{k2}).$$

Then we have the following theorem.

Theorem1. *Suppose H is a strong one way function. Then any two or more users can't forge a valid private key for a designated user, i.e. our scheme is secure against collude attack.*

Proof. Assume that there are two users p_i with identity ID_i and $p_{j \neq i}$ with identity $ID_{j \neq i}$ want to forge the private key of the user p_k whose identity is ID_k. The users p_i and p_j choose $m, n \in Z_q^*$, respectively. And then these two users perform as follows.

$$d_{k1} = (d_{i1} \cdot d_{j1})^{1/(m+n)} = g^{(mH(ID_i)r_i + nH(ID_j)r_j)/(m+n)} g_2^{a(mr_i + nr_j)/(m+n)}$$
$$d_{k2} = (d_{i2} \cdot d_{j2})^{1/(m+n)} = g^{a(mr_i + nr_j)/(m+n)}$$
$$d_{k3} = (d_{i3} \cdot d_{j3})^{1/(m+n)} = g^{ak} g^{(mH(ID_i)r_i + nH(ID_j)r_j)/(m+n)}$$

They send $\{d_{k1}, d_{k2}, d_{k3}\}$ to the user p_k. As we have mentioned above, the private key can be used to decrypt a ciphertext encrypted for group A. Since $\{d_{k1}, d_{k2}, d_{k3}\}$ is p_k 's private key, it means that the equation (1) holds. Then we have

$$e(g^{ak} g^{(mH(ID_i)r_i + nH(ID_j)r_j)/(m+n)}, g^a) = e(g^{a^2 k}, g) e(g^{H(ID_k)}, g^{a(mr_i + nr_j)/(m+n)})$$
$$e(g^{mH(ID_i)r_i + nH(ID_j)r_j}, g) = e(g, g^{H(ID_k)(mr_i + nr_j)}).$$

It means that $H(ID_i) = H(ID_j) = H(ID_k)$. Since H is assumed to be a strong one way function, we have $ID_i = ID_j = ID_k$. It is contradictory to our assumption. In other words, any two or more users can't forge a valid private key for any designated user.

■

As we have mentioned, our encryption scheme is secure against adaptively chosen ciphertext attack, i.e. IND-CCA2. Then we give following theorem.

Theorem 2: *Suppose the **k-E** and **k-SDH** assumption holds. Then our encryption scheme is secure against adaptively chosen ciphertext attack (**IND-CCA2**).*

Proof: Assume that if the attacker Eve can break the encryption scheme via chosen ciphertext attack, we can prove that there exists challenger Alice that can solve **k-E** or **K-SDH** problems. In other words, given $g, g^a, g^s, g^{sb}, g^k, (h, g^{(h+s)^{-1}})$, Alice can compute $e(g,g)^{aks}$ or $(h', g^{(h'+s)^{-1}})$. The challenger Alice interacts with Eve by simulating G, H and H_1 random oracles and **Decrypt** oracle. It gives the receiver g^k and g^a as the public keys.

a) Seek Phase

G **queries:** To every new query q_i, except below special case, Alice chooses $G_i \in \{0,1\}^l$ uniformly at random as the answer and preserves the data (q_i, G_i) in $List\_1$.

H **queries:** To every new query h_i, Alice chooses $H_i \in Z_q^*$ uniformly at random as the answer and preserves the data (h_i, H_i) in $List\_2$.

H_1 **queries:** To every new query g_i, except below special case, Alice chooses $H_1^i \in G_1$ uniformly at random as the answer and preserves the data (g_i, H_1^i) in $List\_3$.

Decrypt queries: When attacker Eve makes a query on $c_i = (c_{i1}, c_{i2}, c_{i3}, c_{i4}, c_{i5})$, if it didn't ask the H before, the ciphertext will be rejected since both H and $h_i = H(c_{i1} \| c_{i2} \| c_{i3} \| c_{i4})$ are random. However, (h_i, c_2, c_5) should satisfy the following equation

$$e(g^{h_i} c_2, c_5) = e(g,g) \quad (2).$$

Since $h_i = H(c_{i1} \| c_{i2} \| c_{i3} \| c_{i4})$ is a random number, the probability of equation holding is negligible, that is to say if the attacker Eve hasn't queried H with $(c_{i1}, c_{i2}, c_{i3}, c_{i4})$, the probability of the challenger Alice rejecting the valid ciphertext is negligible. Otherwise, Alice can seek g_i that satisfies the following equation in $List\_3$.

$$e(g_i, g) = e(g^k, c_2) \quad (3)$$

According to the above equation (3), g_i should satisfy $g_i = c_2^k$. If g_i exists, the challenger Alice computes $e(g_i, g^a)$ and searches the matching answer G_i of the query $q_i = e(g_i, g^a)$ in $List\_1$ and then outputs $M_i = c_{i1} \oplus G_i$ as the corresponding **Decrypt** answer. If g_i doesn't exist, the challenger Alice chooses G_i uniformly at random and defines $G(q_i) = G_i$. Then the challenger Alice computes

$$M_i = c_{i1} \oplus G_i = c_{i1} \oplus G(q_i)$$

as the corresponding **Decrypt** answer.

