Identity-based Threshold Broadcast Encryption in the Standard Model

Leyou Zhang[1] and Yupu Hu[2] and Qing Wu[3]
[1] Department of Mathematical Science, Xidian University,
Xi'an, 710071, China
[e-mail: leyouzhang77@yahoo.com.cn]
[2] Key Laboratory of Computer Networks and Information Security,Ministry of Education,
Xidian University, Xi'an, 710071, China
[e-mail: xidianzly@163.com]
[3] School of Automation, Xi'an Institute of Posts and Telecommunications,
Xi'an, 710061, China;
*Corresponding author: Leyou Zhang

Received March 3, 2010; revised May 23, 2010; accepted June 15, 2010;
published June 30, 2010

Abstract

In an identity-based threshold broadcast encryption (IDTHBE) scheme, a broadcaster chooses a set of n recipients and a threshold value t, and the plaintext can be recovered only if at least t receivers cooperate. IDTHBE scheme is different from the standard threshold public key encryption schemes, where the set of receivers and the threshold value are decided from the beginning. This kind of scheme has wide applications in ad hoc networks. Previously proposed IDTHBE schemes have ciphertexts which contain at least n elements. In addition, the security of theses schemes relies on the random oracles. In this paper, we introduce two new constructions of IDTHBE for ad hoc networks. Our first scheme achieves $|S|$-size private keys while the modified scheme achieves constant size private keys. Both schemes achieve approximately (n-t)-size ciphertexts. Furthermore, we also show that they are provable security under the decision bilinear Diffie-Hellman Exponent (BDHE) assumption in the standard model.

Keywords: Identity-based encryption, broadcast encryption, threshold broadcast encryption, provable security, standard model

This paper was partially supported by the Nature Science Foundation of China under grant (60970119, 60803149), the National Basic Research Program of China(973) under grant 2007CB311201 and the Fundamental Research Funds for the Central Universities.

DOI: 10.3837/tiis.2010.06.012

1. 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/her private key to decrypt the broadcast. 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]. In their scheme, there was a set of N receivers and a subset of n privileged receivers. A broadcaster encrypts a message and broadcasts the corresponding ciphertext to all receivers. In such a way, the cooperation of at least t users from the privileged set can recover the original message. TBE has some advantages over traditional threshold encryptions: (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. Note that, when t=1, the resulting scheme will be a broadcast encryption scheme [1].

Identity-Based encryption was originally 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][6][7][8][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* proposed a scheme based on identity. But the length of the ciphertexts was n +1 and the security relied on the random oracles. Vanesa Daza *et al* proposed another scheme [11]. However, its security was still relying on the random oracles. The recent work [12] had short ciphertexts, but the security of their IBTHBE scheme also relied on the random oracles.

In this paper, we propose two new efficient identity-based threshold broadcast encryption schemes. The proposed schemes are constructed in the standard model (without using random oracles) and achieve approximately (n-t)-size ciphertexts. The threshold value and the privileged set can be picked dynamically by the broadcaster. In addition, under the selective-identity security model, we reduce the security of our schemes to the decision bilinear Diffie-Hellman Exponent (BDHE) assumption.

2. Preliminaries

2.1 Bilinear Groups

We briefly review bilinear maps and use the following notations:

1. G and G_1 are two (multiplicative) cyclic groups of prime order p;
2. g is a generator of G .
3. e is a bilinear map $e:\ G \times G \to G_1$.

Let G and G_1 be two groups as above. A bilinear map is a map $e : G \times G \to G_1$ with the properties:

1. Bilinearity: for all $u, v \in G$, $a, b \in Z_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$.
3. Computability**:** There is an efficient algorithm to compute $e(u, v)$ for all $u, v \in G$.

