# Efficient Anonymous Broadcast Encryption with Adaptive Security

**Fu-Cai Zhou[1], Mu-Qing Lin[2, 3], Yang Zhou[1] and Yu-Xi Li[1]**
[1] Software College, Northeastern University
Shenyang, 110819, China
[e-mail: fczhou@mail.neu.edu.cn, zhouyang921@gmail.com, eliyuxi@gmail.com]
[2] College of Information Science and Engineering, Northeastern University
Shenyang, 110819, China
[e-mail: linmuqing@gmail.com]
[3] Information Security Institute, Beijing Electronic Science and Technology Institute
Beijing, 100070, China
[e-mail: linmq@besti.edu.cn]
*Corresponding author: Fu-Cai Zhou

*Received October 23, 2014; revised August 15, 2015; accepted September 2, 2015; published November 30, 2015*

## *Abstract*

Broadcast encryption is an efficient way to distribute confidential information to a set of receivers using broadcast channel. It allows the broadcaster to dynamically choose the receiver set during each encryption. However, most broadcast encryption schemes in the literature haven't taken into consideration the receiver's privacy protection, and the scanty privacy preserving solutions are often less efficient, which are not suitable for practical scenarios. In this paper, we propose an efficient dynamic anonymous broadcast encryption scheme that has the shortest ciphertext length. The scheme is constructed over the composite order bilinear groups, and adopts the Lagrange interpolation polynomial to hide the receivers' identities, which yields efficient decryption algorithm. Security proofs show that, the proposed scheme is both secure and anonymous under the threat of adaptive adversaries in standard model.

***Keywords:*** Anonymous broadcast encryption, privacy protection, Lagrange interpolation, adaptive security

## 1. Introduction

Broadcast encryption (BE) [1] is a cryptographic primitive that allows a broadcaster to encrypt a message for a dynamic set of users and use a broadcast channel to distribute the ciphertext. Only the users within the set can use their private key to decrypt the message, users outside the set can obtain no confidential information about the encryption, even if all of these users collude. The receiver set is dynamic, i.e., it is selected by the broadcaster at the time of encryption, not at the time of system initialization. This means the broadcaster can select an arbitrary set of users to receive the message during each encryption. This features gives broadcast encryption great flexibility compares to those pre-shared group key communication schemes. Broadcast encryption has many practical applications [21-23], such as the access control in encrypted file systems, digital right management systems for satellite TV and DVD content protection, and secure group communications.

There are two kinds of broadcast encryption, the symmetric key based [1-5] and the public key based [6-9]. In a symmetric key based broadcast encryption system, only the trusted authority that generates all private keys can act as the broadcaster; ordinary users can only receive and decrypt messages, but cannot send messages. However, in a public key based system, anyone who knows the public key can broadcast messages to the receiver set he selects.

Identity-based broadcast encryption (IBBE) [10-12] is a special kind of public key broadcast encryption that it combines identity-based encryption (IBE) with broadcast encryption. In an identity-based broadcast encryption system, the encryption and the decryption are based on receivers' identities, in which the users in a normal broadcast encryption are usually indexed sequentially from 1 to $n$. Therefore, the most important difference between broadcast encryption and identity-based broadcast encryption is the number of users in the system. In a normal broadcast encryption system, the size $n$ of the user's universal set should be determined during the system initialization. When adding a new user, the system's public key should be updated to include the new user's information. However, the identity-based broadcast encryption doesn't need to determine the size of user set, so it can support exponentially many users since users' identities are merely bit strings of arbitrary length.

The prior works on broadcast encryption have mainly focused on enhancing the system's efficiency and security properties. Comparatively little attention has been paid to user's privacy protection. In a standard broadcast encryption system, the receiver set is usually transmitted along with the ciphertext. User can not only examine whether he belongs to the set, but also learn other users' identities that belong to the same set. For example, in the scheme of Boneh et al. [7], even the one that outside the system can learn the target set, because the decryption algorithm needs the information of the whole set as input to decrypt messages. This is inapplicable for those privacy-sensitive scenarios, since the identities of the users that in the receiver set are often as sensitive as the encrypted content itself.

In a privacy preserving broadcast encryption, user should only able to examine whether himself is in the set, but cannot find any other users' identities. The first work that considers the privacy problem in broadcast encryption is done by Barth et al. [13]. They introduced the notion of private broadcast encryption scheme, explicitly aimed to protect the identities of the receivers. Barth et al.'s work has subsequently attracted researchers' attentions to construct more privacy-preserving broadcast encryption schemes [14-16]. However, all existing

schemes, no matter identity-based or not, are inefficient in decrypting ciphertexts. In their schemes, the techniques they adopted to hide the receiver's identity into the ciphertext require the decryption algorithm to try to find the right part in the ciphertext to decrypt the message. That is, they need multiple times decryption attempts to compute the right output. In addition, the security models of the previous solutions, especially the identity-based ones, are all against selective adversaries, which still have room for further improvement.

In this paper, we construct a novel dynamic anonymous identity-based broadcast encryption scheme. In the scheme, everyone can receive the broadcasted ciphertext, but only the receivers selected by the broadcaster can successfully decrypt the message. Besides, one can examine whether himself is in the receiver set, but no one except the broadcaster knows who the other receivers are. Different from those previous constructions, we use Lagrange interpolation polynomial to hide the receivers' identities. Lagrange interpolation theorem can not only be used in secret sharing and traitor tracing, but also very suitable for identity hiding and restoring, which can be used to construct efficient decryption algorithm. Since the scheme is identity-based, it allows user identity to be arbitrary length bit-string, and supports dynamic joining after the initialization. The scheme is built over the composite order bilinear groups, and is secure and anonymous against adaptive adversaries under the composite decisional Bilinear Diffie- Hellman assumption and the subgroup decision assumption. The main contributions of this paper are:

1. We proposed an efficient dynamic anonymous broadcast encryption scheme. Compared to those existing constructions, the proposed scheme adopts the Lagrange interpolation polynomial to hide the receivers' identities, which result in a more efficient decryption algorithm.

2. We defined a more rigorous security model to characterize the security threats, and gave the formal mathematical proofs under standard model to claim that our scheme is both secure and anonymous against the adaptive chosen ciphertext attack.

The remainder of this paper is arranged as follows. Section 2 lists a few works in the literature that relate to our topic; section 3 introduces some preliminaries about the backgrounds and complexity assumptions; section 4 gives the formal definitions of the system; section 5 gives the construction of the proposed scheme; the formal security proofs and performance evaluations are given in section 6, and the whole paper is concluded in section 7.

## 2. Related Work

The first work that formally explores the broadcast encryption was done by Fiat el al. [1]. The solution they proposed is secure against a collusion of at most $t$ users and has ciphertext size of $O(t\log^2 t\log n)$, where $n$ is the total number of users. After that, several full collusion resistant broadcast encryption schemes [2, 3, 6] have been proposed to achieve shorter ciphertext and private keys.

In CRYPTO 2005, Boneh et al. [7] proposed a fully collusion resistant broadcast encryption with constant size ciphertexts and private keys. Their construction is based on the symmetric bilinear maps and the bilinear Diffie-Hellman exponent (BDHE) assumption, and is secure against chosen plaintext attacks (CPA). They also adopted the strongly existentially unforgeable signature scheme to construct a chosen ciphertext attack (CCA) secure scheme. However, both schemes are selectively secure. The work done by Gentry et al. [17] is the first that achieves adaptive CPA security. They proposed two broadcast encryption schemes and two identity-based broadcast encryption schemes; each has constant ciphertext size in random

oracle model. Phan et al. [9] modified the first scheme in [7] and proposed a selectively CCA secure scheme with constant size ciphertexts and private keys. They then managed to prove that their scheme is adaptively CCA secure under generalized versions of the BDHE assumption and the knowledge-of-exponent assumption (KEA).

Identity-based broadcast encryption was first studied by Delerablée [10]. They proposed a selectively CPA secure identity-based scheme which has constant size ciphertexts and private keys. Since their solution is identity-based, it can support dynamic user joining without updating the previous keys. Boneh et al. [11] brought the concept of hierarchical IBE (HIBE) and built a broadcast HIBE scheme in standard model. Zhang et al. [12] presented an IBBE scheme using dual encryption technique that achieves adaptive security in standard model using static assumptions.

The above researches on broadcast encryption haven't taken into consideration the user's privacy since the decryptions often need the receiver set as the explicit input. The concept of private broadcast encryption was introduced by Barth et al. in [13]. They built two public key based constructions that do not leak any information about the receiver set. Fazio et al. [14] proposed a broadcast encryption scheme with sublinear ciphertext size that achieves receiver anonymity. The receiver's identity in their scheme is anonymous to the outsider, but not to the ones that in the same receiver set. Libert et al. [15] claimed that the anonymity notion in [14] is weak and is not suit for real-world applications. They gave a formal security definition for anonymous broadcast encryption (ANOBE), which allows the adversary to make adaptive corruptions. The solution they proposed is based on the Kurosawa-Desmedt hybrid encryption scheme [18]. However, this solution uses multiple copies of symmetric ciphertexts as the broadcast body to achieve anonymity. Fan et al. [19] proposed an anonymous multi-receiver identity-based encryption scheme that adopts Lagrange interpolating polynomial mechanism to hide user's identity. The scheme is only selectively secure in the random oracle model. Hur et al. [16] constructed a privacy-preserving identity-based broadcast encryption scheme, where the ciphertext size is linear in the number of receivers. The scheme is also selectively secure, and needs multiple times decryption attempts to decrypt the ciphertext.

