Efficient Dynamic Data Encryption Algorithm for Mobile Ad Hoc Network

Abstract. Two proper threshold broadcast encryption schemes are proposed for the mobile ad hoc network. The initial scheme achieves constant size private keys and O(n-t)-size ciphertexts. Under n+1-Decision Bilinear Diffie-Hellman Exponent (n+1 BDHE) assumption, it is provable security in the selective-identity model. Based on the dual system encryption, we propose our main construction. It also has constant size private keys and O(n-t)-size ciphertexts. But it achieves full security under the static assumptions which are more natural than them in the existing schemes.

Introduction

The concept of Broadcast Encryption (BE) was introduced by Fiat and Naor in [1]. In a broadcast encryption scheme a broadcaster encrypts a message for some subset S of users who are listening on a broadcast channel. Any user in S can use his private keys to decrypt the broadcasts. Any user outside the privileged set S should not be able to recover the message. The threshold broadcast encryption (TBE) problem is generalization of the concept of broadcast encryption. It was first introduced by Ghodosi et al. [2]. TBE has some advantages over traditional threshold encryptions. It is specified as follows: (1) The trusted party is eliminated and the system can be set up by individual users independently; (2) The broadcaster can choose the privileged set and the threshold value at the time of encryption which allows a certain dynamism in the system.

Identity-Based encryption was first proposed by Shamir[3], which a major advantage was that it allowed one to encrypt a message by using recipient’s identifiers such as an email address. Now it has been an active area. The first practical identity-based encryption (IBE) scheme was proposed in 2001 by Boneh and Franklin [4], which was provably secure against adaptive chosen ciphertext attack in random oracle model. Then, many other kinds of identity-based encryption were proposed [5-9]. Identity-based cryptography significantly reduces the system complexity and the cost for establishing and managing the public key authentication framework known as PKI (Public Key Infrastructure). As a result, we focus on the construction of identity-based threshold broadcast encryption (IBTHBE) in this paper. To the best of our knowledge, very few works have dealt with this problem. In [10], Chai and Cao et al propose a scheme based on identity. But the length of the ciphertexts is n+1 and the security relies on the random oracles. Vanesa Daza et al propose another scheme [11]. However, its security is still relying on the random oracles. The recent work [12] has short ciphertexts, but the security of their scheme based on the identity (IBTHBE) is also relying on the random oracles. In [13], authors also proposed an efficient scheme in the standard model. But the security only achieves a weak security model-selective-identity model. In [14], an efficient scheme was proposed. However, the public key size is too long and computation cost is high, which is not suitable to the ad hoc networks. In addition, the security is based on a strong hardness assumption.

As a natural extension of the efforts to improve schemes in the standard model, we propose two new efficient identity-based threshold broadcast encryption schemes in this paper. The proposed schemes are constructed in the standard model. In the selective-identity security model, we reduce the security of our first scheme to the n+1-Decision Bilinear Diffie-Hellman Exponent (n+1 BDHE) assumption. Based on the dual system encryption[15,16], the second scheme achieves the full security under the static assumption. In addition, two schemes have the dynamic feature which is suitable to mobile ad hoc networks. In the proposed schemes, any user can dynamically join the system as a possible recipient, and the sender can dynamically choose the set of recipients S and the threshold value t.

Decisional bilinear Diffie-Hellman Exponent assumption (BDHE)

The decisional bilinear Diffie-Hellman Exponent (BDHE) problem is defined as follows. Algorithm B is given as input a random tuple (g, h0, y1, …, yN, yN+2, …, y2n+2, T), where yi = gyi. Algorithm B’s goal is to output 1 when T = e(g, h0)yi and 0 otherwise. Let TU = (g, h0, y1, …, yn, yN+2, …, y2n+2). Algorithm B that outputs b ∈ {0, 1} has advantage ε in solving decision BDHE in G if

\[|Pr[B(TU, e(g, h0)yi) = 1] - Pr[B(TU, T = 0)]| \leq \varepsilon.\]

The (ε, ε) decisional BDHE assumption holds if no t-time algorithm has a non-negligible advantage ε in solving the above game.

Identity-based Threshold Broadcast Encryption (IDTHBE)

More formally, an IDTHBE consists of five algorithms.