2.2 Decisional bilinear Difiie-Hellman Exponent assumption (BDHE)

The decisional bilinear Difiie-Hellman Exponent (BDHE) problem is defined as follows. Algorithm B is given as input a random tuple

$$(g\ , h_0, y_1, \cdots, \ y_n\ , y_{n+2}, \cdots , y_{2n+2}, T),$$

where $y_i = g^{\alpha^i}$. Algorithm B's goal is to output 1 when $T = e(g,h_0)^{\alpha^{n+1}}$ and 0 otherwise. Let TU =(g , h_0, $y_1, \cdots , y_n, y_{n+2}, \cdots , y_{2n+2}$). Algorithm B that outputs $b \in \{0,1\}$ has advantage ε in solving decision BDHE in G if

$$|Pr[B(TU, e(g,h_0)^{\alpha^{n+1}}) = 0]\text{-}Pr[B(TU, T) = 0]| \leq \varepsilon\ .$$

Definition 1: The (t, ε) decisional BDHE assumption holds if no t-time algorithm has a non-negligible advantage ε in solving the above game.

2.3 Identity-based Threshold Broadcast Encryption (IDTHBE)

More formally, a threshold broadcast encryption scheme 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_i. It takes as input some public parameters params and returns a correspondence private key d_{ID_i} .

***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 $1 \leq t \leq\ n$, 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 d_{ID_i} of a receiver. The output is a partial decryption value k_i or a special symbol $\perp$.

***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 $\perp$.

2.4 Security Model

There have been many methods to convert an IND-sID-CPA (Chosen plaintext secure under the selective-identity model) scheme to an IND-sID-CCA (Chosen ciphertext secure under the selective-identity model) scheme. Therefore, we only focus on constructing the IND-sID-CPA scheme in this paper. To define the notion of IND-sID-CPA, let us consider the following game between an adversary A and a challenger:

***Init*:** The adversary A firstly outputs a set $S^* = \{ ID_1^*, \cdots, ID_n^* \}$ of identities that he wants to attack and a set $\tilde{S}$ of identities that he wants to corrupt, with $|S^*| \leq n-1$ and $|\tilde{S} \cap S^*| \leq t-1$.

***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).

***Query Phase* 1:** The adversary *A* adaptively issues private key extraction queries $q_1, \cdots, q_{s0}$, where q_i is one of the following:

Extraction query: On a private key extraction query upon ID_i with the constraint that $ID_i \notin S^*$, the challenger runs *Extract* to generate the private key associated to ID_i, then sends it to *A*.

***Challenge*:** When *A* decides that phase 1 is over, *A* outputs two same-length messages M_0 and M_1 on which it wishes to be challenged. The challenger picks a random $b \in \{0,1\}$ and sets the challenge ciphertext C^* =*Encrypt*(*params*, M_b, S^*). The challenger returns C^* to *A*. Note, *A* may already have learned about the private keys of at most t-1.

***Query phase* 2:** The adversary continues to issue queries $q_{s0+1}, \cdots, q$, where q_i is one of the following:

Extraction query (ID_i): as in phase 1.

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

3. New Constructions

3.1 Basic Construction

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 $\{g, g_2, h, h_1, \cdots, h_n\}$ in G and an element α 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 $T_i = g^{t_i}$ for $1 \le i \le n$ and $g_1 = g^{\alpha}$. The public parameters *PK* are

$$PK = (g, g_1, g_2, T_1, \cdots, T_n, h, h_1, \cdots, h_n)$$

and the master key is α.

***Extract*(*ID*$_i$) :** To generate a private key for a user $ID_i \in Z_p$, the PKG picks randomly $r_i \in Z_p$, and outputs the private key:

$$\begin{aligned} d_{ID_i} &= (d_{i0}, d'_{i0}, d_{i1}, \cdots, d_{i(i-1)}, d_{i(i+1)}, \cdots, d_{in}) \\ &= (g_2^{t_i}(h_i^{ID_i} h)^{r_i}, g^{r_i}, h_1^{r_i}, \cdots, h_{i-1}^{r_i}, h_{i+1}^{r_i} \cdots, h_n^{r_i}). \end{aligned}$$

***Threshold Encryption*:** To encrypt a message M for a set $S = \{ID_1, \cdots, ID_n\}$ of n players, with threshold $t \le n$ for the decryption, the idea is to set up an (n, N)-threshold secret sharing scheme, where $N = 2n - t$. The n public keys ($T_1, \cdots, T_n$) 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_1).Then a sender acts as follows:

- Select a random element $s \in Z_p^*$ and compute $C_1 = g^s$, $C_2 = e(g_1, g_2)^s M$ and

$C_3 = (\prod_{i=1}^{n} h_i^{ID_i} h)^s$.