## 3. Preliminaries

We begin by briefly introducing the basic idea of public key broadcast encryption systems. Then we state some preliminaries needed for our construction.

### 3.1 Public Key Broadcast Encryption System

A public key broadcast encryption system [7] can be seen as a key encapsulation mechanism. It is made up of the following three algorithms.

- $Setup(n)$. A probabilistic algorithm that takes as input the total number of users $n$, outputs $n$ private keys $d_1,\ldots,d_n$ and a public key $PK$.
- $Encrypt(PK,S)$. A probabilistic algorithm that takes as input the public key $PK$ and a subset $S\in\{1,\ldots,n\}$, outputs a pair $(Hdr,K)$. The $Hdr$ is called the header, and the $K$ is a temporary session key that is for encrypting the message $M$ using standard symmetric encryption. Let $C_M$ be the ciphertext of $M$, then the broadcast consists of $(S,Hdr,C_M)$. The pair $(S,Hdr)$ is usually called the full header, and $C_M$ is usually called the broadcast body.
- $Decrypt(PK,S,i,d_i,Hdr)$. A deterministic algorithm that takes as input the public key $PK$, a target set $S\in\{1,\ldots,n\}$, a user index $i\in\{1,\ldots,n\}$, the private key $d_i$ of user $i$, and a header

$Hdr$. If $i \in S$, then the algorithm outputs the session key $K$. User $i$ can use this $K$ to decrypt $C_M$ and recover the message $M$.

This public key broadcast encryption system is correct if for all subsets $S \subseteq \{1,\ldots,n\}$ and all $i \in S$,

$$\Pr[(PK, d_1, \ldots, d_n) \leftarrow Setup(n); (Hdr, K) \leftarrow Encrypt(PK, S); K' \leftarrow Decrypt(PK, S, i, d_i, Hdr) : K = K'] = 1$$

## 3.2 Composite Order Bilinear Groups

The definition of composite order bilinear groups was first introduced in [20]. Given the security parameter $k$, an algorithm $\mathsf{G}$ generates a tuple $(p, q, G, G_T, e)$, where $p$ and $q$ are distinct primes, $G$ and $G_T$ are cyclic groups of order $n = pq$, and $e: G \times G \to G_T$ is a bilinear map such that:

1. (Bilinearity) $e(g^a, h^b) = e(g, h)^{ab}$ for all $g, h \in G, a, b \in \mathbb{Z}_n$.

2. (Non-degeneracy) $\exists g \in G$ such that $e(g, g) \neq 1$ and is a generator in $G_T$.

3. (Computability) The group operations in $G, G_T$ as well as the map $e: G \times G \to G_T$ can be computed in polynomial time with respect to $k$.

Let $G_p$ and $G_q$ be the subgroups of order $p$ and $q$ respectively, $g$ be a generator of group $G$, and $g_p, g_q$ be the generators of $G_p$ and $G_q$ respectively. Then for any $h_p \in G_p$ and $h_q \in G_q$, $e(h_p, h_q)$ is the identity element in $G_T$:

$$e(h_p, h_q) = e(g_p^a, g_q^b) = e(g^{qa}, g^{pb}) = e(g^a, g^b)^{pq} = e(g^a, g^b)^n = 1, \quad (1)$$

since the order of group $G_T$ is $n$.

### 3.2.1 The Subgroup Decision Assumption

The subgroup decision problem is defined as follows. Given $(n, G, G_T, e)$ and an element $x \in G$, determine whether the order of $x$ is $p$ or otherwise. In other words, without knowing the factorization of the group order $n$, decide if an element is in a subgroup of $G$.

An algorithm's advantage in solving the subgroup decision problem can be defined as the probability $\left|\Pr[solved] - 1/2\right|$. The subgroup decision assumption (SDA) holds if for any probabilistic polynomial time (PPT) algorithm $\mathsf{A}$, the advantage in solving the subgroup decision problem in $(n, G, G_T, e)$ is at most negligible.

### 3.2.2 The Composite Decisional Bilinear Diffie-Hellman Assumption

The composite Decisional Bilinear Diffie-Hellman (DBDH) problem is defined as follows. Given $(p, q, G, G_T, e)$ and $(g_p, g_q, g_p^\alpha, g_p^\beta, g_p^\gamma, T)$, where $g_p, g_q$ are random generators of $G_p, G_q$ respectively, $\alpha, \beta, \gamma$ are randomly selected from $\mathbb{Z}_n$, $T \in G_T$. Determine whether the element $T$ equals $e(g_p, g_p)^{\alpha\beta\gamma}$, or is a random element in $G_T$.

An algorithm's advantage in solving the composite DBDH problem can be defined as the probability $\left|\Pr[solved] - 1/2\right|$. The composite DBDH assumption holds if for any PPT algorithm $\mathsf{A}$, the advantage in solving the composite DBDH problem in $(p, q, G, G_T, e)$ is at most negligible.

## 3.3 Lagrange Interpolating Polynomial

The Lagrange interpolation method can give a polynomial function that exactly through a

number of known points in the two-dimensional plane.

Given $n$ distinct points $(x_1, y_1),\ldots,(x_n, y_n)$. Let the $n$ monic polynomials of degree $(n-1)$ be denoted by

$$f_i(x) = \prod_{j=1, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}, \qquad i = 1,\ldots,n,$$

such that

$$f_i(x_j) = \delta_{ij} = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases}$$

Then the interpolation polynomial can be given by

$$F(x) = \sum_{i=1}^{n} y_i f_i(x),$$

since

$$F(x_k) = \sum_{i=1}^{n} y_i \delta_{ik} = y_k .$$

## 4. Definitions

We formally define the proposed scheme's system model and security models in this section. There are three types of entities in our system: the private key generator (PKG), the broadcaster and the receiver. The following explains the roles of these entities.

1. PKG. The private key generator is a trusted entity in the system that is responsible for system initialization and private key generating. It generates system's master key and public key, and responds to the joining request to generate private key for each user. We assume the private keys are transferred in secure channel. The PKG doesn't participate in the encryption and decryption, i.e. it could be offline if there are currently no new users to join.

2. Broadcaster. A broadcaster is a user who sends message. In a public key broadcast encryption system, any user can be the broadcaster and send messages to others. During each broadcast, the broadcaster needs to explicitly specify the receiver set, and then encrypts the message for the set and sends it.

3. Receiver. A receiver is a user who belongs to a target set in a broadcast. He can decrypt the ciphertext using his private key. In the meantime, he is not able to discover who else is in this set.

The broadcaster and the receiver are actually the same kind of users. We divide users into these two types mainly based on their behaviors in a broadcast. A user can be one message's broadcaster and at the same time be some other message's receiver.

### 4.1 System Model

The formal system model of our scheme is defined as follows.

**Definition 1.** *A dynamic anonymous broadcast encryption is a tuple of four polynomial-time algorithms* $\text{ANOBE} = (Setup, Join, Encrypt, Decrypt)$ *such that:*

$(MSK, PK) \leftarrow Setup(1^k)$ *: is a probabilistic algorithm run by the PKG that takes as input a security parameter* $k$ *, outputs a master key* $MSK$ *and a public key* $PK$ *.*

$sk_i \leftarrow Join(MSK, ID_i)$ *: is a probabilistic algorithm run by the PKG that takes as input the master key* $MSK$ *and a user's identity* $ID_i$ *, outputs a private key* $sk_i$ *for this identity.*

$(Hdr, K) \leftarrow Encrypt(PK, S)$ *: is a probabilistic algorithm run by the broadcaster that takes as input the public key* $PK$ *and a set of user's id* $S = \{\ldots, ID_i, \ldots\}$*, outputs a header* $Hdr$ *and a session key* $K$*.*

$K \leftarrow Decrypt(ID_i, sk_i, Hdr)$ *: is a deterministic algorithm run by the receiver that takes as input a user identity* $ID_i$*, a private key* $sk_i$ *corresponds to this identity, and a header* $Hdr$*, outputs a session key* $K$*.*

At the beginning, the PKG runs the *Setup* algorithm to generate the master key and the public key. Then for each new user, the PKG runs the *Join* algorithm to generate a private key for this user according to his identity. This step can be seen as the user's registration. New user can join at any time, as long as the PKG is online.

Given a message $M$ to be sent, the broadcaster first chooses a receiver set[1], and then runs the *Encrypt* algorithm to output a header $Hdr$ and a session key $K$. He then uses symmetric encryption to encrypt the message $M$ using the session key $K$ to obtain the ciphertext $C_M$. The final broadcast content is $(Hdr, C_M)$.