Hereafter, when the attacker Eve queries H_1 with $g_i = c_2^k$, Alice answers with matching H_1^i. When the attacker Eve queries G with $q_i = e(g_i, g^a)$, Alice answers with matching G_i. Considering the randomness of these oracles, the attacker can't distinguish the simulative answer from the actual results. With above description, we say that the challenger Alice perfectly simulates G, H, H_1 and **Decrypt**.

b) Challenge Phase

The attacker Eve adaptively outputs two equal length messages (M_0, M_1). The challenger Alice chooses $T \in \{0,1\}^l$, $U \in Z_q^*$ and $j \in \{0,1\}$ uniformly at random and computes

$$c_1 = T \oplus M_j \qquad c_2 = g^s \qquad c_3 = g^{sb} \qquad c_4 = U \qquad c_5 = g^{(h+s)^{-1}}.$$

Let

$$T = G(e(g^{ks}, g^a)) \qquad U = H_1(g^{ks}) \qquad H(c_1, c_2, c_3, c_4) = h$$

When Eve queries H with (c_1, c_2, c_3, c_4), Alice uses h as the corresponding answer. When Eve queries H_1 with g^{ks}, which can be detected by equation (3), the challenger uses U as the

answer. When Eve queries G with $q_i = e(g^{ks}, g^a)$, Alice uses T as the answer. Because of T and U 's randomness, the attacker Eve can't tell the simulative answer from actual results.

c) Guess Phase

It is the same to simulate the oracles G, H and H_1 as above.

Decrypt queries: Assume that the attacker Eve has the ciphertext $(c_1', c_2', c_3', c_4', c_5')$. If Eve has not queried random oracle H with (c_1', c_2', c_3', c_4') or the ciphertext doesn't satisfy the equation (2), it will be rejected. Otherwise, if $(c_2', c_3') \neq (c_2, c_3)$, the simulation approach is as the **Seek Phase**. If $(c_2', c_3') = (c_2, c_3)$, the challenger Alice uses $c_1' \oplus M_j \oplus c_1$ as the answer. The attacker Eve can't tell the simulative result from the actual considering that

$$c_1 \oplus M_j \oplus c_1' = T \oplus M_j \oplus M_j \oplus T \oplus M .$$

All above simulation process is perfect, so the attacker Eve can't tell the simulative result from the actual.

According to the assumption, the attacker Eve can give the right answer about j with non-negligible advantage ε. Then we can say that the probability of Eve using $g_i = g^{sk}$ and $q_i = e(g^{ks}, g^a)$ to query H_1 and G respectively or constructing the valid ciphertext $(c_1', c_2, c_3, c_4', c_5') \neq (c_1, c_2, c_3, c_4, c_5)$ is ε.

To ciphertext $(c_1', c_2', c_3', c_4', c_5')$, the probability of $(c_2', c_3') = (c_2, c_3)$ is negligible because (c_2, c_3) is random to the attacker Eve in the phase of simulation. In the **Guess Phase**, if Eve didn't construct the ciphertext that satisfies $(c_1', c_2, c_3, c_4', c_5') \neq (c_1, c_2, c_3, c_4, c_5)$, then the decrypt result gotten from the **Decrypt** oracle which satisfies $(c_2', c_3') \neq (c_2, c_3)$ is the plaintext of $(c_1', c_2', c_3', c_4', c_5')$. According the simulation process, the plaintext information in $c_1 = T \oplus M_j$ is not been leaked out since T is isolated from $G(q_i)$.