- Choose a set $\overline{S}$ of $n-t$ dummy players, such that $\overline{S} \bigcap S = \phi$. For each user $ID_i' \in \overline{S}$, compute $T_i' = \prod_{ID_i \in S} T_i^{\lambda_{ij}}$ and $K_i = \frac{1}{e(T_i', g_2^s)}$, where λ_{ij} denotes the Lagrange coefficients.
- The ciphertexts are $(C_1, C_2, C_3, \{K_i\}_{ID_i' \in \overline{S}})$.

Note: $K_i = \frac{1}{e(T_i', g_2^s)} = \frac{1}{e(g^{t_i'}, g_2^s)}$ by using Lagrange interpolation where $t_i' = f(ID_i')$.

***Partial Decryption*:** Given the ciphertexts $(C_1, C_2, C_3, \{K_i\}_{ID_i' \in \overline{S}})$, the receiver $ID_i \in S$ with his corresponding private d_{ID_i} computes as follows:

$$K_i = \frac{e(C_3, d_{i0}')}{e(d_{i0} \prod_{j=1, j \neq i}^{n} d_{ij}^{ID_j}, C_1)} = \frac{1}{e(g^{t_i}, g_2)^s} .$$

***Decryption*:** Given the valid ciphertexts $(C_1, C_2, C_3, \{K_i\}_{ID_i' \in \overline{S}})$, a subset $S_1 \subset S$ with $|S_1| = t$ and corresponding t partial decryptions, the algorithm computes with the whole set $S' = S_1 \bigcup \overline{S}$ as follows:

$$K = \prod_{ID_i \in S'} K_i^{\lambda_{i0}} = \frac{1}{e(g_1, g_2)^s} .$$

and $M = K \cdot C_2$.

3.2 The Modified Construction

Our basic construction has short ciphertexts in the standard model, but the size of the private achieves |*S*|-size. In this section, we modify it such that the size of the private achieves constant.

Let S = { $ID_1, \cdots, ID_n$ } be *n* users, where $ID_i \in Z_p$. These users want to form an ad hoc network.

***Setup*:** To generate the system parameters, the PKG picks randomly generators { $g, g_2, h, h_1, \cdots, h_n$ } in G and an element α 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 $T_i = g^{t_i}$ for $1 \le i \le n$ and $g_1 = g^{\alpha}$. The public parameters *PK* are

$$PK = (g, g_1, g_2, T_1, \cdots, T_n, h, h_1, \cdots, h_n)$$

and the master key is α .

***Extract(ID$_i$)*:** To generate a private key for a user $ID_i \in Z_p$, the PKG generates a random $r_i \in Z_p$, and outputs the private key:

$$d_{ID_i} = (d_{i0}, d_{i1})$$

$$= (g_2^{t_i} (\prod\nolimits_{i=1}^{n} h_i^{ID_i} h)^{r_i}, g^{r_i}).$$

***Threshold Encryption*:** To encrypt a message M for a set $S = \{ID_1, \cdots, ID_n\}$ of n players, with threshold $t \le n$ for the decryption, the idea is to set up an (n, N)-threshold secret sharing scheme, where $N = 2n - t$. The n public keys ($T_1, \cdots, T_n$) 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_1).Then a sender acts as follows:

- Select a random element $s \in Z_p^*$ and compute $C_1 = g^s$, $C_2 = e(g_1, g_2)^s M$ and $C_3 = (\prod_{i=1}^{n} h_i^{ID_i} h)^s$.
- Choose a set $\overline{S}$ of $n-t$ dummy players, such that $\overline{S} \cap S = \phi$. For each user $ID_i' \in \overline{S}$, compute $T_i' = \prod_{ID_i \in S} T_i^{\lambda_{ij}}$ and $K_i = \frac{1}{e(T_i', g_2^s)}$, where λ_{ij} denotes the Lagrange coefficients.
- The ciphertexts are $(C_1, C_2, C_3, \{K_i\}_{ID_i' \in \overline{S}})$.

Note: $K_i = \frac{1}{e(T_i', g_2^s)} = \frac{1}{e(g^{t_i'}, g_2^s)}$ by using Lagrange interpolation where $t_i' = f(ID_i')$.