After receiving $(Hdr, C_M)$, a user can run *Decrypt* algorithm to try to decrypt the header. If he is the receiver, the algorithm can output the right session key; otherwise, the algorithm outputs some other value that cannot be used to decrypt the $C_M$. We observe that the decryption algorithm no longer requires the set $S$ as input.

The dynamic anonymous broadcast encryption is correct if for all sets $S$ and all $ID_i \in S$,

$$\Pr[(MSK, PK) \leftarrow Setup(1^k); sk_i \leftarrow Join(MSK, ID_i); (Hdr, K) \leftarrow Encrypt(PK, S); K' \leftarrow Decrypt(ID_i, sk_i, Hdr) : K = K'] = 1$$

## 4.2 Security Model

Intuitively, a secure anonymous broadcast encryption scheme should meet the following properties.

1. The scheme should be fully collusion resistant. Given a broadcast content, only the users in the receiver set can decrypt the message. Anyone that outside the set cannot recover anything from the ciphertext, even all these users collude.

2. The receivers' identities should be fully anonymous. Given a broadcast content, anyone should only be able to determine if he is the receiver, but cannot discover any other user's identity in the receiver set.

We define the chosen-ciphertext attack model for an anonymous broadcast encryption system using attack games between a challenger $\mathsf{C}$ and an adversary $\mathsf{A}$. Both challenger and adversary are probabilistic processes that communicate with each other. Compares to the chosen-plaintext attack model, the chosen-ciphertext attack means in a game, the adversary can make additional decryption queries before or after the challenge phase. Our security model is based on those in [15, 16]. We refined them by splitting the definition in [15] into two parts (the ciphertext confidentiality and the receiver anonymity), and allow the adversary in the definition in [16] to be adaptive to characterize the security threats more precisely. The *adaptive* means the adversary could choose the set of identities he wants to attack *after* the first query phase of the game, which gives the adversary more ability to attack than those selective ones.

[1] Actually in this step, the set may include any user, even those haven't joined the system. A user can later request his private key from PKG and then try to decrypt the message.

In our model, the security goal is ciphertext indistinguishability, i.e., in a game, the adversary is unable to efficiently distinguish the real ciphertext from a random value, therefore cannot win the game with non-negligible advantage.

Based on these notions, we define the scheme's ciphertext confidentiality, which indicates that an adversary should not able to distinguish a real ciphertext from a random value, even with the ability to query all other users' private keys. We use the following definition to describe the scheme's indistinguishability of encryptions under the adaptive chosen-ciphertext attack (IND-ADA-CCA1), where the IND is short for *indistinguishability*, ADA is short for *adaptive*, and CCA1 means the first type chosen-ciphertext attack, which the adversary may make decryption queries before the challenge phase.

**Definition 2.** *Given the dynamic ANOBE scheme described in Definition 1, describe* $\mathrm{A}$ *as an adaptive adversary,* $\mathrm{C}$ *as a challenger. Consider the following game:*

***Setup.*** *The challenger* $\mathrm{C}$ *runs* $Setup(1^k)$ *to generate master key* $MSK$ *and pubic key* $PK$*, and gives the* $PK$ *to the adversary* $\mathrm{A}$.

*Phase 1.* $\mathrm{A}$ *can issue at most polynomial times queries to the follow oracles combined in any sequences he chooses:*

- *Joining query: For any* $ID_i$ *that* $\mathrm{A}$ *chosen,* $\mathrm{C}$ *runs* $Join(MSK, ID_i)$ *to get the* $sk_i$ *and returns it to* $\mathrm{A}$.
- *Decryption query: For the* $(Hdr_i, S_i)$ *that* $\mathrm{A}$ *chosen,* $\mathrm{C}$ *first chooses an* $ID \in S_i$*, generates* $sk$ *for this* $ID$*, and then runs* $Decrypt(ID, sk, Hdr_i)$ *to get the* $K_i$ *and returns it to* $\mathrm{A}$.

***Challenge.*** $\mathrm{A}$ *chooses a set* $S^*$ *of identities that he hasn't queried the private keys before in phase 1.* $\mathrm{C}$ *runs* $Encrypt(PK, S^*)$ *to encrypt the set* $S^*$ *and obtain a pair* $(Hdr^*, K^*)$. $\mathrm{C}$ *then chooses a random value* $b \in \{0,1\}$*, and set* $K_b = K^*$, $K_{1-b} = K'$*, where* $K'$ *is a random element in* $G_T$. $\mathrm{C}$ *then gives* $(Hdr^*, K_0, K_1)$ *to* $\mathrm{A}$.

***Phase 2.*** $\mathrm{A}$ *issues additional polynomial times joining queries as in phase 1, with the restriction that he cannot issue queries for* $ID_i \in S^*$.

***Guess.*** *The adversary* $\mathrm{A}$ *outputs its guess* $b'$ *for* $b$ *and wins the game if* $b' = b$.

*Define* $\mathrm{A}$*'s advantage as* $Adv_{\mathbf{A},ANOBE}^{IND-ADA-CCA1}(k) = \left|\Pr[b' = b] - 1/2\right|$*, then the dynamic ANOBE scheme is IND-ADA-CCA1 secure if all probabilistic polynomial time adversaries* $\mathrm{A}$ *have at most negligible advantage in the above game.*

We also define the scheme's receiver anonymity, which requires that, given a ciphertext and two receiver sets, an adversary should not able to distinguish which set is used for the encryption. The following definition describes the scheme's anonymous indistinguishability of encryptions under the adaptive chosen ciphertext attack (ANON-ADA-CCA1), where ANON stands for the *anonymous*.

**Definition 3.** *Given the dynamic ANOBE scheme described in Definition 1, describe* $\mathrm{A}$ *as an adaptive adversary,* $\mathrm{C}$ *as a challenger. Consider the following game:*

***Setup.*** *The challenger* $\mathrm{C}$ *runs* $Setup(1^k)$ *to generate master key* $MSK$ *and pubic key* $PK$*, and gives the* $PK$ *to the adversary* $\mathrm{A}$.

*Phase 1.* $\mathrm{A}$ *can issue at most polynomial times queries to the follow oracles combined in any sequences he chooses:*

- *Joining query: For any* $ID_i$ *that* $\mathrm{A}$ *chosen,* $\mathrm{C}$ *runs* $Join(MSK, ID_i)$ *to get the* $sk_i$ *and returns it to* $\mathrm{A}$.
- *Decryption query: For the* $(Hdr_i, S_i)$ *that* $\mathrm{A}$ *chosen,* $\mathrm{C}$ *first chooses an* $ID \in S_i$*, generates* $sk$ *for this* $ID$*, and then runs* $Decrypt(ID, sk, Hdr_i)$ *to get the* $K_i$ *and returns it to* $\mathrm{A}$.

***Challenge.*** A *chooses two distinct receiver sets* $S_0$ *and* $S_1$ *such that* $|S_0|=|S_1|$, *with the requirement that he hasn't queried the private keys for any* $ID_i \in S_0 \Delta S_1 = (S_0 \setminus S_1) \cup (S_1 \setminus S_0)$ *before.* C *chooses a random value* $b \in \{0,1\}$, *and runs* $Encrypt(PK, S_b)$ *to encrypt the set* $S_b$ *and obtain a pair* $(Hdr_b, K_b)$. C *then gives* $(Hdr_b, K_b)$ *to* A.

***Phase 2.*** A *issues additional polynomial times joining queries as in phase 1, with the restriction that he cannot issue queries for* $ID_i \in S_0 \Delta S_1$.

***Guess.*** *The adversary* A *outputs its guess* $b'$ *for* $b$ *and wins the game if* $b'=b$.

*Define* A*'s advantage as* $Adv_{\mathbf{A},ANOBE}^{ANON-ADA-CCA1}(k) = |\Pr[b'=b] - 1/2|$, *then the dynamic ANOBE scheme is ANON-ADA-CCA1 secure if all probabilistic polynomial time adversaries* A *have at most negligible advantage in the above game.*

## 5. The Construction of Dynamic Anonymous Broadcast Encryption

In this section, we present the construction for the anonymous broadcast encryption that achieves receiver privacy. In the construction, we adopt the Lagrange interpolation polynomial mechanism to form the broadcast header. It allows the legitimate receivers to recover their own polynomial values, and then compute the session key using their private keys. The polynomial curve's smooth nature determines that, once an interpolation polynomial has been formed, the points that generate this polynomial can be hidden naturally.

### 5.1 Main Idea

The main idea of the construction works as follows. In the system, each user in the receiver set is associated with a point $(x, y)$. The $x$ coordinate is user's ID, which is an $l$-length bit string; the $y$ coordinate is an element in the group that related to the user's ID (obviously, this is not the usually point in the two-dimensional plane, since the $x$ coordinate is a large number, and the $y$ coordinate is a group element). The $y$ coordinate is not a fixed value; it changes each time in the encryption due to the randomly chosen exponent and factor.