Setup The randomized Setup algorithm takes as input a security parameter k and outputs some public parameters params, which will be common to all the users of the system.

Extract The key generation algorithm is run by each user ID. It takes as input some public parameters params and returns a correspondence private key dID.

Threshold Encryption The encryption algorithm takes as input a set of public keys corresponding to a set P of n receivers, a threshold t satisfying 1 ≤ t ≤ n, and a message M. The output is a ciphertext C, which contains the description of P and t. Partial Decryption Partial Decryption algorithm takes as input a ciphertext C for the pair (P, t) and a secret key dID.
of a receiver. The output is a partial decryption value \( k_i \) or a special symbol \( \bot \).

**Decryption** The deterministic final decryption algorithm takes as input a ciphertext \( C \) for the pair \( (P, t) \) and \( t \) partial decryptions corresponding \( k_i \) to receivers in some subset \( S \subset P \). The output is a message \( m \) or a special symbol \( \bot \).

**Security Model**

**Setup** The challenger runs Setup. Then challenger gives the resulting common parameter to \( A \), and keeps master key secret. \( A \) issues the threshold parameters \( (n, t) \).

**Phase 1** \( A \) issues private key extraction and decryption queries adaptively. The adversary \( A \) adaptively issues queries \( q_1, \cdots, q_\alpha \), where \( \alpha \) is one of the following:

- On a private key extraction query upon \( ID_i \), the challenger runs \( \text{Extract} \) to generate the private key associated to \( ID_i \), then sends it to \( A \).
- On a decryption queries, the challenger runs \( \text{Decryption} \) to generate decryption shares and gives them to \( A \).

**Challenge** When \( A \) decides that phase 1 is over, it submits a set of identities \( S \), a threshold value \( t \) and two messages \((M_0, M)\) on which it wants to be challenged. The adversary’s choice of \( S \) is restricted to the identities that he did not request a private key for in Phase 1. The challenger runs \( \text{Encrypt} \) algorithm to obtain \((\text{Hdr}, K) = \text{Encrypt}(S, PK, f)\) and returns them to \( A \). Note, \( A \) may already learned about the private keys of at most \( t - 1 \). There is the natural constraint that \( S \) contains at most \( t - 1 \) corrupted identities.

**Phase 2** The adversary continues to issue queries \( q_{\alpha+1}, \cdots, q_\beta \), where \( \beta \) is one of the following:

- **Extraction query** (\( ID_i \)), as in phase 1;
- **Decryption query**, as in phase 1, but with the constraint that \( \text{Hdr} \neq \text{Hdr}' \). The challenger responds as in phase 1.

**Guess** Finally, the adversary \( A \) outputs a guess \( b' \in \{0, 1\} \) and wins the game if \( b = b' \).

We say that if the above indistinguishability game allow no decryption oracle query, then the IDTHBE scheme is only chosen plaintext (IND-fullID-CPA) secure. If the challenge identity is outputted before the setup algorithm, it is called selective-identity security model. There have been many methods to convert an IND-fullID-CPA scheme to an IND-fullID-CCA scheme. Therefore, we only focus on constructing the IND-fullID-CPA scheme in this paper.

**Our initial Constructions**

We first give an initial construction which will derive a high efficient scheme in the next section.

Let \( S = \{ ID_1, \cdots, ID_n \} \) be \( n \) players where \( ID_i \in Z_p \).

These users want to form an ad hoc network. Our construction works as follows:

**Setup:** To generate the system parameters, the PKG picks randomly generators \( \{ g, g_2, h \} \) in \( \mathbb{G} \) and an element \( \alpha \) from \( Z_p \). Note that any user \( ID_i \) will be associated to a different element \( t_i \). This can be done by defining \( t_i = f(ID_i) \) for some \( n-1 \) degree polynomial function \( f(x) \), where \( f(0) = \alpha \). PKG sets \( T_i = g_i^{\alpha} \), for \( 1 \leq i \leq n \) and \( g_i = g^{r_i} \). The public parameters PK are \( PK = \{ g, g_2, T_i, h, \ldots, h, 1 \leq i \leq n \} \) and \( \alpha \) is master key.