***Partial Decryption*:** Given the ciphertexts ($(C_1, C_2, C_3, \{K_i\}_{ID_i' \in \overline{S}})$), the receiver $ID_i \in S$ with his corresponding private d_{ID_i} computes as follows:

$$K_i = \frac{e(C_3, d_{i1})}{e(d_{i0}, C_1)} = \frac{1}{e(g^{t_i}, g_2)^s}.$$

***Decryption*:** Given the valid ciphertexts $(C_1, C_2, C_3, \{K_i\}_{ID_i' \in \overline{S}})$, a subset $S_1 \subset S$ with $|S_1| = t$ and corresponding t partial decryptions , the algorithm computes with the whole set $S' = S_1 \cup \overline{S}$ as follows:

$$K = \prod_{ID_i \in S'} K_i^{\lambda_{i0}} = \frac{1}{e(g_1, g_2)^s}.$$

and $M = K \cdot C_2$.

Security analysis is similar to our basic scheme. So we omit it.

3.3 Efficiency

In our schemes, the size of ciphertexts is approximately $n-t$. They are constructed in the standard model for the identity-based threshold broadcast encryption. In addition, if the values $e(g_1, g_2)$ and $e(T_i, g_2)$ can be precomputed and cached, so no pairing computations are needed at the phase of *Threshold Encryption*. **Table 1-3** give the comparisons between our schemes and the others IDTHBEs.

Table 1. Comparison I of the Efficiency with the others IDTHBE

Schemes	**Hardness Assumption**	**Ciphertext Size**	**Private Key Size**	**Without Relying on R.O.**
[10]	BDH	$n+2$	1	NO
[11]	BDH	$n-t+4$	1	NO
[12]	(l,m,t)-MSEDDH	2	2	NO
Our 1st scheme	BDHE	$n-t+3$	$n+1$	YES
Our 2nd scheme	BDHE	$n-t+3$	2	YES

Note: En. and De. denote the phase of Encryption and Decryption respectively. R.O. denotes the random oracles.

Table 2. Comparison II of the Computation Efficiency with the others IDTHBE

Schemes	**Parings**		**PM**		**M**		**Ex**		**Mi**	
	En.	**De.**	**En.**	**De.**	**En.**	**De.**	**En.**	**De.**	**En.**	**De.**
[10]	0	$2t$	$2n+1$	t	$n+t$	0	1	t	0	0
[11]	$3t$	$2t$	$2n+8$	$3n+t$	0	0	1	n	0	0
[12]	0	$t+1$	0	1	$m+t$	$2(m+t)$	$m+1$	2	1	$(\approx)t^2+t+2$
Our 1st scheme	0	2	$n+1$	n	n	n	$n+3$	$2n$	n-t	0
Our 2nd scheme	0	2	$n+1$	n	n	0	$n+3$	n	n-t	0

In Table 2, PM denotes point multiplications operations, M multiplications operations, Ex exponentiation operations, Mi modular inverse. In addition, m denotes the maximal size of an authorized set and $n \le m$. It is worth noting that the computation cost is mainly denoted by pairing operations since one pairing operation is about 11110 multiplications (We assume that all schemes are all using the GDH group derived from the curve $E/F_{3^{163}}$ defined by the equation $y^2 = x^3 - x + 1$). Hence our schemes are more efficient in terms of computation cost than others. In fact, let (m,n,t)=(200,100,50) and obtain a concrete computation cost (see Table 3). We would also like to point out that in a mobile ad hoc networks, it is not common to have a very large group.

Table 3. Comparison III of the Computation Efficiency with the others IDTHBE((m,n,t)=(200,100,50))

Schemes	[10]	[11]	[12]	Our 1st scheme	Our 2nd scheme
Computation cost	$(\approx)87\ s$	$(\approx)217.5s$	$(\approx)44.37\ s$	$(\approx)1.194s$	$(\approx)1.194s$

In Table 3, s denots cputime (*Second*). In addition, we only consider the computation cost of the *En.* and *De.* phases. All experiments are run on a personal computer with Pentium Dual core E6500 (2.94 GHz) and a maximum of 2.0 GB of the memory available. The program of the algorithms is written in Matlab 7.1 language.