In the process of simulating oracle G, the challenger answers all the queries using random numbers which isolates from T as long as the query is not $q_i = e(g^{ks}, g^a)$, so the probability of collision can be neglected. On the other hand, because $c_1 = T \oplus M_j$ and T are chosen uniformly at random, it is impossible to get any plaintext information directly from $(c_1, c_2, c_3, c_4, c_5)$. Furthermore, we can see from the process of simulating oracle H and H_1 that the attacker can't get any information from these two oracles. As indicated above, only three aspects may leak out the information about the plaintext. The first is the answer which oracle G outputs for the query $q_i = e(g^{ks}, g^a)$. The second is the matching answer of the query $g_i = g^{ks}$, and the third is the matching answer of the decrypt query $(c_1', c_2, c_3, c_4', c_5') \neq (c_1, c_2, c_3, c_4, c_5)$. If the attacker Eve queries the oracle H_1 with $g_i = g^{sk}$, the challenger Alice can detect and find it by the equation (3). Then Alice can compute $e(g_i, g^a)$ via the Eve. If the attacker Eve uses valid ciphertext $(c_1', c_2, c_3, c_4', c_5') \neq (c_1, c_2, c_3, c_4, c_5)$ to query **Decrypt** oracle, then the ciphertext must satisfy the equation (2), i.e.

$$e(g^{h'} c_2, c_5') = e(g, g),$$

where $h' \neq h$. With the properties of bilinear pairing, we have $c_5' = g^{(h'+s)^{-1}}$. The challenger Alice can compute $(h', g^{(h'+s)^{-1}})$ via the information fed back from the attacker Eve.

From above description, if the attacker Eve can output j 's right answer with a non-negligible advantage ε, then the challenger Alice can compute $e(g, g)^{aks}$ or $(h', g^{(h'+s)^{-1}})$ with non-negligible probability. It is contradictory to the complexity assumptions.

■

6 Conclusions

In an open network environment, clients maybe form some special groups because of reasonable relationship. For example, the clients in a virtual community can be considered as such group. Then, how to protect the group communication becomes a crucial problem, since lots of clients are involved in the communication and the instance of the network is variable, even worse, a malicious client is waiting a chance to attack the communication by any possible artifice. In this paper, we present a group-based encryption scheme which can withstand adaptively chosen ciphertext attack and collude attack. Finally, we prove its security in random oracle model.

References

1. Ma C., Mei Q., and Li J.. "Broadcast Group-oriented Encryption for Group Communication". Journal of Computational Information Systems 3:1 (2007) 63-71.
2. Fiat A. and Naor M.. "Broadcast Encryption". Advances in Cryptology-CRYPTO 1993, Springer-Verlag, Lecture Notes in Computer Science 773: 480-491.
3. Dalit Naor, Moni Naor, and Jeff Lotspiech. "Revocation and Tracing Schemes for Stateless Receivers". In Advances in Cryptology-CRYPTO 2001. Springer-Verlag, Lecture Notes in Computer Science 2139: 41-62.
4. D. Halevy and A. Shamir. "The 1sd broadcast encryption scheme". In Proceedings of Cryptology-CRYPTO 2002. Springer-Verlag, Lecture Notes in Computer Science 2442: 47-60.
5. Y. Dodis and Nelly Fazio. "Public key broadcast encryption for stateless receivers". In Proceedings of the Digital Rights Management Workshop 2002. Springer-Verlag, Lecture Notes in Computer Science 2696: 61-80.
6. D. Wallner, E. Harder, and R. Agee. "Key management for multicast: Issues and architectures". Available at ftp://ftp.ietf.org/rfc/rfc2627.txt, 1997.
7. M. Luby and J. Staddon. "Combinatorial Bounds for Broadcast Encryption". In Advances in Cryptology-EuroCrypt'98. Springer-Verlag, Lecture Notes in Computer Science 1403: 512-526.
8. A. Garay, J. Staddon, and A. Wool. "Long-Lived Broadcast Encryption". In Advances in Cryptology-CRYPTO 2000. Springer-Verlag, Lecture Notes in Computer Science 1880: 333-352.
9. D. Boneh, C. Gentry, and B. Waters. "Collusion Resistant Broadcast Encryption With Short Ciphertexts and Private Keys". In Advances in Cryptology-CRYPTO 2005. Springer-Verlag, Lecture Notes in Computer Science 3621: 258-275.
10. F. Zhang, R. Safavi-Naini, W. Susilo. An Efficient Signature Scheme from Bilinear Pairings and Its Applications. Practice and Theory in Public Key Cryptography-PKC 2004, Lecture Notes in Computer Science 2947, 277-290, Springer-Verlag, 2004.
11. D.Boneh and X. Boyen. Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles. Advances in Cryptology Eurocrypt 2004. Berlin:Springer-Verlag,2004: 223-238.
12. S. Goldwasser and S. Micali. Probabilistic Encryption. Journal of Computer and System Sciences, 1984, 28: 270-299.
13. C. Rackhoff and D. R. Simon. Non interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Advanced in Cryptology-CRYPTO'91. Springer-Verlag, 1992: 434-444.