Let $(x_1,\ldots,x_n)$ be the identities in the receiver set. During an encryption, the broadcaster first selects a random exponent and computes the session key together with each receiver's $y$ coordinate, and then uses the points $((x_1,y_1),\ldots,(x_n,y_n))$ to form a polynomial using Lagrange interpolation:

$$F(x) = \sum_{i=1}^{n} y_i f_i(x).$$

The main purpose is to hide these $y$ values into the polynomial, because only these values can be used for decryption. If one substitute an ID $x_i \in (x_1,\ldots,x_n)$ into $F(x)$, he can get the corresponding $y_i = F(x_i)$. If $x_i \notin (x_1,\ldots,x_n)$, since the coefficients of the polynomial are all discrete elements in the group, the value $F(x_i)$ is unpredictable. This also means, given the $F(x)$, no one can discover the points $((x_1,y_1),\ldots,(x_n,y_n))$ that form the polynomial. The polynomial $F(x)$ is the main component of the broadcast header.

For each user, there is a corresponding relationship among the $y$ value, the private key and the session key. The session key can be extracted through these values.

The receiver's identity is confidential to other users. For example, a malicious user may try the polynomial $F(x)$ with other user's ID. In this case, he may obtain the right $y$ value, but

cannot recover the session key, because he doesn't possess the corresponding private key. Therefore, he cannot distinguish whether the $y$ value he outputs is right, i.e. he cannot determine whether the user he selects is a receiver.

### 5.2 Explicit Construction

The whole system works upon the composite order bilinear groups, which allow the random pairing factor to be eliminated if the pairing's inputs are from both subgroups.

The explicit construction is given as follows. For notation convenience, we use both multiplication and addition to represent the group operations. The addition is used to express the polynomial, since the coefficients of the interpolation polynomial are all group elements.

- $(MSK, PK) \leftarrow Setup(1^k)$:

Let $H: \{0,1\}^* \rightarrow \{0,1\}^l$ be a hash function that maps user's identity to an $l$-length bit string, where $l$ is a polynomially bounded value. We'll use the hash value $ID \in \{0,1\}^l$ to represent user's identity in the following content.

Choose two large primes $p$ and $q$ with respect to the security parameter $k$, and generate bilinear pairing parameters $(n, G, G_T, e)$ where $n = pq$ is the order of group $G$ and $G_T$. Let $G_p$ and $G_q$ be the subgroups of order $p$ and $q$ respectively, and $g_p, g_q$ be the generators of $G_p$ and $G_q$ respectively.

Randomly choose $a \in \mathbf{Z}_n^*$, $g_1 \in G_p$, and $\mathbf{U} = (u_0, u_1, \ldots, u_l) \in G_p^{l+1}$. Randomly choose $R \in G_q$, and $(R_0, R_1, \ldots, R_l) \in G_q^{l+1}$. Compute $g_2 = g_p^a$, $Q = g_p \cdot R$, and for $0 \le i \le l$, compute $Q_i = u_i \cdot R_i$. Let $\mathbf{Q} = (Q_0, Q_1, \ldots, Q_l)$. Compute $g_1^a$ and $e(g_1, g_2)$.

Output the master key $MSK = (g_p, g_1^a, \mathbf{U})$, public key $PK = (g_q, Q, \mathbf{Q}, e(g_1, g_2))$.

- $sk_i \leftarrow Join(MSK, ID_i)$:

Parse user's identity $ID_i$ as $(ID_{i,1}, \ldots, ID_{i,l})$, where each $ID_{i,j} \in \{0,1\}$. Randomly choose $r \in \mathbf{Z}_n$, and compute the private key

$$sk_i = (d_{i,1}, d_{i,2}) = \left( g_1^a \cdot \left( u_0 \prod\nolimits_{ID_{i,j}=1} u_j \right)^r, g_p^r \right).$$

- $(Hdr, K) \leftarrow Encrypt(PK, S)$:

Given the public key $PK = (g_q, Q, \mathbf{Q}, e(g_1, g_2))$ and a receiver set $S = (ID_1, \ldots, ID_{|S|})$, proceed as follows.

1. Let $x_i = ID_i$ for $i = 1, 2, \ldots, |S|$, form the polynomial

$$f_i(x) = \prod_{1 \le j \ne i \le |S|} \frac{x - x_j}{x_i - x_j} = a_{i,1} + a_{i,2}x + \cdots + a_{i,|S|}x^{|S|-1},$$

such that $f_i(x_i) = 1$ and $f_i(x_j) = 0$.

2. Randomly choose $t \in \mathbf{Z}_n$, $v \in G_q$, $(v_1, \ldots, v_{|S|}) \in G_q$, compute $K = e(g_1, g_2)^t$, $C = v \cdot Q^t$.

3. For $i = 1, 2, \ldots, |S|$, compute

$$y_i = v_i \cdot \left( Q_0 \cdot \prod\nolimits_{ID_{i,k}=1} Q_k \right)^t.$$

4. For $i = 1, 2, \ldots, |S|$, compute

$$T_i = \sum_{j=1}^{|S|} \left( a_{j,i} \cdot y_j \right).$$

5. Let $Hdr = (T_1, \ldots, T_{|S|}, C)$, and output $(Hdr, K)$.

- $K \leftarrow Decrypt(ID_i, sk_i, Hdr)$ :

Given $ID_i$, $sk_i = (d_{i,1}, d_{i,2})$ and $Hdr = (T_1, \ldots, T_{|S|}, C)$, decrypt the header in the following steps:

1. Let $x_i = ID_i$, compute $\delta_i = T_1 + x_i T_2 + \cdots + x_i^{|S|-1} T_{|S|}$.
2. Compute

$$K = \frac{e(d_{i,1}, C)}{e(d_{i,2}, \delta_i)}.$$

3. Output $K$ as the session key.

**Correctness:** The correctness of this scheme is shown as follows.

$$\begin{aligned}
\delta_i &= T_1 + x_i T_2 + \cdots + x_i^{|S|-1} T_{|S|} \\
&= \left(a_{1,1} y_1 + a_{2,1} y_2 + \cdots + a_{|S|,1} y_{|S|}\right) + \left(x_i a_{1,2} y_1 + x_i a_{2,2} y_2 + \cdots + x_i a_{|S|,2} y_{|S|}\right) \\
&\quad + \cdots + \left(x_i^{|S|-1} a_{1,|S|} y_1 + x_i^{|S|-1} a_{2,|S|} y_2 + \cdots + x_i^{|S|-1} a_{|S|,|S|} y_{|S|}\right) \\
&= \left(a_{1,1} + a_{1,2} x_i + \cdots + a_{1,|S|} x_i^{|S|-1}\right) y_1 + \left(a_{2,1} + a_{2,2} x_i + \cdots + a_{2,|S|} x_i^{|S|-1}\right) y_2 \\
&\quad + \cdots + \left(a_{|S|,1} + a_{|S|,2} x_i + \cdots + a_{|S|,|S|} x_i^{|S|-1}\right) y_{|S|} \\
&= f_1(x_i) y_1 + f_2(x_i) y_2 + \cdots + f_{|S|}(x_i) y_{|S|} \\
&= f_i(x_i) y_i \\
&= y_i
\end{aligned}$$

According to equation (1), for any element $h_p \in G_p$ and $h_q \in G_q$, there always has $e(h_p, h_q) = 1$. Thus, in the decryption algorithm,

$$\begin{aligned}
\frac{e(d_{i,1}, C)}{e(d_{i,2}, \delta_i)} &= \frac{e\left(g_1^a \cdot \left(u_0 \prod_{ID_{i,j}=1} u_j\right)^r, v \cdot Q^t\right)}{e\left(g_p^r, v_i \cdot \left(Q_0 \prod_{ID_{i,k}=1} Q_k\right)^t\right)} = \frac{e\left(g_1^a \cdot \left(u_0 \prod_{ID_{i,j}=1} u_j\right)^r, v \cdot \left(g_p \cdot R\right)^t\right)}{e\left(g_p^r, v_i \cdot \left(u_0 \cdot R_0 \prod_{ID_{i,k}=1} u_k \cdot R_k\right)^t\right)} \\
&= \frac{e\left(g_1^a \cdot \left(u_0 \prod_{ID_{i,j}=1} u_j\right)^r, \left(g_p\right)^t\right)}{e\left(g_p^r, \left(u_0 \prod_{ID_{i,k}=1} u_k\right)^t\right)} = \frac{e\left(g_1^a, \left(g_p\right)^t\right) e\left(\left(u_0 \prod_{ID_{i,j}=1} u_j\right)^r, \left(g_p\right)^t\right)}{e\left(g_p^r, \left(u_0 \prod_{ID_{i,k}=1} u_k\right)^t\right)} \\
&= e\left(g_1^a, g_p^t\right) \cdot \frac{e\left(\left(u_0 \prod_{ID_{i,j}=1} u_j\right)^t, g_p^r\right)}{e\left(g_p^r, \left(u_0 \prod_{ID_{i,k}=1} u_k\right)^t\right)} = e(g_1, g_p^a)^t = e(g_1, g_2)^t \\
&= K
\end{aligned}$$

The tuple $(T_1, \ldots, T_{|S|})$ in the $Hdr$ can be seen as the coefficients of the Lagrange interpolation polynomial $F(x)$, where

$$\begin{aligned}
F(x) &= T_1 + x T_2 + \cdots + x^{|S|-1} T_{|S|} \\
&= f_1(x) y_1 + f_2(x) y_2 + \cdots + f_{|S|}(x) y_{|S|}
\end{aligned},$$

such that $F(x_i) = \delta_i = y_i$.