**Extract(ID) :** To generate a private key for a user \( ID_i \in Z_p \), the PKG picks randomly \( r_i \in Z_p \) and computes private keys as follows:

\[
d_{i_0} = (d_{a_1}, d_{a_2}, d_{i_0}) = (g_{i_0}^{H_i(a_1)}, g_{i_0}^{H_i(a_2)}, H_i)
\]

where \( H = \prod_{i=1}^{n} h_i^{\alpha} \).

**Threshold Encryption:** To encrypt a message \( M \) for a set \( S = \{ ID_1, \cdots, ID_n \} \) of \( n \) players, with threshold \( t \leq n \) for the decryption, the idea is to set up an \( (n, N) \)-threshold secret sharing scheme where \( N = 2n - t \). The public keys \( \{ T_s, \cdots, T_s \} \) of users implicitly define a \( n - 1 \) degree polynomial. The idea is to compute the values of this polynomial in the points \( x = 0 \) (This will lead to obtain the value of \( g_i \)). Then a sender acts as follows:

- Select a random element \( s \in Z_p \) and compute \( C_1 = g^s, C_2 = e(g_i, g_2)^{M} \) and \( C_1 = (\prod_i^{n} h_i^{\alpha}) \).

- Choose a set \( S' = \{ n-t \} \) dummy players, such that \( S \cap S' = \emptyset \). For each user \( ID_i' \in S' \), compute \( T_i' = \prod_{i \in S} T_i^{\alpha} \) and \( K_i = \frac{1}{e(T_i, g_2)} \), where \( \lambda_{g_i} \) denotes the Lagrange coefficients.

**The ciphertexts are** \( (C_1, C_2, C_i, K_i)^{ID_{S'}} \).

Note: \( K_i' = \frac{e(C_1, d_{i_0})}{e(d_{i_0}, C_2)} = \frac{e(g_i, g_2)^{\alpha}}{e(g_i, g_2)^{y_{i_0}}} \) by using Lagrange interpolation if \( f'(ID_i) \).

**Partial Decryption:** Given the ciphertexts \( (C_1, C_2, C_i, K_i)^{ID_{S'}} \), the receiver \( ID_i \in S \) with his corresponding private \( d_{i_0} \) computes as follows:

\[
K_i' = \frac{e(C_1, d_{i_0})}{e(d_{i_0}, C_2)} = \frac{1}{e(g_i, g_2)^{y_{i_0}}}
\]

**Decryption:** Given the valid ciphertexts \( (C_1, C_2, C_i, K_i)^{ID_{S'}} \), a subset \( S \subset S \) with \( |S| = t \) and corresponding \( t \) partial decryption \( K_i' \), the algorithm computes with the whole set \( S' = S \cup S' \) as follows:

\[
K' = \prod_{i \in S'} K_i'^{\alpha} = \frac{1}{e(g_i, g_2)^{y_{i_0}}} \text{ and } M = K \cdot C_2.
\]

**Our main construction**

The initial construction will be provable security in the selective-identity model. In this section, based on the dual system encryption over the composite group, we will give our main construction which will achieve the full security.

Let \( G \) be cyclic groups of order \( N = p_1 p_2 p_3 \) and \( t \) denote the maximum number of the set of possible users. Let \( S = \{ ID_1, \cdots, ID_n \} \) be \( n \) players where \( ID_i \in Z_p \). These users want to form an ad hoc network. Our scheme works as follows.

**Our construction works as follows:**

**Setup:** To generate the system parameters, the PKG picks randomly generators \( \{ g, g_2, h \} \) in \( \mathbb{G} \) and an element \( \alpha \) from \( Z_p \). Note that any user \( ID_i \) will be associated to a different element \( t_i \). This can be done by defining \( t_i = f(ID_i) \) for some \( n-1 \) degree polynomial
function \( f(x) \), where \( f(0) = \alpha \). PKG sets \( T_i = g^h \) for \( 1 \leq i \leq n \) and \( g_0 = g^\alpha \). The public parameters PK are

\[
PK = (g, g_1, g_2, T_1, \ldots, T_n, h)
\]

and \( \alpha \) is master key.