4. Security Analysis

This section is devoted to the proof of the IND-sID-CPA security for our construction.

Theorem 1: Suppose the decision BDHE assumption holds. Then the proposed scheme above is semantically secure against selective identity, chosen plaintext attacks (IND-sID-CPA).

Proof: Suppose an adversary A has advantage ε in attacking our scheme. Using A, we build an algorithm B that solves the decision BDHE problem in G with the advantage ε. For a generator $g \in G$ and $\alpha \in Z_p$, set $y_i = g^{\alpha^i} \in G$. Algorithm B is given as input a random tuple (g, h_0, $y_1, \cdots, y_n, y_{n+2}, \cdots, y_{2n+2}$, T). Algorithm B's goal is to output 1 when $T = e(g,h_0)^{\alpha^{n+1}}$ and 0 otherwise. Algorithm B works by interacting with A in a threshold selective-identity game as follows:

Init: A outputs a set $S^* = (ID_1^*, \cdots, ID_n^*)$ of identities that it wants to attack, and a set $\tilde{S}$ of identities that it wants to corrupt, with $|S^*| \le n-1$ and $|\tilde{S} \bigcap S^*| \le t-1$.

Setup: B does the following:

- First, B selects $n-1$ random integers $\alpha_1, \alpha_2, \cdots, \alpha_{n-1} \in Z_p$. 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 α. For $ID_i \in \tilde{S}$, B computes $T_i = g^{\alpha_i}$. Otherwise, B computes

$$\alpha_i = f(ID_i) = \lambda_0 \alpha + \sum_{j=1}^{n-1} \lambda_j \alpha_j$$

with the Lagrange coefficients λ_j. Note that these Lagrange coefficients are easily calculated since they do not depend on f. Then B sets

$$T_i = g_1^{\lambda_0} \prod\nolimits_{ID_j \in \tilde{S}} T_j^{\lambda_j} \ .$$

- Next, B picks randomly $\gamma \in Z_p^*$ and sets $g_1 = y_1 = g^{\alpha}$, $g_2 = y_n g^{\gamma} = g^{\gamma+\alpha^n}$. Then B selects randomly $\gamma_1, \gamma_2, \cdots, \gamma_n$ in Z_p^* and sets $h_i = g^{\gamma_i} / y_{n-i+1}$ for $1 \le i \le n$. In addition, B selects randomly $\upsilon \in Z_p^*$ and sets $h = g^{\upsilon} \prod\nolimits_{i=1}^{n} y_{n-i+1}^{ID_i^*}$.
- Finally, B gives the public keys PK=($g, g_1, g_2, h, T_1, \cdots, T_n, h_1, \cdots, h_n$) to A.

***Query phase* 1:** A issues up to q_s private key generation queries to the uncorrupt servers. Each query q_i works as follows: Suppose A asks for the private key corresponding to an identity $ID_i^* \notin S^*$. The restriction ensures that $ID - ID_i^* \neq 0$. B first computes the Lagrange coefficients $\lambda_0, \lambda_1, \cdots, \lambda_{n-1}$ such that $t_i = f(ID_i) = \lambda_0 \alpha + \sum_{j=1}^{n-1} \lambda_j \alpha_j$. Then B selects a random $r \in Z_p$ and computes the corresponding private key $d_{ID_i} = (d_{i0}, d'_{i0}, d_{i1}, \cdots, d_{i(i-1)}, d_{i(i+1)}, \cdots, d_{in})$ as follows:

$$(g_2^{t_i} (h_i^{ID_i} h)^{\tilde{r}_i}, g^{\tilde{r}_i}, h_1^{\tilde{r}_i}, \cdots, h_{i-1}^{\tilde{r}_i}, h_{i+1}^{\tilde{r}_i} \cdots, h_n^{\tilde{r}_i}),$$