## 5.3 Discussion

A user may not discover whether he is the receiver if only based on the decryption algorithm itself. This is because even a user get a wrong $y_i'$, he may continue proceed the algorithm to output a session key $K'$. At this point, he still couldn't distinguish whether it's valid or not. One solution in practical is to encrypt the message together with its checksum to form the broadcast body. The user could try to decrypt the broadcast body using $K'$, and then verify the

validity of the decrypted message. If the message is valid, then $K'$ is the right session key, which means this user is the receiver.

The encryption algorithm is a probabilistic algorithm due to the choice of the random elements $v_i$ from $G_q$ and exponent $t$ from $Z_n$. This means in the encryption algorithm, one single input may produce a plurality of different valid outputs. Consider the following scenario: given a broadcast header $Hdr$, the adversary multiplies each component in $Hdr$ with a random element in $G_q$, and outputs the new $Hdr'$. According to the decryption algorithm and equation (1), the $G_q$ components in $Hdr$ will be eliminated eventually. This means the $Hdr'$ he outputs is also a valid header, and can be decrypted to recover the same session key. It is the main reason that the security model of our scheme is only CCA1 secure, since the adversary $\mathsf{A}$ can easily win the game every time if he can make the Decryption queries in phase 2.

## 6. Analysis

The security proof of the proposed scheme is given in this section. At the same time, the performance and functionality of the proposed scheme are analyzed and compared to the previous anonymous broadcast encryption schemes.

### 6.1 Security

We prove our scheme's security in the following subsection. The security is proven through games between a challenger $\mathsf{C}$ and an adversary $\mathsf{A}$. The proofs show that our proposed scheme has both ciphertext confidentiality and receiver anonymity against adaptive adversaries under the composite DBDH assumption and the SDA assumption.

#### 6.1.1 Ciphertext Confidentiality

**Theorem 1.** *If the composite DBDH assumption holds, then the proposed dynamic ANOBE scheme is IND-ADA-CCA1 secure.*

**Proof.** The IND-ADA-CCA1 security means that all adversaries have at most negligible advantage in winning the game in Definition 2. We now illustrate that, if there exists a PPT adversary $\mathsf{A}$ that can win the game with non-negligible advantage $\varepsilon$, then there exists a PPT challenger $\mathsf{C}$ that breaks the composite DBDH assumption with non-negligible advantage. That is, not strictly speaking, the challenger can utilize the adversary's ability to solve the DBDH problem.

Given the parameters $(p,q,G,G_T,e)$ and an DBDH instance $(g_p,g_q,g_p^{\alpha},g_p^{\beta},g_p^{\gamma},T)$, the challenger $\mathsf{C}$ interacts with the adversary $\mathsf{A}$ according to the game in Definition 2, and finally outputs a guess $\sigma$, where $\sigma=1$ means $T=e(g_p,g_p)^{\alpha\beta\gamma}$, $\sigma=0$ means $T$ is a random element. Suppose $\mathsf{A}$ can issue up to $\mu$ times joining queries and $\mu'$ times decryption queries. Let $l$ be the length of user's identities in bitstrings. The challenger $\mathsf{C}$ interacts with $\mathsf{A}$ in the following way.

**Setup.** First, $\mathsf{C}$ randomly chooses $k\in\{0,\ldots,l\}$, and let $m=4\mu$. It then chooses a vector $(v_0,v_1,\ldots,v_l)\in\{0,\ldots,m-1\}^{l+1}$ and a vector $(w_0,w_1,\ldots,w_l)\in Z_n^{l+1}$ uniformly at random. These values are all kept internal to the challenger.

For user's identity $ID_i=(ID_{i,1},\ldots,ID_{i,l})$, defines three assistant functions

$$L(ID_i)=(n-mk)+v_0+\sum_{ID_{i,j}=1}v_j\ ,$$

$$J(ID_i)=w_0+\sum_{ID_{i,j}=1}w_j\ ,$$

$$K(ID_i)=\begin{cases}0, & \text{if } v_0+\sum_{ID_{i,j}=1}v_j\equiv 0 \pmod m\\ 1, & \text{otherwise}\end{cases}$$

C then let $g_1=g_p^{\alpha}$, $g_2=g_p^{\beta}$, $u_0=g_1^{n-mk+v_0}\cdot g_p^{w_0}$. For $1\le i\le l$, computes $u_i=g_1^{v_i}\cdot g_p^{w_i}$. Let $\mathbf{U}=(u_0,u_1,\ldots,u_l)$. Then, C randomly chooses $R\in G_q$, and $(R_0,R_1,\ldots,R_l)\in G_q^{l+1}$. It computes $Q=g_p\cdot R$, and for $0\le i\le l$, computes $Q_i=u_i\cdot R_i$. Let $\mathbf{Q}=(Q_0,Q_1,\ldots,Q_l)$. It then computes $e(g_1,g_2)$. The public key $PK=(g_q,Q,\mathbf{Q},e(g_1,g_2))$. C gives the public parameter $(n,G,G_T,e)$ and $PK$ to the adversary A. From the perspective of A, the distribution of the public parameters is identical to the real construction.

**Phase 1.** In this phase, A can issue two types queries to C. For the joining queries, C responds in the following way.

Suppose the adversary A issues a query for identity $ID_i$. If $K(ID_i)=0$, the challenger C aborts the game and outputs a random guess $\sigma$. Otherwise, C chooses a random $r\in Z_n$, and constructs the private key

$$sk_i=(d_{i,1},d_{i,2})=\left(g_2^{\frac{-J(ID_i)}{L(ID_i)}}\cdot\left(u_0\prod\nolimits_{ID_{i,j}=1}u_j\right)^r, g_2^{\frac{-1}{L(ID_i)}}\cdot g_p^r\right).$$

This is a valid private key from the perspective of A, because if we let

$$r'=r-\frac{\beta}{L(ID_i)},$$

we have

$$\begin{aligned}d_{i,1}&=g_2^{\frac{-J(ID_i)}{L(ID_i)}}\cdot\left(u_0\prod\nolimits_{ID_{i,j}=1}u_j\right)^r=g_2^{\frac{-J(ID_i)}{L(ID_i)}}\cdot\left(g_1^{n-mk+v_0}\cdot g_p^{w_0}\prod\nolimits_{ID_{i,j}=1}\left(g_1^{v_j}\cdot g_p^{w_j}\right)\right)^r\\&=g_2^{\frac{-J(ID_i)}{L(ID_i)}}\cdot\left(g_1^{L(ID_i)}\cdot g_p^{J(ID_i)}\right)^r=g_1^{\beta}\cdot g_1^{-\beta}\cdot g_p^{\beta\frac{-J(ID_i)}{L(ID_i)}}\cdot\left(g_1^{L(ID_i)}\cdot g_p^{J(ID_i)}\right)^r\\&=g_1^{\beta}\cdot\left(g_1^{L(ID_i)}\cdot g_p^{J(ID_i)}\right)^{\frac{-\beta}{L(ID_i)}}\cdot\left(g_1^{L(ID_i)}\cdot g_p^{J(ID_i)}\right)^r\\&=g_1^{\beta}\cdot\left(g_1^{L(ID_i)}\cdot g_p^{J(ID_i)}\right)^{r-\frac{\beta}{L(ID_i)}}\\&=g_1^{\beta}\cdot\left(u_0\prod\nolimits_{ID_{i,j}=1}u_j\right)^{r'}\end{aligned}$$

and $d_{i,2}=g_2^{\frac{-1}{L(ID_i)}}\cdot g_p^r=g_p^{r-\frac{\beta}{L(ID_i)}}=g_p^{r'}$. This private key is computable if $L(ID_i)\ne 0$, which can be implied by $K(ID_i)\ne 0$ for easy analysis.

For each decryption query $(Hdr_i,S_i)$, C generates the private key for an identity in $S_i$, then runs the decryption algorithm to get the session key $K_i$ and returns it to A.

**Challenge.** A chooses a set $S^*$ of identities that he hasn't queried the private keys before. C encrypts the set $S^*$ in the following way.

For any $ID_i\in S^*$, if $v_0+\sum_{ID_{i,j}=1}v_j\ne mk$, C aborts the game and outputs a random guess $\sigma$. Otherwise, C encrypts it as follows.