\textbf{Extract(ID)}: To generate a private key for a user \( ID_i \in \mathbb{Z}_N \), the PKG picks randomly \( r_i \in \mathbb{Z}_N \) and \( R_{i0}, R_{i1} \in G_p \), computes private keys as follows:

\[
d_{i0} = (d_{i0}, d_{i1}) = (g_0^*(h^0)^{r_i} R_{i0}, g^0 R_{i1})
\]

\textbf{Threshold Encryption}: It is same as initial scheme.

\textbf{Decryption}: It is same as initial scheme.

\textbf{Correctness(Partial Decryption)}: In fact, if the ciphertexts \( C = (C_0, C_1, C_2) \) is valid, then one can obtain the following equation holds.

\[
e(C_0, d_{i0}) = e(d_{i0}, \prod_{j 
eq i} g_{ij}^{d_{ij}}, C_1) = e(g_0^*(h^0)^{r_i} R_{i0}, \prod_{j 
eq i} (g_{ij}^{d_{ij}})^0, g^0)
\]

\[
e(h_0^*(h^0)^{r_i} R_{i0}, \prod_{j 
eq i} (g_{ij}^{d_{ij}})^0, g^0) = e(g_0^* \prod_{j 
eq i} (g_{ij}^{d_{ij}})^0, g^0) e(h_0^* R_{i0}, g^0) e(h_0^* R_{i0}, g^0)
\]

\[
e(h_0^* R_{i0}, g^0) e(h_0^* R_{i0}, g^0) = 1
\]

In the previous equations, the orthogonality property of \( G_p, G_p, G_p \) is used. It is described simply as follows.

\textbf{Lemma 1}[17] When \( h_i \in G_p, h_j \in G_p \), for \( i \neq j \),

\[
e(h_i, h_j) \) is the identity element in \( G_1 \).

By using this lemma, one can obtain

\[
e(R_{i0}, g^0) = e(h_0^* R_{i0}, g^0) e(h_0^* R_{i0}, g^0) = 1
\]

\textbf{Table 1. Comparison of the Efficiency with the others Scheme}\n
<table>
<thead>
<tr>
<th>Schemes</th>
<th>C-Size</th>
<th>pk Size</th>
<th>Parings</th>
<th>S.M</th>
</tr>
</thead>
<tbody>
<tr>
<td>[10]</td>
<td>n + 1</td>
<td>1</td>
<td>1 + 2t</td>
<td>Full</td>
</tr>
<tr>
<td>[11]</td>
<td>n + t</td>
<td>3</td>
<td>3 + 2t</td>
<td>Full</td>
</tr>
<tr>
<td>[12]</td>
<td>n</td>
<td>2</td>
<td>0 + t</td>
<td>Full</td>
</tr>
<tr>
<td>[13]</td>
<td>O(n + t)</td>
<td>2</td>
<td>0 + t</td>
<td>s-ID</td>
</tr>
<tr>
<td>[14]</td>
<td>O(n + t)</td>
<td>2</td>
<td>0 + t</td>
<td>s-ID</td>
</tr>
<tr>
<td>Initial</td>
<td>O(n + t)</td>
<td>2</td>
<td>0 + t</td>
<td>Full</td>
</tr>
</tbody>
</table>

\textbf{Efficiency Analysis}\n
In our scheme, the size of ciphertexts is \( O(n - t) \) and the size of private key is constant as it consists of two group elements. This is the efficient construction which has full security in the standard model for the identity-based threshold broadcast encryption. In addition, if the values \( e(g_i^1, g_2^1) \) and \( e(T_i^1, g_2^1) \) can be precomputed and cached, so no pairing computations are needed at the phase of Threshold Encryption. Another advantage in our scheme is the natural security basis. The security of our initial scheme is achieved in the decision BDH assumption. And the security of our main scheme is reduced to three static assumptions. Table 1 gives the efficiency comparison between ours and the others IDTHBEs. From the table 1, our main construction has a natural security basis over [14]. In addition, our main scheme has much shorter public keys than the scheme in [14].

Note: R.O. denotes the random oracles. C-Size is the size of ciphertext and pk is the private keys. SM denotes the security model. Full and s-ID are full security and selective-identity model.

\textbf{Security Analysis}\n
At first, we give the security proof of initial scheme.