where $\tilde{r}_i = r + \dfrac{\lambda_0 \alpha^i}{ID_i - ID_i^*}$. In fact,

$$
\begin{aligned}
(h_i^{ID_i} h)^{\tilde{r}_i} &= ((g^{\gamma_i} / y_{n-i+1})^{ID_i} g^{\upsilon} \prod\nolimits_{j=1}^{n} y_{n-j+1}^{ID_j^*})^{\tilde{r}_i} \\
&= (g^{\gamma_i ID_i + \upsilon} y_{n-i+1}^{ID_i - ID_i^*} g^{\upsilon} \prod\nolimits_{j=1, j\neq i}^{n} y_{n-j+1}^{ID_j^*})^{\tilde{r}_i} \\
&= (y_{n-i+1}^{ID_i - ID_i^*})^{\tilde{r}_i} (g^{\gamma_i ID_i + \upsilon} \prod\nolimits_{j=1, j\neq i}^{n} y_{n-j+1}^{ID_j^*})^{\tilde{r}_i} ,
\end{aligned}
$$

where

$$
(y_{n-i+1}^{ID_i - ID_i^*})^{\tilde{r}_i} = (y_{n-i+1}^{ID_i - ID_i^*})^{r} (y_{n-i+1}^{ID_i - ID_i^*})^{\frac{\lambda_0 \alpha^i}{ID_i - ID_i^*}} = (y_{n-i+1}^{ID_i - ID_i^*})^{r} \Big/ y_{n+1}^{\lambda_0} .
$$

Therefore, one can obtain

$$
(h_i^{ID_i} h)^{\tilde{r}_i} = (g^{\gamma_i ID_i + \upsilon} \prod\nolimits_{j=1, j\neq i}^{n} y_{n-j+1}^{ID_j^*})^{\tilde{r}_i} (y_{n-i+1}^{ID_i - ID_i^*})^{r} \Big/ y_{n+1}^{\lambda_0} . \qquad (1)
$$

Then we have

$$
\begin{aligned}
g_2^{t_i} (h_i^{ID_i} h)^{\tilde{r}_i} &= g_2^{\lambda_0 \alpha + \sum_{j=1}^{n-1} \lambda_j \alpha_j} (h_i^{ID_i} h)^{\tilde{r}_i} = g_2^{\lambda_0 \alpha} g_2^{\sum_{j=1}^{n-1} \lambda_j \alpha_j} (h_i^{ID_i} h)^{\tilde{r}_i} \\
&= (y_n g^{\gamma})^{\lambda_0 \alpha} g_2^{\sum_{j=1}^{n-1} \lambda_j \alpha_j} (h_i^{ID_i} h)^{\tilde{r}_i} = g^{\lambda_0 \alpha^{n+1}} g_1^{\lambda_0 \gamma} g_2^{\sum_{j=1}^{n-1} \lambda_j \alpha_j} (h_i^{ID_i} h)^{\tilde{r}_i} \\
&= y_{n+1}^{\lambda_0} g_1^{\lambda_0 \gamma} g_2^{\sum_{j=1}^{n-1} \lambda_j \alpha_j} (h_i^{ID_i} h)^{\tilde{r}_i} ,
\end{aligned}
$$

According (1), we have

$$
\begin{aligned}
g_2^{t_i} (h_i^{ID_i} h)^{\tilde{r}_i} &= y_{n+1}^{\lambda_0} g_1^{\lambda_0 \gamma} g_2^{\sum_{j=1}^{n-1} \lambda_j \alpha_j} (g^{\gamma_i ID_i + \upsilon} \prod\nolimits_{j=1, j\neq i}^{n} y_{n-j+1}^{ID_j^*})^{\tilde{r}_i} (y_{n-i+1}^{ID_i - ID_i^*})^{r} \Big/ y_{n+1}^{\lambda_0} \\
&= g_1^{\lambda_0 \gamma} g_2^{\sum_{j=1}^{n-1} \lambda_j \alpha_j} (g^{\gamma_i ID_i + \upsilon} \prod\nolimits_{j=1, j\neq i}^{n} y_{n-j+1}^{ID_j^*})^{\tilde{r}_i} (y_{n-i+1}^{ID_i - ID_i^*})^{r} .
\end{aligned}
$$

Note that all the terms in this expression are known to *B*. Hence, *B* can compute the first private key component. In addition, *B* computes

$$
g^r y_i^{\frac{\lambda_0}{ID_i - ID_i^*}} = g^r g^{\frac{\lambda_0 \alpha^i}{ID_i - ID_i^*}} = g^{\tilde{r}_i} .
$$

Then *B* can obtain the second private key component. Similarly, the remaining elements are obtained. So $d_{ID_i} = (d_{i0}, d'_{i0}, d_{i1}, \cdots, d_{i(i-1)}, d_{i(i+1)}, \cdots, d_{in})$ is a valid respondence to *A*.