1. Let $x_i=ID_i$ for $i=1,2,\ldots,|S^*|$, forms the polynomial

$$f_i(x)=\prod_{1\le j\ne i\le|S^*|}\frac{x-x_j}{x_i-x_j}=a_{i,1}+a_{i,2}x+\cdots+a_{i,|S^*|}x^{|S^*|-1},$$

such that $f_i(x_i)=1$ and $f_i(x_j)=0$.

2. Randomly chooses $v\in G_q$, $(v_1,\ldots,v_{|S^*|})\in G_q$. Let $C=v\cdot g_p^{\gamma}$, $K^*=T$.

3. For $i = 1, 2, \ldots, |S^*|$, computes $y_i = v_i \cdot g_p^{\gamma \cdot J(ID_i)}$.

4. For $i = 1, 2, \ldots, |S^*|$, computes

$$T_i = \sum_{j=1}^{|S^*|} \left( a_{j,i} \cdot y_j \right).$$

5. Let $Hdr^* = (T_1, \ldots, T_{|S^*|}, C)$, and outputs $(Hdr^*, K^*)$.

If $T = e(g_p, g_p)^{\alpha\beta\gamma}$, this is a valid ciphertext, since $K^* = T = e(g_p, g_p)^{\alpha\beta\gamma} = e(g_1, g_2)^{\gamma}$, and

$$\begin{aligned} y_i &= v_i \cdot g_p^{\gamma \cdot J(ID_i)} \\ &= v_i \cdot g_p^{\gamma \cdot \left( w_0 + \sum_{ID_i, j=1} w_j \right)} = v_i \cdot g_p^{\gamma \cdot \left( w_0 + \sum_{ID_i, j=1} w_j \right)} \cdot g_1^{\gamma \cdot \left( n - mk + v_0 + \sum_{ID_i, j=1} v_j \right)} \\ &= v_i \cdot \left( g_1^{n-mk+v_0} \cdot g_p^{w_0} \cdot g_1^{\sum_{ID_i, j=1} v_j} \cdot g_p^{\sum_{ID_i, j=1} w_j} \right)^{\gamma} \\ &= v_i \cdot \left( g_1^{n-mk+v_0} \cdot g_p^{w_0} \cdot \prod_{ID_i, j=1} g_1^{v_j} \cdot \prod_{ID_i, j=1} g_p^{w_j} \right)^{\gamma} \\ &= v_i \cdot \left( u_0 \cdot \prod_{ID_i, j=1} u_j \right)^{\gamma} \end{aligned}$$

If the decryption oracle in phase 1 has answered the value $K^*$ to A, C aborts the game and outputs a random guess $\sigma$. This is because answered the value $K^*$ means A has queried $(Hdr^*, S^*)$ or other valid forms in phase 1. A can easily win the game in this case.

Otherwise, A cannot get any useful information from the decryption oracle. In this case, the challenger C selects a random element $K' \in G_T$ and a random bit $b \in \{0,1\}$, sets $K_b = K^*$, $K_{1-b} = K'$, and gives $(Hdr^*, K_0, K_1)$ to A.

**Phase 2.** A issues additional joining queries as in phase 1, with the restriction that he cannot issue queries for $ID_i \in S^*$.

**Guess.** A outputs a guess $b'$ for $b$.

If $b' = b$, C outputs the guess $\sigma = 1$, which means $T = e(g_p, g_p)^{\alpha\beta\gamma}$. If $b' \neq b$, C outputs the guess $\sigma = 0$, which means $T$ is a random element in $G_T$.

**Analysis.** We now analyze the probability that the given DBDH problem can be solved by the challenger.

The game may be aborted before it finishes. If the aborting happens, C has only $1/2$ probability to solve the problem. Moreover, if the given $T$ is a random value in $G_T$, the adversary A has no advantage in winning the game, so C also has only $1/2$ probability to solve the problem, no matter the game is aborted or not. Therefore the probability

$$\begin{aligned} \Pr[solved] &= \Pr\left[\sigma = 0 \mid T \neq e(g_p, g_p)^{\alpha\beta\gamma}\right] \Pr\left[T \neq e(g_p, g_p)^{\alpha\beta\gamma}\right] \\ &\quad + \Pr\left[\sigma = 1 \mid T = e(g_p, g_p)^{\alpha\beta\gamma}\right] \Pr\left[T = e(g_p, g_p)^{\alpha\beta\gamma}\right] \\ &= \frac{1}{4} + \frac{1}{2} \Pr\left[\sigma = 1 \mid T = e(g_p, g_p)^{\alpha\beta\gamma}\right] \end{aligned}$$

If $T = e(g_p, g_p)^{\alpha\beta\gamma}$, the probability

$$\begin{aligned} \Pr[\sigma = 1] &= \Pr[\sigma = 1 \mid abort] \Pr[abort] + \Pr\left[\sigma = 1 \mid \overline{abort}\right] \Pr\left[\overline{abort}\right] \\ &= \frac{1}{2} \Pr[abort] + \Pr\left[b' = b \mid \overline{abort}\right] \Pr\left[\overline{abort}\right] \\ &= \frac{1}{2} \left(1 - \Pr\left[\overline{abort}\right]\right) + \left(\frac{1}{2} + \varepsilon\right) \Pr\left[\overline{abort}\right] \\ &= \frac{1}{2} + \varepsilon \cdot \Pr\left[\overline{abort}\right] \end{aligned}$$

where $\varepsilon$ is $\mathrm{A}$'s advantage in winning the game.

We now discuss the probability that the game aborts when $T = e(g_p, g_p)^{\alpha\beta\gamma}$. The abort may happen in three cases. 1) In the joining query, game aborts when $K(ID_i) = 0$. 2) In the challenge phase, game aborts when there exists an $ID_i$ such that $v_0 + \sum_{ID_i, j=1} v_j \neq mk$. 3) In the challenge phase, game aborts if the $K^*$ equals some former value that the decryption oracle has answered in phase 1.

The first two cases are independent from each other since the identities in the joining query are all different from the identities in the challenge set. So we have

$$\begin{aligned}\Pr\left[\overline{abort_1}\right] &\geq \Pr\left[\bigwedge_{i=1}^{\mu} K(ID_i) = 1\right] = 1 - \Pr\left[\bigvee_{i=1}^{\mu} K(ID_i) = 0\right] \\ &\geq 1 - \sum_{i=1}^{\mu} \Pr\left[K(ID_i) = 0\right] \\ &= 1 - \frac{\mu}{m}\end{aligned}$$

and

$$\begin{aligned}\Pr\left[\overline{abort_2}\right] &= \Pr\left[\bigwedge_{i=1}^{|S^*|} v_0 + \sum_{ID_i, j=1} v_j = mk\right] = \frac{1}{(l+1)^{|S^*|}} \cdot \Pr\left[\bigwedge_{i=1}^{|S^*|} K(ID_i) = 0\right] \\ &\geq \frac{1}{(l+1)^{|S^*|}} \cdot \frac{1}{m^{|S^*|}} = \frac{1}{(ml+m)^{|S^*|}}\end{aligned}$$

The probability of the third case is also independent from the first two cases since the $K$ in the encryption algorithm is only affected by the random exponent $t$ selected from $Z_n$. For a single encryption, the probability that $K$ equals $K^*$ is $1/p$. Thus we have

$$\Pr\left[\overline{abort_3}\right] \geq \left(\frac{p-1}{p}\right)^{\mu'}.$$

Then

$$\begin{aligned}\Pr\left[\overline{abort}\right] = \Pr\left[\overline{abort_1} \wedge \overline{abort_2} \wedge \overline{abort_3}\right] &\geq \left(1 - \frac{\mu}{m}\right) \cdot \frac{1}{(ml+m)^{|S^*|}} \cdot \left(\frac{p-1}{p}\right)^{\mu'} \\ &= \frac{3}{4(4\mu l + 4\mu)^{|S^*|}} \cdot \left(\frac{p-1}{p}\right)^{\mu'}\end{aligned}$$

Since $\mu$ and $\mu'$ are polynomially bounded, and $l, |S^*|$ are reasonably small values, the probability $\Pr[\overline{abort}]$ is non-negligible. Therefore the probability

$$\begin{aligned}\Pr\left[solved\right] &= \frac{1}{4} + \frac{1}{2}\Pr\left[\sigma = 1 \mid T = e(g_p, g_p)^{\alpha\beta\gamma}\right] = \frac{1}{4} + \frac{1}{4} + \frac{\varepsilon}{2} \cdot \Pr\left[\overline{abort}\right] \\ &= \frac{1}{2} + \frac{\varepsilon}{2} \cdot \frac{3}{4(4\mu l + 4\mu)^{|S^*|}} \cdot \left(\frac{p-1}{p}\right)^{\mu'}\end{aligned}$$

This indicates that, if $\varepsilon$ is non-negligible, the challenger $\mathrm{C}$'s advantage in solving the composite DBDH problem is also non-negligible. So we have the conclusion that if the composite DBDH assumption holds, then no PPT adversary can win the game with non-negligible advantage. □

### 6.1.2 Receiver Anonymity

**Theorem 2.** *If the SDA assumption holds, then the proposed dynamic ANOBE scheme is ANON-ADA-CCA1 secure.*

**Proof.** We prove this theorem by describing a sequence of games $0,1,2,\cdots$ such that Game 0 is the original game in Definition 3, and the last game is the target game that the adversary's winning probability is equal to $1/2$. The transitions between each Game $i$ and Game $i+1$ are negligible to all PPT adversaries. Since the number of games is a constant, we can have the conclusion that the adversary's winning probability in the original game is negligibly close to $1/2$. Suppose $\mathrm{A}$ can issue up to $\mu$ times joining queries and $\mu'$ times decryption queries. We define these games as follows.