\textbf{Theorem 1} Suppose the decision BDHE assumption holds. Then the proposed initial scheme above is semantically secure against selective identity, chosen plaintext attacks.

\textbf{Proof} Suppose an adversary \( A \) can attack the initial scheme with advantage \( \varepsilon \). We will show that there is an algorithm \( B \) that solves the decision BDHE problem in \( G \) with the advantage \( \alpha \). For a generator \( g \in G \) and \( \alpha \in \mathbb{Z}_p \), set \( y_i = g^{\alpha_i} \in G \). Algorithm \( B \) is given as input a random tuple \((g, h_0, y_1, \ldots, y_n, y_{n+2}, \ldots, y_{2n+2}, T)\). Algorithm \( B \)'s goal is to output 1 when \( T = e(g, h_0)^{y_{n+1}} \) and 0 otherwise.

Algorithm \( B \) works by interacting with \( A \) in a threshold selective-identity game as follows. \textbf{Init}: \( A \) outputs a set \( S' = \{(ID_1', \ldots, ID_m')\} \) of identities that it wants to corrupt, and a set \( \tilde{S} \) of identities that it wants to corrupt, with \( |\tilde{S}| \leq n - 1 \) and \( |\tilde{S}| \leq t - 1 \).

\textbf{Setup}: \( B \) does the following:

- First, \( B \) picks a random \( \gamma \in \mathbb{Z}_p^* \), sets \( g_0 = \gamma_1 = g^\alpha \) and \( g_2 = y_\gamma g^\gamma \). Then \( B \) selects randomly \( \gamma_1, \gamma_2, \ldots, \gamma_m \in \mathbb{Z}_p^* \) and sets \( h_0 = g_{\gamma_1}^{y_0} / g_{\gamma_2}^{y_0} \) for \( 1 \leq i \leq n \). In addition, \( B \) selects randomly \( v \in \mathbb{Z}_p^* \) and sets \( h = g_v^{y_{n+i}} \).

- Next, \( B \) selects \( n - 1 \) random integers \( \alpha_1, \alpha_2, \ldots, \alpha_{n-1} \in \mathbb{Z}_2 \). Let \( f(x) \) be the degree \( n - 1 \) polynomial implicitly defined to satisfy \( f(0) = \alpha \) and \( f(ID_i) = \alpha_i \) for \( ID_i \in \tilde{S} \), note that \( B \) does not know \( f \) since it does not know \( \alpha \). For \( ID_i \in \tilde{S} \), \( B \) computes \( T_i = g^{\alpha_i} \). Otherwise, \( B \) computes \( \alpha_i = f(ID_i) = \lambda_\alpha + \sum_{j=1}^{n-1} \lambda_\alpha \gamma_j \) with the Lagrange coefficients \( \lambda_\alpha \). Note that these Lagrange coefficients are easily calculated since they do not depend on \( f \). Then \( B \) selects \( T_i = g_v^{y_{n+i}} \prod_{j=1}^{n-1} T_j^{\gamma_\gamma_{j-1}} \).

Finally, \( B \) gives the public keys \( PK = (g, g_1, g_2, h_0, T_1, \ldots, T_n, h_1, \ldots, h_t) \) to \( A \).

\textbf{Query phase 1}: \( A \) issues up to \( q_d \) private key generation queries to the uncorrupt servers. Each query \( q_d \) works as follows: Suppose \( A \) asks for the private key corresponding to an identity \( ID_i \in S' \). The restriction ensures that \( |ID_0 - ID_i'| \neq 0 \). \( B \) first computes the Lagrange coefficients \( \lambda_\alpha, \lambda_\alpha, \ldots, \lambda_{n-1} \) such that \( t = f(ID_i) = \lambda_\alpha \alpha + \sum_{j=1}^{n-1} \lambda_\alpha \gamma_j \). Then \( B \) selects a random \( r \in \mathbb{Z}_p^* \) and computes the corresponding private keys as follows:

\[
d_{i0} = (d_{i0}, d_{i1}, d_{i2}) = (g_0^\gamma \prod_{j=1}^{n-1} S_{i0,j}^{y_{j+1}} (g_v^{y_{n+i}})^{y_{n+i}} \prod_{j=1}^{n-1} T_j^{\gamma_{j-1}}) \prod_{i=1}^{n-1} (g_0^{y_{i-1}})^{y_{n+i}}
\]