Challenge**:** *A* outputs two same-length messages M_0 and M_1 on which it wishes to be challenged. *B* picks a random $b \in \{0,1\}$ and constructs the challenge ciphertexts as follows:

$$
\begin{aligned}
C^* &= (C_1^*, C_2^*, C_3^*, \{K_i\}) \\
&= (h_0, M_b e(g_1, h_0^{\gamma}) T, h_0^{\upsilon + \sum_{i=1}^{n} ID_i^* \gamma_i}, \{K_i\}_{ID_i \in S_0})
\end{aligned}
$$

where S_0 is a set of *n - t* dummy users. In addition, K_i is computed in the following manner.

B first chooses a set S_0 of $n-t$ dummy users such that $S_0 \cap S^* = \phi$. For each dummy user $ID_i \in S_0$, *B* computes the Lagrange coefficients λ_{ji} with $1 \leq j \leq n$ such that $t'_j = f(ID_j) = \sum_{ID_i^* \in S^*} \lambda_{ji} \alpha'_i$, where α'_i is known to *B* since *B* can compute it by using $(\alpha_1, \alpha_2, \cdots, \alpha_{n-1})$ and satisfies $g^{\alpha'_i} = T_i$. Then *B* computes $T'_j = \prod_{ID_i^* \in S^*} T_i^{\lambda_{ji}}$. Finally, *B* computes

$$
K_i = \frac{1}{e(h_0^{\sum_{i=1}^{n} \lambda_{ji} \alpha'_i}, g_2)} .
$$

Let $h_0 = g^{\mu}$ for some unknown $\mu \in Z_p$. If $T = e(g, h_0)^{\alpha^{n+1}}$, one can obtain that C^* is a valid encryption for M_b. In fact,

$$\begin{aligned}
C_1^* &= g^{\mu},\\
C_2^* &= M_b e(g_1, h_0^{\gamma})T = M_b e(g_1, h_0^{\gamma})\ e(g, h_0)^{\alpha^{n+1}}\\
&= M_b e(g_1, h_0^{\gamma})\ e(g^{\alpha^{n+1}}, h_0)\ = M_b(e(g_1, g^{\gamma})e(g^{\alpha^n}, g_1))^{\mu}\\
&= M_b e(g_1, g^{\gamma} y_n)^{\mu} = M_b e(g_1, g_2)^{\mu}.\\
C_3^* &= h_0^{\upsilon+\sum_{i=1}^{n} ID_i^* \gamma_i} = (g^{\upsilon+\sum_{i=1}^{n} ID_i^* \gamma_i})^{\mu} = (\prod_{j=1}^{n} g^{ID_j^* \gamma_j} g^{\upsilon})^{\mu}\\
&= (\prod_{j=1}^{n} (g^{\gamma_j} / y_{n-j+1})^{ID_j^*} g^{\upsilon} \prod_{j=1}^{n} (y_{n-j+1})^{ID_j^*})^{\mu} = (\prod_{i=1}^{n} h_i^{ID_i} h)^{\mu}.
\end{aligned}$$

and

$$K_i = \frac{1}{e(h_0^{\Sigma_{i=1}^{n} \lambda_{ji} \alpha_i'}, g_2)} = \frac{1}{e(g^{\Sigma_{i=1}^{n} \lambda_{ji} \alpha_i'}, g_2)^{\mu}} = \frac{1}{e(g^{\Sigma_{i=1}^{n} \lambda_{ji} \alpha_i'}, g_2)^{\mu}} = \frac{1}{e(\prod_{i=1}^{n} T_i^{\lambda_{ji}}, g_2)^{\mu}} = \frac{1}{e(T_i', g_2)^{\mu}}.$$

If T is a random element of G_1, C^* gives no information about B's choice of b.