**Game 0.** In this game, the challenger $\mathrm{C}$ interacts with the adversary $\mathrm{A}$ using real algorithms. $\mathrm{C}$ first runs $Setup(1^k)$ to generate master key $MSK$ and pubic key $PK$, and gives $PK$ to $\mathrm{A}$. In phase 1, $\mathrm{A}$ issues queries to the two oracles. For $\mathrm{A}$'s each query, $\mathrm{C}$ runs the corresponding algorithm and sends the result to $\mathrm{A}$. In the challenge phase, $\mathrm{C}$ receives $\mathrm{A}$'s sets $S_0$ and $S_1$, chooses a random value $b \in \{0,1\}$, and encrypts the set $S_b$ to get $(Hdr_b, K_b)$. $\mathrm{C}$ gives the challenge pair $(Hdr_b, K_b)$ to $\mathrm{A}$. In phase 2, $\mathrm{C}$ continues responding $\mathrm{A}$'s queries for $ID_i \notin S_0 \Delta S_1$ by running the algorithm $Join(MSK, ID_i)$. At last, $\mathrm{A}$ outputs a bit $b'$ and wins the game if $b' = b$.

We define $\mathrm{S}_0$ be the event that $\mathrm{A}$ wins the game in Game 0.

**Game 1.** We now make one small change to the above Game 0. We define the Game 1 to be the game that the decryption oracle has never answered the $K_b$ in phase 1.

Let $\mathrm{S}_1$ be the event that $\mathrm{A}$ wins the game in Game 1.

Define $F$ to be the event that $\mathrm{C}$ has answered the $K_b$ in phase 1. If $F$ occurs, $\mathrm{A}$ may have some advantages to win the game. If $F$ does not occur, $\mathrm{A}$ cannot get any useful information from the decryption oracle, at the same time the Game 0 and Game 1 proceed identically, with the same output. That is, we can say that $\mathrm{S}_0 \wedge \neg F \Leftrightarrow \mathrm{S}_1 \wedge \neg F$. Therefore, we have

$$\begin{aligned} \left|\Pr[\mathrm{S}_0] - \Pr[\mathrm{S}_1]\right| &= \left|\Pr[\mathrm{S}_0 \wedge F] + \Pr[\mathrm{S}_0 \wedge \neg F] - \Pr[\mathrm{S}_1 \wedge F] - \Pr[\mathrm{S}_1 \wedge \neg F]\right| \\ &= \left|\Pr[\mathrm{S}_0 \wedge F] - \Pr[\mathrm{S}_1 \wedge F]\right| \\ &\le \Pr[F] \end{aligned}$$

From the analysis in Theorem 1, we know that for a single encryption, the probability that $K$ equals $K_b$ is $1/p$. Since $\mathrm{A}$ can issue up to $\mu'$ times decryption queries, it is clear that

$$\left|\Pr[\mathrm{S}_0] - \Pr[\mathrm{S}_1]\right| \le \Pr[F] \le \frac{\mu'}{p},$$

which is negligible.

**Game 2.** In this game, we make changes to the way that the set $S_b$ is encrypted in the challenge phase. Let $(ID_1, \ldots, ID_m) \in S_b$ be the elements that aren't exist in both $S_0$ and $S_1$, i.e., $\{ID_1, \ldots, ID_m\} = S_b \setminus (S_0 \cap S_1)$, which $1 \le m \le |S_b|$. For notation convenience, we use $\{ID_1, \ldots, ID_m, ID_{m+1}, \ldots, ID_{|S_b|}\}$ to represent the set $S_b$. In the challenge phase, we add one judgment to the encryption step 3: for $i = 1, 2, \ldots, |S_b|$, if $1 \le i \le m$, the $y_i$ is computed in the following way:

$$y_i = \left(u_0 \prod_{ID_{i,j}=1} u_j\right)^t,$$

otherwise, $y_i$ is computed in the normal way.

Let $\mathrm{S}_2$ be the event that $\mathrm{A}$ wins the game in Game 2. We now claim that $\mathrm{A}$ can only distinguish Game 1 from Game 2 with negligible probability.

The ciphertext $(Hdr_b, K_b)$ in Game 2 is valid since the modified $y_1,\ldots,y_m$ are valid forms of the original ones. Given the $(Hdr_b, K_b)$, A may substitute an identity $ID_i$ he chooses into the interpolation polynomial to get a $y_i$. If $ID_i \in S_0 \cap S_1$, the $y_i$ he gets is the same in both Game 1 and Game 2. If $ID_i \in \{ID_1,\ldots,ID_m\}$, the $y_i$ he gets in Game 2 is the modified one in $G_p$, and in Game 1 is the original one in group $G$. If A chooses $ID_i \notin S_0 \cap S_1$ and from the set other than $S_b$, the $y_i$ he gets in the both games are some random-like elements in group $G$. According to the SDA assumption, A cannot distinguish whether a given element is in group $G$ or in the subgroup $G_p$. Therefore for each $y_i \in \{y_1,\ldots,y_m\}$, the difference between Game 1 and Game 2 is at most negligible. Let $\varepsilon_1$ be A's advantage in the SDA assumption, which is negligible, then we have

$$\left|\Pr[S_1]-\Pr[S_2]\right| \le |S_b|\cdot\varepsilon_1,$$

which is also negligible.

**Game 3.** We continue making changes to $y_1,\ldots,y_m$. In this game's challenge phase, C assigns $m$ random elements from $G_p$ for $y_1,\ldots,y_m$ in encryption step 3. Other steps remain unchanged.

Let $S_3$ be the event that A wins the game in Game 3.

The private values $(u_0,u_1,\ldots,u_l)$ are uniformly distributed in group $G_p$, which are unknown to A. In the two different games, unless the random exponents $t$ in encryption step 2 are the same value, it is hard for A to distinguish the $y_i$ in Game 2 from the random $y_i$ in game 3. Thus in A's view, the $y_i$ in Game 2 and the $y_i$ in Game 3 are indistinguishable. We then have

$$\left|\Pr[S_2]-\Pr[S_3]\right| \le \frac{1}{p}.$$

In Game 3, If A tries $ID_i \notin S_0 \cap S_1$ into the polynomial, no matter $ID_i$ is from $S_b$ or not, he'll get a random element. Moreover, since A cannot issue joining queries for $ID_i \in S_0 \Delta S_1$, he cannot distinguish whether the chosen $ID_i$ is in $S_b$ by trying to decrypt $y_i$. That is, the ciphertext C outputs contains no extra information about the identities $ID_i \in S_0 \Delta S_1$. Then

$$\Pr[S_3] = \frac{1}{2}.$$

Finally we have the conclusion that

$$\left|\Pr[S_0]-\frac{1}{2}\right| = \left|\Pr[S_0]-\Pr[S_3]\right| \le \frac{\mu'}{p} + |S_b|\cdot\varepsilon_1 + \frac{1}{p},$$

which is also negligible. □

## 6.2 Comparison

The scheme's performance and functionality are analyzed in this subsection, in contrast to previous anonymous broadcast encryption schemes.

We compared our scheme with the schemes in [13-16] in the following aspects: system's public key length, user's private key length, ciphertext length and decryption attempts. In the meantime, the functionality features of our scheme are also compared with those previous schemes. The arbitrary sender feature allows any user in the system to broadcast messages, not just the trusted party. The dynamic joining feature allows the user to join the system freely, without the need to update the public key. The identity-based feature means the scheme is based on IBE, in which the user's identity in the system could be an arbitrary bit string. The

indistinguishability level and the security model (random oracle model / standard model) of these schemes are also compared. **Table 1** shows the comparison results among our scheme and schemes in [13-16].