One can verify \( d_{i0} = (d_{i0}, d_{i1}, d_{i2}) \) is the valid simulation. In fact,
\[ g_1^{k_1} \sum_{i=0}^{n-1} \left( g_1^m \cdot g_1^{\alpha_{ij}} \right) \prod_{j \neq i} g_1^{y_{ij}} = g_1^{k_1} \sum_{i=0}^{n-1} \left( y_{ni}^{\alpha_{ij}} \right) \left( y_{ni}^{\alpha_{ij}} \right) = g_1^{k_1} \sum_{i=0}^{n-1} \left( y_{ni}^{\alpha_{ij}} \right) \cdot \left( y_{ni}^{\alpha_{ij}} \right) \]

\[ g_2^{k_2} \sum_{i=0}^{n-1} \left( g_2^{\alpha_{ij}} \right) \prod_{j \neq i} g_2^{y_{ij}} = g_2^{k_2} \sum_{i=0}^{n-1} \left( y_{ni}^{\alpha_{ij}} \right) \left( y_{ni}^{\alpha_{ij}} \right) = g_2^{k_2} \sum_{i=0}^{n-1} \left( y_{ni}^{\alpha_{ij}} \right) \cdot \left( y_{ni}^{\alpha_{ij}} \right) \]

\[ = g_1^{k_1} \sum_{i=0}^{n-1} \left( g_1^m \cdot g_1^{\alpha_{ij}} \right) \prod_{j \neq i} g_1^{y_{ij}} = g_1^{k_1} \sum_{i=0}^{n-1} \left( g_1^m \cdot g_1^{\alpha_{ij}} \right) \cdot \left( g_1^{\alpha_{ij}} \right) \cdot \left( g_1^{\alpha_{ij}} \right) \]

\[ = g_2^{k_2} \sum_{i=0}^{n-1} \left( g_2^{\alpha_{ij}} \right) \prod_{j \neq i} g_2^{y_{ij}} = g_2^{k_2} \sum_{i=0}^{n-1} \left( g_2^{\alpha_{ij}} \right) \cdot \left( g_2^{\alpha_{ij}} \right) \cdot \left( g_2^{\alpha_{ij}} \right) \]

**Phase 2:** The adversary continues to issue queries and \( B \) responds as in phase 1.

**Guess:** \( A \) outputs a guess \( b' \in \{0,1\} \) and wins the game if \( b' = b \). If \( b' = b \), \( B \) will output 1 to indicate that \( B \) solves the DDBHE problem, otherwise it outputs 0 to mean that it learns nothing from \( C' \).

When \( A \) outputs 1, it means \( \Pr(b = b') - \frac{1}{2} \geq \epsilon \). Otherwise \( \Pr(b = b') = \frac{1}{2} \). Therefore, we have

\[ \Pr(B(TU,e(g,g)) = 0) - \Pr(B(TU,T = 0) = \frac{1}{2} + \epsilon - \frac{1}{2} \geq \epsilon. \]

**Static Hardness Assumption**

In this section, we give our complex assumption. These assumptions have been used in [15,16].

**Assumption 1** (Subgroup decision problem for 3 primes) Given \( \langle N = p_1 \cdot p_2 \cdot p_3, G, G_1, e \rangle \), select randomly \( g \in G_{p_1}, X_1 \in G_{p_1}, T_1 \in G_{p_2}, T_2 \in G_{p_3} \) and set \( D = \langle N = p_1 \cdot p_2 \cdot p_3, G, G_1, e, g, X_1 \rangle \). It is hard to distinguish \( T_1 \) from \( T_2 \). The advantage of an algorithm is defined as

**Assumption 2** Given \( \langle N = p_1 \cdot p_2 \cdot p_3, G, G_1, e \rangle \), pick randomly \( g, l_1, g_1 \in G_{p_1}, Y_1 \in G_{p_1}, T_1 \in G_{p_2}, T_2 \in G_{p_3} \) and set \( D = \langle N = p_1 \cdot p_2 \cdot p_3, G, G_1, e, g, X_1 \rangle \). Then select \( T_1 \in G, T_2 \in G_{p_3} \), at random. It is hard to distinguish \( T_1 \) from \( T_2 \).