***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 DBDHE problem, otherwise it outputs 0 to mean that it learns nothing from C^*.

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

$$| Pr(B(TU, e(g, h_0)^{\alpha^{n+1}}) = 0) - Pr(B(TU, T) = 0) | \geq | \tfrac{1}{2} \pm \varepsilon - \tfrac{1}{2} | = \varepsilon.$$

5. Conclusions

We propose two new constructions of identity-based threshold broadcast encryption in the standard model. In our schemes, the broadcaster can dynamically choose the set of *n* recipients and the threshold value *t*. Both schemes have short ciphertexts, where the length of ciphertexts achieves *n-t*. In addition, we reduce their security to the decision bilinear Diffie-Hellman Exponent (BDHE) problem under the selective-identity security model.

Unfortunately, our scheme only achieves the selective-identity security. In addition, 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.

References

[1] A. Fiat, M. Naor, “Broadcast encryption,” in *Proc. of CRYPTO*, Springer-Verlag, LNCS 773, pp. 480-491, 1994.

[2] H. Ghodosi, J. Pieprzyk and R. Safavi-Naini, “Dynamic threshold cryptosystems: a new scheme in group oriented cryptography,” in *Proc. of Pragocrypt 96*, CTU Publishing House, pp. 370-379, 1996.

[3] A. Shamir, “Identity-based Cryptosystems and Signature Schemes,” in *Proc. of CRYPTO*,

Springer-Verlag, LNCS 196, pp. 47–53, 1984.

[4] D. Boneh and M. Franklin, “Identity-based encryption from the well pairing,” in *Proc. of CRYPTO*, Springer-Verlag, LNCS 2193, pp. 213-229, 2001.

[5] D. Boneh and X. Boyen, “Efficient selective-id secure identity based encryption without random oracles,” in *Proc. of EuroCryp*, Springer-Verlag, LNCS 3027, pp. 223-238, 2004,.

[6] C. Cocks, “An identity based encryption scheme based on quadratic residues,” in *Proc. of Cryptography and coding,* Springer-Verlag, LNCS 2260, pp. 360-363, 2001.

[7] D. Boneh and J. Katz, “Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption,” in *Proc. of CT-RSA*, Springer-Verlag, LNCS 3376, pp. 87-103, 2005.

[8] R. Canetti, S. Halevi, and J. Katz, “Chosen-ciphertext security from identity-based encryption,” in *Proc. of EuroCrypt*, Springer-Verlag, LNCS 3027, pp. 207-222, 2004.

[9] S. Chattterjee and P. Sarkar, “Generalization of the Selectve-ID Security Model for HIBE Protocols,” in *Proc. of PKC*, Springer-Verlag, LNCS 3958, pp. 241-256, 2006.

[10] Z. Chai, Z. Cao and Y. Zhou, “Efficient ID-based Broadcast Threshold Decryption in Ad Hoc Network,” in *Proc. of IMSCCS 06*, IEEE Computer Society, Volume 2, pp. 148-154, 2006.

[11] V. Daza, J. Herranz and P. Morillo, “CCA2-Secure Threshold Broadcast Encryption with Shorter Ciphertexts,” in *Proc. of ProvSec 2007*, Springer-Verlag, LNCS 4784, pp. 35-50, 2007.

[12] C. Delerablee and D. Pointcheval, “Dynamic Threshold Public-Key Encryption,” in *Proc. of CRYPTO*, Springer-Verlag, LNCS 5157, pp. 317-334, 2008.

Leyou Zhang was born in 1977. He received the Ph. D degrees in applied mathematics form Xidian University of China in 2009. Now he is an associate professor in Department of Mathematical Sciences of Xidian University. His main research interests include secure protocol and public key cryptography

Yupu Hu was born in 1955. He received the Ph. D degrees in School of Telecommunication Engineering form Xidian University of China in 1999. Now he is a professor in School of Telecommunication Engineering of Xidian University in Xidian University. His main research interests include information security and cryptography.

Qing Wu was born in 1975. She received the Ph. D degrees in applied mathematics form Xidian University of China in 2009. Now she is an associate professor in School of Automation, Xi’an Institute of Posts and Telecommunications. Her main research interest is applied mathematics.