**Table 1.** Comparison with previous anonymous broadcast encryption schemes

| | BBW[13] | FP[14] | LPQ[15] | HPH[16] | Our |
|---|---|---|---|---|---|
| Public key length | $O(N)$ | $O(1)$ | $2N \cdot v$ | $v$ | $(l+3)v+e$ |
| Private key length | $O(1)$ | $O(\log N)$ | $4v$ | $v$ | $2v$ |
| Ciphertext length | $O(t)+\|M\|$ | $O(r\log(N/r))+\|M\|$ | $O(t)\cdot\|M\|$ | $(t+2)v+\|M\|$ | $(t+1)v+\|M\|$ |
| Decryption attempts | $O(t)$ | $O(r\log(N/r)\log N)$ | $O(t)$ | $O(t)$ | 1 |
| Arbitrary sender | ○ | × | ○ | ○ | ○ |
| Dynamic joining | × | × | × | ○ | ○ |
| Identity-based | × | × | × | ○ | ○ |
| Adaptive security | × | ○ | ○ | × | ○ |
| Indistinguishability | CCA2 | CCA2 | CCA2 | CCA2 | CCA1 |
| Security model | ROM | SM | SM | ROM | SM |

$N$ the total number of users in the system
$t$ the number of users in the receiver set
$r$ the number of revoked users in scheme [14]
$l$ the bit length of user's identity
$v$ the length of a group element
$e$ the length of a bilinear pairing (element in $G_T$)
$|M|$ the length of the plaintext message

As is shown in **Table 1**, among the previous anonymous broadcast encryption schemes, our proposed scheme only needs single decryption attempt, while the decryption algorithms in [13-16] need to continually try to find the right part in the ciphertext to decrypt the message. Meanwhile, we trade the public key and private key length for the ciphertext length to obtain the least ciphertext length. In comparison with previous schemes, our scheme achieves all these functionalities, and is CCA1 secure against adaptive adversaries in standard model.

## 7. Conclusion

The anonymous broadcast encryption is an important variant of the broadcast encryption schemes. It can effectively protect the receiver's privacy during the broadcast, which makes it very suitable for those privacy sensitive scenarios.

Compares with the previous schemes, the proposed ANOBE scheme improves the efficiency of the decryption. It needs only one decryption attempt to successfully decrypt the ciphertext. Besides, the ciphertext size in this scheme is the least among those existing schemes. However, despite the least ciphertext size, it is linear in the number of receivers. How to further reduce the ciphertext size is still a challenging problem.

In the future, we will investigate the feasibility of designing schemes with more efficient algorithms to meet practical needs. Moreover, we will give consideration to the security model to achieve the CCA2 security to enhance the scheme's security features.

## Acknowledgments

This work was supported in part by the National Science and Technology Major Project under Grant No.2013ZX03002006, the Liaoning Province Science and Technology Projects under Grant No.2013217004, the Liaoning Province Doctor Startup Fund under Grant NO.20141012, the Fundamental Research Funds for the Central Universities under Grant No. N130317002, and the Shenyang Province Science and Technology Projects under Grant No. F14-231-1-08.

## References

[1] A. Fiat and M. Naor, "Broadcast encryption," *Advances in Cryptology—CRYPTO'93*. Springer Berlin Heidelberg, pp. 480-491, 1994. Article (CrossRef Link)

[2] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," *Advances in Cryptology—CRYPTO 2001*, Springer Berlin Heidelberg, pp.480-491, 2001. Article (CrossRef Link)

[3] D. Halevy and A. Shamir, "The LSD broadcast encryption scheme," *Advances in Cryptology—CRYPTO 2002*, Springer Berlin Heidelberg, pp.47-60, 2002. Article (CrossRef Link)

[4] S. Bhattacherjee, and P. Sarkar, "Analysis and Trade-Offs for the (Complete Tree) Layered Subset Difference Broadcast Encryption Scheme," *IACR Cryptology ePrint Archive 2012 (2012)*: 337.

[5] D. H. Phan, D. Pointcheval, and M. Strefler, "Decentralized dynamic broadcast encryption," *Security and Cryptography for Networks*, Springer Berlin Heidelberg, pp.166-183, 2012. Article (CrossRef Link)

[6] Y. Dodis, and N. Fazio, "Public key broadcast encryption for stateless receivers," *Digital Rights Management*, Springer Berlin Heidelberg, pp.61-80, 2003. Article (CrossRef Link)

[7] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," A*dvances in Cryptology–CRYPTO 2005*, Springer Berlin Heidelberg, pp.258-275, 2005. Article (CrossRef Link)

[8] R. Dubois, A. Guillevic, and M. S. Le Breton, "Improved broadcast encryption scheme with constant-size ciphertext," *Pairing-Based Cryptography–Pairing 2012*, Springer Berlin Heidelberg, pp.196-202, 2013. Article (CrossRef Link)

[9] D. H. Phan, D. Pointcheval, S. F. Shahandashti, and M. Strefler, "Adaptive CCA broadcast encryption with constant-size secret keys and ciphertexts," *International journal of information security*, vol. 12, no. 4, pp.251-265, 2013. Article (CrossRef Link)

[10] C. Delerablée, "Identity-based broadcast encryption with constant size ciphertexts and private keys," *Advances in Cryptology–ASIACRYPT 2007*, pp.200-215, 2007. Article (CrossRef Link)

[11] D. Boneh and M. Hamburg, "Generalized identity based and broadcast encryption schemes," *Advances in Cryptology-ASIACRYPT 2008*, Springer Berlin Heidelberg, pp.455-470, 2008. Article (CrossRef Link)

[12] L. Zhang, Y. Hu, and Q. Wu, "Adaptively Secure Identity-based Broadcast Encryption with constant size private keys and ciphertexts from the Subgroups," *Mathematical and computer Modelling*, vol. 55, no. 1, pp.12-18, 2012. Article (CrossRef Link)

[13] A. Barth, D. Boneh, and B. Waters, "Privacy in encrypted content distribution using private broadcast encryption," *Financial Cryptography and Data Security*, Springer Berlin Heidelberg, pp.52-64, 2006. Article (CrossRef Link)

[14] N. Fazio and I. M. Perera, "Outsider-anonymous broadcast encryption with sublinear ciphertexts," *Public Key Cryptography–PKC 2012*, Springer Berlin Heidelberg, pp.225-242, 2012. Article (CrossRef Link)

[15] B. Libert, K. G. Paterson, and E. A. Quaglia, "Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model," *Public Key Cryptography–PKC 2012*, Springer Berlin Heidelberg, pp.206-224, 2012. Article (CrossRef Link)

[16] J. Hur, C. Park, and S. O. Hwang, "Privacy-preserving identity-based broadcast encryption," *Information Fusion*, vol. 13, no. 4, pp.296-303, 2012. Article (CrossRef Link)

[17] C. Gentry and B. Waters, "Adaptive security in broadcast encryption systems (with short ciphertexts)," *Advances in Cryptology-EUROCRYPT 2009*, Springer Berlin Heidelberg, pp.171-188, 2009. Article (CrossRef Link)

[18] K. Kurosawa and Y. Desmedt, "A new paradigm of hybrid encryption scheme," *Advances in Cryptology–Crypto 2004*, Springer Berlin Heidelberg, pp.426-442, 2004. Article (CrossRef Link)

[19] C. I. Fan, L. Y. Huang, and P. H. Ho, "Anonymous multireceiver identity-based encryption," *Computers, IEEE Transactions*, vol. 59, no. 9, pp.1239-1249, 2012. Article (CrossRef Link)

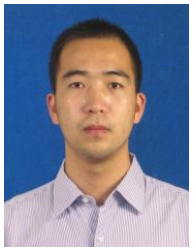
[20] D. Boneh, E. J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," *Theory of cryptography*, Springer Berlin Heidelberg, pp.325-341, 2005. Article (CrossRef Link)

[21] J. Lotspiech, S. Nusser, F. Pestoni, "Broadcast encryption's bright future". *Computer*, vol. 35, no. 8, pp.57-63, 2002. Article (CrossRef Link)

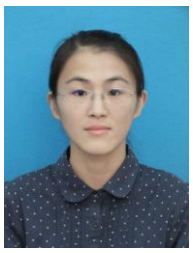
[22] A. R. Chickerur, "Introduction to broadcast encryption". http://www.cs.fsu.edu/~yasinsac/group/slides/chickerur.pdf

[23] O. Bodriagov, S. Buchegger, "P2P social networks with broadcast encryption protected privacy," *Privacy and Identity Management for Life*, Springer Berlin Heidelberg, pp.197-206, 2012. Article (CrossRef Link)

**Fu-Cai Zhou** received the Ph.D degree of Computer Software and Theory at Northeastern University, in 2001. He is currently a Professor and Doctoral Supervisor of Software College in Northeastern University. His research interests include cryptography, network security, trusted computing, basic theory and critical technology in electronic commerce.



**Mu-Qing Lin** received the Ph.D degree of Computer Application Technology at Northeastern University, in 2015. He is currently an assistant researcher of Information Security Institute in Beijing Electronic Science and Technology Institute. His main research interests include cryptography, secure cloud storage, privacy protection and network security.



**Yang Zhou** received the B.S. degree of Information Security and the M.S. degree of Software Engineering from Northeastern University, in 2012 and 2014, respectively. Her research interests include broadcast encryption and cryptography.

**Yu-Xi Li** received her B.S. degree of computer science and technology in 2012 from Sichuan University, and received her M.S. degree of computer application technology in 2014 from Northeastern University. She is currently working toward the PhD degree in Software College, Northeastern University. Her research interests include cryptography and network security.