**Next, we will prove the security of the main scheme. We first define semi-functional keys and semi-functional ciphertexts. Let \( g_2 \) denote a generator of \( G_{p_2} \).**

**Semi-functional keys:** At first, a normal key \( \left( \tilde{d}_1, \tilde{d}_2, \tilde{d}_3 \right) \) is obtained using the Extract algorithm. Then some random elements \( y_0, y_1, y_2 \) are chosen in \( Z_N \). The semi-functional keys are set as follows:

\[ d_0 = \tilde{d}_1 g_0^y, \quad d_1 = \tilde{d}_2 g_1^y, \quad d_2 = \tilde{d}_3 g_2^y. \]

**Semi-functional ciphertexts:** At first, a normal semi-functional ciphertext \( \langle C'_0, C'_1, C'_2 \rangle \) is obtained using the Encrypt algorithm. Then two random elements \( \tilde{\lambda}_1, \tilde{\lambda}_2 \) are chosen in \( Z_N \). The semi-functional ciphertexts are set as follows:

\[ C_0 = C'_0, \quad C_1 = C'_1 \tilde{g}_1^{\tilde{\lambda}_1}, \quad C_2 = C'_2 \tilde{g}_2^{\tilde{\lambda}_2}. \]

We organize our proof as a sequence of games:

**Game real:** This is a real threshold IBHE security game. For \( 0 \leq i \leq \ell \), the Game is defined as follows.

**Game:** Let \( \Omega \) denote the set of private keys which the adversary queries during the games. This game is a real IBHE security game with the two exceptions: (1) The challenge ciphertext will be a semi-functional ciphertext on the challenge set \( S' \). (2) The first \( i \) keys will be semi-functional private keys. The rest of keys in \( \Omega \) will be normal.
Note: In game$_0$, the challenge ciphertext is semi-functional. In game$_n$, the challenge ciphertexts and all keys are semi-functional.

Game$_{final}$: This game is same with Game$_{q}$ except that the challenge ciphertext is a semi-functional encryption of random group element of $G_1$.

Lemma 2 Suppose that there exists an algorithm $A$ such that $\text{Adv}_{\text{ciphertext}}^G A - \text{Adv}_{\text{ciphertext}}^C A = \epsilon$. Then we can build an algorithm $B$ with advantage $\epsilon$ in breaking Assumption 1.

Lemma 3 Suppose that there exists an algorithm $A$ that makes at most $q$ queries and such that $\text{Adv}_{\text{ciphertext}}^G A - \text{Adv}_{\text{ciphertext}}^C A = \epsilon$ for $1 \leq k \leq q$. Then we can build an algorithm $B$ with advantage $\epsilon$ in breaking Assumption 2.

Lemma 4 Suppose that there exists an algorithm $A$ that makes at most $q$ queries and such that $\text{Adv}_{\text{ciphertext}}^G A - \text{Adv}_{\text{ciphertext}}^C A = \epsilon$. Then we can build an algorithm $B$ with advantage $\epsilon$ in breaking Assumption 3.

By using the proof of Theorem 1 and techniques in [16], we can obtain the proof of these lemmas. For the concision, we omit them here. Then we have the following theorem.

Theorem 5 If Assumption 2, 3 and 4 hold, then our scheme is IND-ID-CPA secure.

Conclusions
Two new constructions of identity-based threshold broadcast encryption are proposed for ad hoc networks. In proposed schemes, the broadcaster can dynamically choose the set of recipients and the threshold value $t$. Both schemes have short ciphertexts, where the length of ciphertexts achieves $O(n^2)$. In addition, we reduce their security to the decision $\Gamma+1$-BDHE problem and some static assumptions respectively.

Unfortunately, in our schemes, the total number of possible users must be fixed in the setup. It is an interesting problem to construct a scheme without the above constraints in the standard model.

Acknowledgement:
This work is supported in part by the Nature Science Foundation of China under grant (61100231, 6097119, 61100165), the National Basic Research Program of China(973) under grant 2007CB311201 and the Fundamental Research Funds for the Central Universities.

REFERENCES